FACIAL RECOGNITION AND AUTOMATIC ATTENDANCE SYSTEM

Dissertation submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Karamjeet Singh Bakshi (131038)

UNDER THE GUIDANCE OF

Salman Raju Talluri

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, SOLAN
MAY 2017

TABLE OF CONTENTS

DECLARATION BY THE SCHOLAR iv

SUPERVISOR'S CERTIFICATE v

ACKNOWLEDGEMENT vi

LIST OF FIGURES vii

LIST OF TABLES ix

ABSTRACT x

CHAPTER-1……………………………………………………………….

INTRODUCTION 1

1. 1 POWER SUPPLY UNIT 2

1.2 CAMERA MODULE 3

1.3 PROCESSOR
- 1.3.1 RASPBERRY PI 3 3
- 1.3.2 INTEL CORE I5 4

1.4 SOFTWARE AND ONLINE SERVICES 5
- 1.4.1 HTML 5
- 1.4.2 PHP 5
- 1.4.3 XAMPP 5
- 1.4.4 PYTHON 5
- 1.4.5 PUBNUB 6
- 1.4.6 MATLAB 6

CHAPTER-2……………………………………………………………….

PROJECT SETUP, ASSEMBLY AND FRONTEND 7

2.1 ASSEMBLING HARDWARE 8

2.2 INSTALLING SOFTWARE 8

2.3 USING PUBNUB 9

2.4 CLICKING IMAGE 10

CHAPTER-3…………………………………………………………….

FACE DETECTION, RECOGNITION AND DISPLAYING ATTENDANCE 11

3.1 CREATING A CLASSIFIER OR MODEL 11
3.1.1 CREATING AN IMAGE DATABASE 12
3.1.2 EXTRACTING FEATURES 13
3.1.3 LEARNING OR MODELLING 17

3.2 FACE DETECTION AND FEATURE EXTRACTION OF TEST IMAGES 21
3.2.1 VIOLA JONES ALGORITHM FOR DETECTING FACES 21
3.2.2 FEATURE EXTRACTION, RECOGNITION OF TEST IMAGES AND DISPLAYING ATTENDANCE 24

CHAPTER-4…………………………………………………………….

CONCLUSION 28

APPENDIX………………………………………………………… 29

REFERENCES……………………………………………………. 34

DECLARATION BY THE SCHOLAR

I hereby declare that the work reported in the B-Tech thesis entitled **"Facial Recognition and automatic attendance system"** submitted at the **Jaypee University of Information Technology, Waknaghat India,** is an authentic record of my work carried out under the supervision of **Salman Raju Talluri**. I have not submitted this work elsewhere for any other degree or diploma.

Karamjeet Singh Bakshi

Department of Electronics and Communication Engineering

Jaypee University of Information Technology, Waknaghat, India

Date:

SUPERVISOR'S CERTIFICATE

I hereby declare that the work reported in the B-Tech thesis entitled **"Facial Recognition and Automatic Attendance System"** submitted at the **Jaypee University of Information Technology, Waknaghat, India,** is an authentic record of my work carried out under the supervision of **Salman Raju Talluri**. I have not submitted this work elsewhere for any other degree or diploma.

Karamjeet Singh Bakshi

Department of Electronics and Communication Engineering

Jaypee University of Information Technology, Waknaghat, India

Date:

ACKNOWLEDGEMENT

Before Commencing this Report, I express my gratitude to my respected guide "**Salman Raju Talluri**" who encouraged me to work on a project based on image processing and hence increasing my knowledge base on the subject and also who has been utmost benign, affable, reasonable and patient towards me.

Last but not the least, I am grateful and indebted to my parents, teachers, mentor, friends and each and every person who has in some way or the other helped me in making this treatise.

Thank You!

List of Figures

Figure Number	Title	Page Number
1.1	A simple power supply circuit of output voltage 5V	2
1.2	The power supply used to power raspberry pi 3	2
1.3 (a)	Camera Pi-Cam	3
1.3 (b)	Camera DSLR	3
1.4	Raspberry Pi 3	4
1.5	Intel Core I5	5
2.1	Project Flow Chart	7
2.2 (a)	Pi connected to camera	8
2.2 (b)	Pi connected with router	8
2.3	Keys of created app	9
2.4	Monitoring real-time data	9
2.5	Web page to capture the image	10
2.6	Debug Console with capture command	10
3.1	Face Recognition Workflow	11
3.2	A sample display of database.	12
3.3	Feature Extraction Spectrum	13
3.4	Description of blocks, cells, and bins	15
3.5	Concatenation of Histogram	15
3.6	Visualizing Hog Features	16
3.7	Visualizing Hog Features from Database	17
3.8	Linear Classification in SVM	18
3.9	XOR Problem	19
3.10	Solving XOR Problem	19

3.11	Hyperplane for linearly non-separable data	19
3.12	Haar Features	22
3.13	Sum of pixels in Integral Image	22
3.14	Adaboost Learning Algorithm	23
3.15	Cascading	23
3.16	Test Images	24
3.17	Checking Test Image with Matched Class	25,26
3.19	Displaying attendance on the web page	27

List of Tables

Table Number	Title	Page Number
1.1	Basic Details of Intel Core i5	4
3.1	Details of Database	12
3.2	Prerequisites for HoG feature calculation for Fig 5.3.	16
3.3	Dimensions of Hog Features from Database	16
3.4	ECOC Code Sample	20
3.5	Details for ECOC classifier	21
3.6	Coding matrix of the classifier	21

ABSTRACT

Human face detection and recognition have significantly been a topic of research as is now used in many areas of applications such as social media, surveillance, defense and national security, etc. In this project both face detection and recognition techniques have been worked on, which includes Viola-Jones algorithm, Feature Extraction, Machine Learning algorithms like error-correcting output codes (ECOC), etc. where an unidentified test image of a person has been recognized by extracting its features and passing it to a classifier which has been trained by the features of the training images stored in the database as well as gives information regarding the person recognized. These techniques work well under robust conditions like the complex background, different face positions. These algorithms give different rates of accuracy under different conditions.

1. INTRODUCTION

Humans are hard-wired to identify faces and we do it so easily that it feels to be effortless. But actually there is a lot going on in the background, we not only are able to just detect faces but also extract necessary information like emotions, subtle variations (age or appearance), etc. which in computational terms is really a lot of data.

Even with modern computational speed still, there is a need to design efficient systems that can achieve the task of real-time face detection and recognition. In today's world face detection and recognition can have numerous applications like criminal identification, security system, image and film processing, identity verification, tagging purposes and human-computer interaction, etc.

Face Detection is used by many social media sites nowadays particularly the sites facilitating pictures like Picassa, Photobucket, and Facebook. The tagging feature adds another dimension to sharing photos among people who are in the photo and furthermore gives an idea to other individuals about who the individual is in the picture.

The objective of this project is to click image by clicking on capture tab on the web page and send it to the server and from there the image is sent to the main processing unit where faces are detected and recognized. These recognized persons are compiled in a list and then this data is sent online back on the portal from where the image capture command was sent.

The project **"Facial Recognition and Automatic Attendance System"** consists of various segments that include:

1. **Power Supply Unit**
2. **Camera Module**
3. **Processors**
4. **Software and Online Services**

1.1 Power Supply Unit

The power supply unit is the piece of hardware that converts the power provided from the main supply into usable power which can be used by the computer. It converts the alternating current (AC) into direct current (DC) as the computer works on a constant DC supply. In addition, it regulates overheating by controlling voltage, which may change automatically or manually depending upon the main supply.

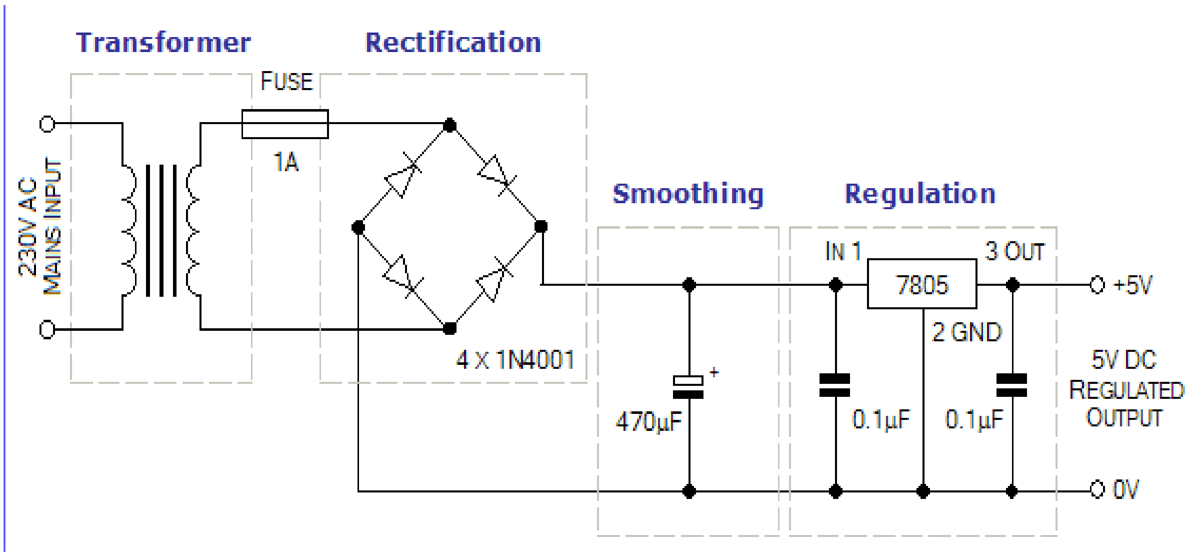


Fig 1.1. A simple power supply circuit of output voltage 5V [10]

For this project, a power supply unit that is required has an output of 5.1V and 2.5 amps with the help of a DC adapter.

Fig 1.2 Power supply used to power raspberry pi 3

1.2 Camera Module

As this project is dealing with images processing, this requires a camera with a suitable specification. The captured image should be crisp and clear so as the image sent to the processor for image processing can detect faces and extract features efficiently with reasonable accuracy. As Raspberry pi 3 has a special segment to mount a camera on it thus making the job easy for capturing images online. The camera that has been used for making our database is Canon 700D as it gives better image quality.

(a). Pi-Cam[11] (b). DSLR[12]

Fig 1.3 Camera

1.3 Processor

1.3.1 Raspberry Pi 3

Taking an image from the camera requires a processor and in the first place to capture an image online and then sending the image to the main processor for image processing, thus Raspberry pi 3 best fits our requirements for this kind of IoT applications. Specifications of Raspberry pi are as follows.

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port

- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

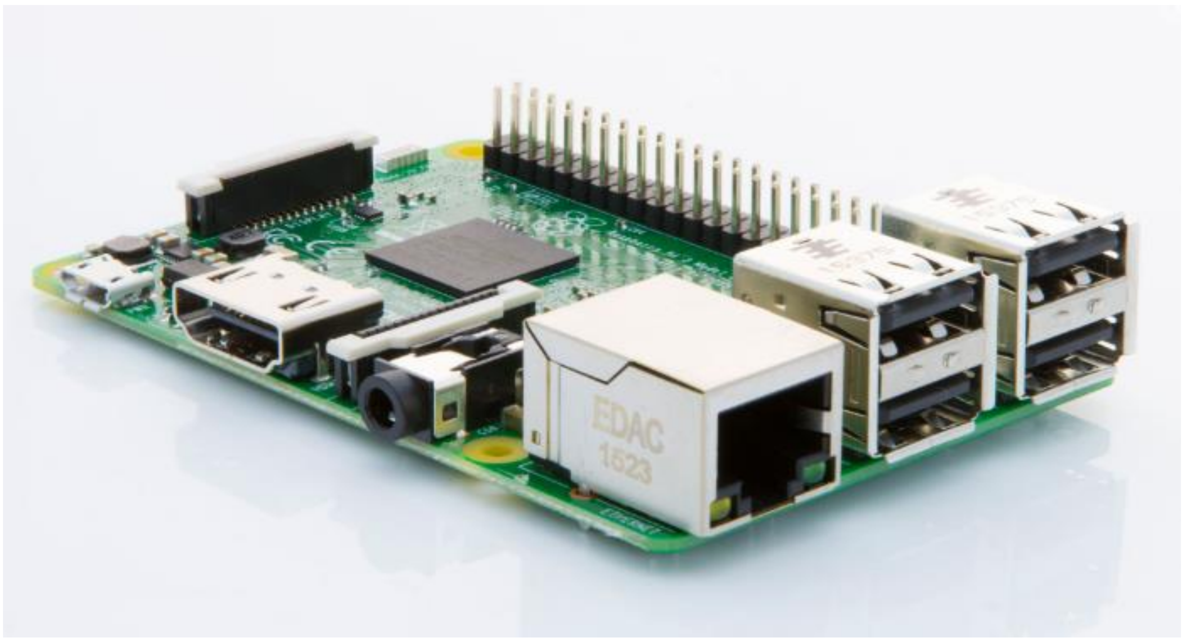


Fig 1.4 Raspberry Pi 3[13]

1.3.2 Intel Core i5

As raspberry pi cannot process large data efficiently due to less processing speed. Thus here Image Processing is done on 4th generation Intel Core i5. Basic details of the processor are shown in Table 1.1.

Number of Cores	Quad-Core
Processor number	I5-4460
Clock Speed	3.2 GHz
Cache	6 MB

Table 1.1 Basic Details of Intel Core i5

Fig. 1.5 Intel Core i5 [14]

1.4 Software and Online Services

These are the following software and online services (cloud) that are required for completion of this project.

1.4.1 HTML

HTML stands for **H**yper **T**ext **M**arkup **L**anguage, which is the most widely used language on Web to develop web pages. HTML can be understood as a skeleton that gives structure to the web page.

1.4.2 PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a broadly used open source general-purpose scripting language that is particularly suited for web development and can be embedded into HTML.

1.4.3 Xampp

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

1.4.4 Python

Python is a widely used general-purpose, high-level programming language. The script that is written on raspberry pi to capture image is a python script.

1.4.5 PubNub

PubNub is a safe worldwide Data Stream Network (DSN) and simple to utilize Programming interface that empowers its clients to connect, scale, and oversee ongoing applications and IoT devices.

1.4.6 MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment with lots of inbuilt functions where problems and solutions are expressed in familiar notations.

2. Project Setup, Assembly, and Frontend

The flow chart of the project is described in Fig 2.1 describing step by step the processes in the project for face detection, recognition and displaying attendance.

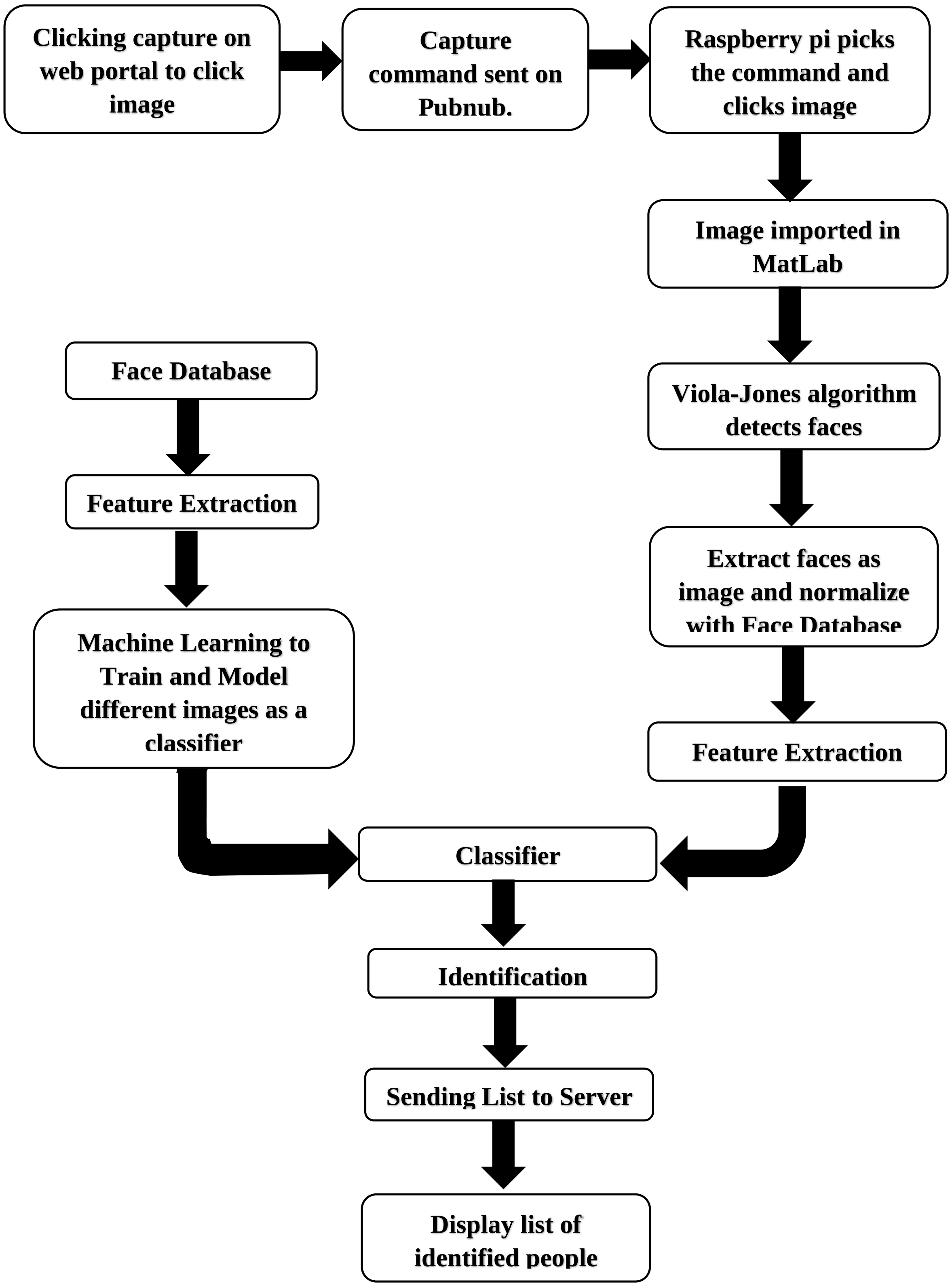
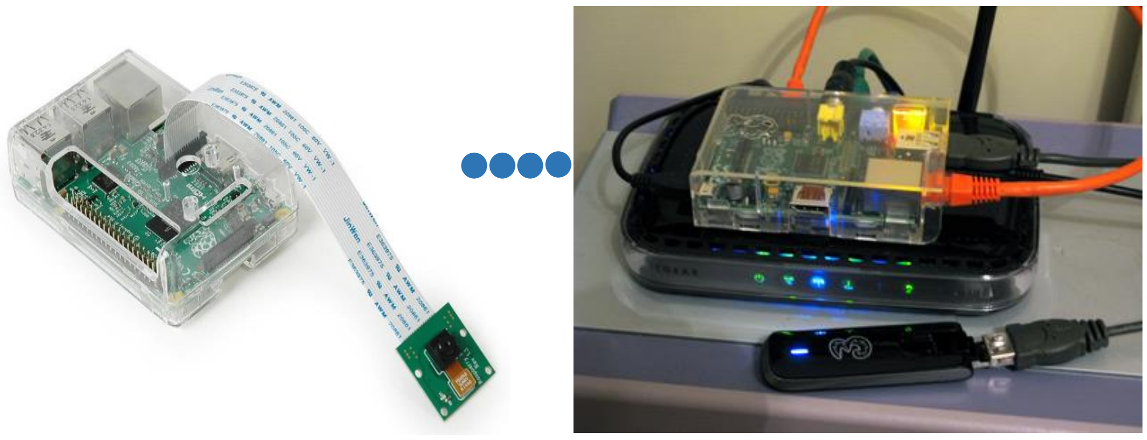

Clicking capture on web portal to click image
Capture command sent on Pubnub.
Raspberry pi picks the command and clicks image
Image imported in MatLab
Viola-Jones algorithm detects faces
Extract faces as image and normalize with Face Database
Feature Extraction
Face Database
Feature Extraction
Machine Learning to Train and Model different images as a classifier
Classifier
Identification
Sending List to Server
Display list of identified people

Fig. 2.1 Project Flow Chart

2.1 Assembling Hardware

The camera module is mounted on Raspberry Pi 3 which is further connected to Modem and then to the Internet as shown in Fig. 2.2.



(a). Pi connected to camera (b). Pi connected with router

Fig. 2.2 Camera module mounted on Pi and connected to the internet

2.2 Installing Software

Burn Raspbian Image onto SD Card

First, SD card is cleared with SD Formatter or using GParted on Ubuntu, but an equivalent utility or operating system can be used for this purpose.

Second, download Raspbian Jessie from http://downloads.raspberrypi.org/raspbian_latest.

Third, Win32 Disk Imager 0.9.5 is used to burn the image to the disk.

Finally, when it’s done, remove the SD card, insert it into Raspberry Pi and connect it to the Modem via Ethernet.

Further, the laptop is connected to Raspberry Pi with PuTTY 0.63 by entering the IP as follows.

```
$ ssh pi@192.168.2.3 # password (default): raspberry
```

Now run the following commands to install some basic tools.

```
$ sudo apt-get install python-dev python-rpi.gpio

$ sudo apt-get update

$ sudo apt-get install python-dev python-pip

$ sudo apt-get update
```

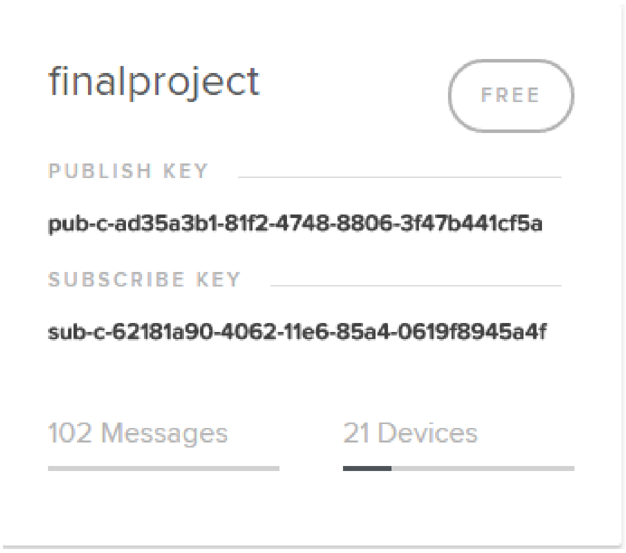
2.3 Using PubNub

PubNub is used as a Data Stream Network where we can throw data. It is being used here in this project by executing capture command and capturing image real-time. The following steps are used to connect PubNub and making an application.

First, open URL www.pubnub.com and Login.

Second, create an app then name it.

Third, click on key info and copy down your Publish Key and Subscribe Keys see Fig. 2.3 as they will be required later.



finalproject

PUBLISH KEY

pub-c-ad35a3b1-81f2-4748-8806-3f47b441cf5a

SUBSCRIBE KEY

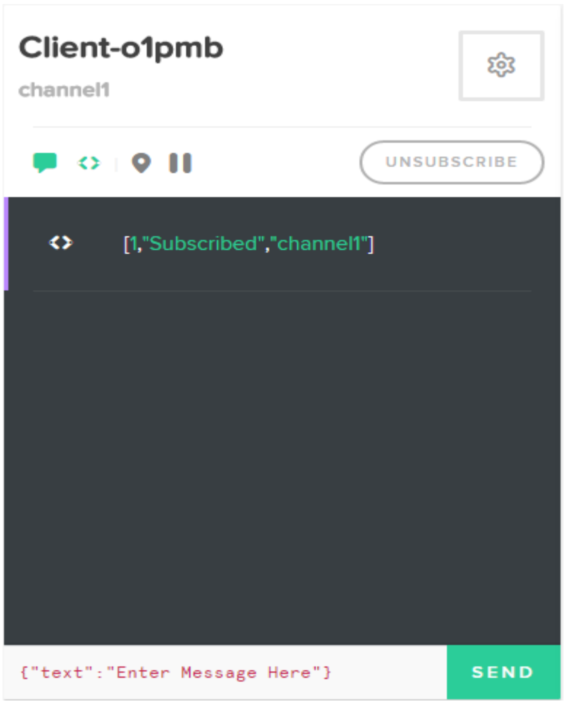
sub-c-62181a90-4062-11e6-85a4-0619f8945a4f

102 Messages

21 Devices

Fig. 2.3 Keys of created app [15]

Fourth, go to Debug Console to send or receive data in real time see Fig 2.4.



Client-o1pmb

channel1

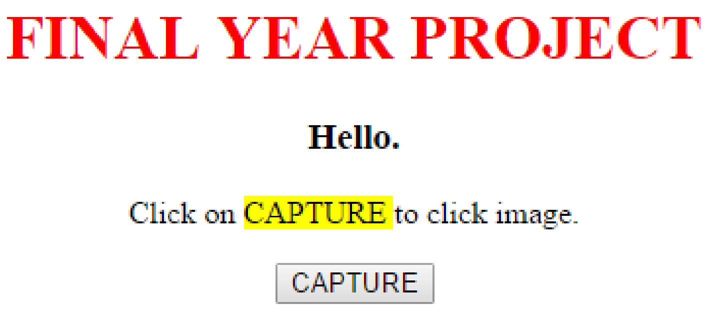
```
[1,"Subscribed","channel1"]
```

```
{"text":"Enter Message Here"}
```

Fig. 2.4 Monitoring real-time data [15]

2.4 Clicking Image

Here a web page has been created see Fig. 2.5 with the CAPTURE button. When this button is clicked capture command is sent on PubNub and image is captured by the camera mounted on Raspberry Pi 3 which is subscribed to PubNub. This image is further used for Face Detection and Recognition.


FINAL YEAR PROJECT

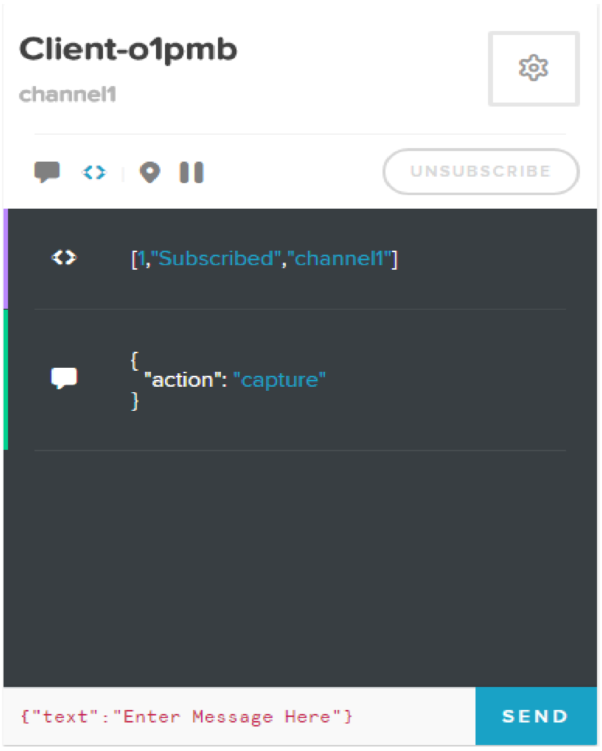
Hello.

Click on CAPTURE to click image.

Fig. 2.5 Webpage to capture the image

This web page on the FRONTEND uses HTML and on the BACKEND uses PHP which is linked with PubNub. The code in HTML for creating a web page to send a command to capture an image and Display attendance and PHP for sending and receiving data on PubNub is mentioned in APPENDIX.

Raspberry pi is subscribed to PubNub and can receive messages to the subscribed channel on Debug Console as shown in Fig. 2.6. If the message received from the channel is "capture" Raspberry pi clicks the image. Code for capturing an image in pi is given in APPENDIX.


Client-o1pmb

channel1

```
[1,"Subscribed","channel1"]
```

```
{
 "action": "capture"
}
```

Fig. 2.6 Debug Console with capture command [15]

3. Face Detection, Recognition and Displaying Attendance

Face detection and recognition work in 3 different segments.

First, is to create a Classifier or Model so that features of existing faces from the database are differentiated in form of different classes which is known as training the model and used for predicting input faces belongs to which class by passing the features of this new faces in this model. Features and Classifier have been discussed below.

Second, is to detect faces from the input frame extract its features pass it to the model that has been trained for recognition.

Third, is a minor segment which displays a list of the persons who have been recognized by the classifier on the web page.

Fig 3.1 describes Face Recognition Workflow and is discussed further in detail below.

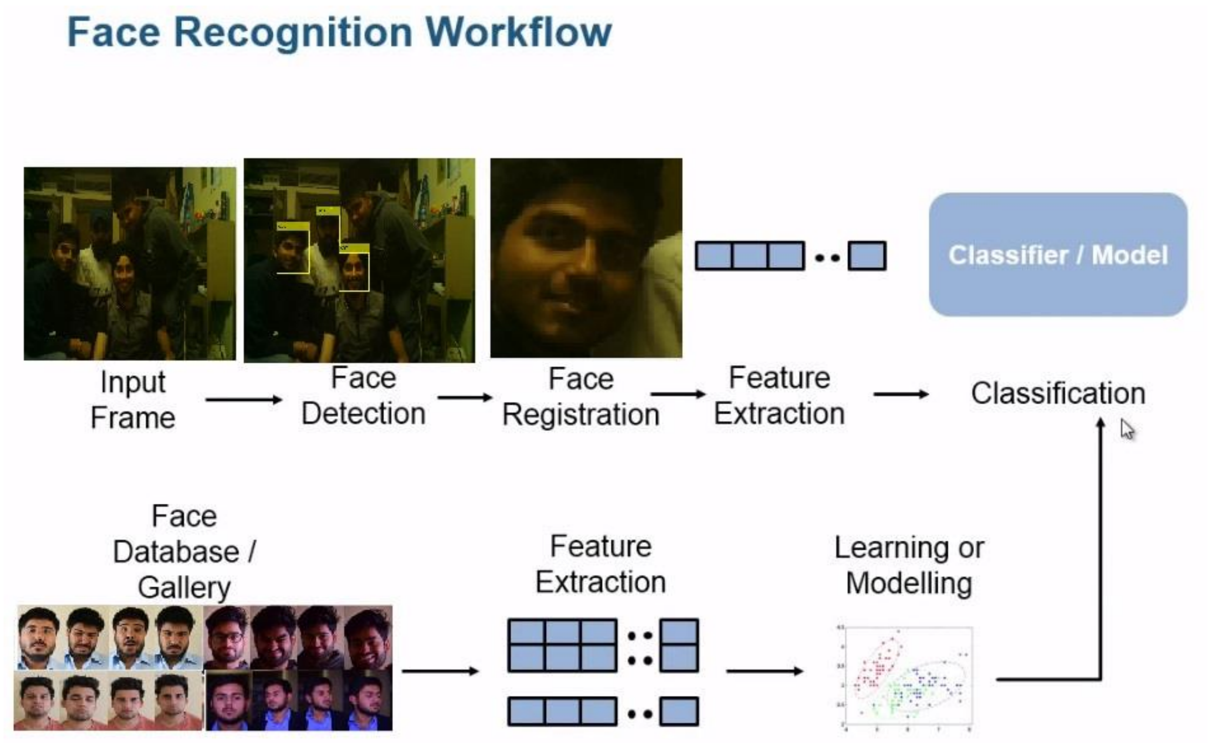


Fig. 3.1 Face Recognition Workflow [16]

3.1 Creating a Classifier or Model

The need to do this is to teach a machine is who is who as a person. To create this classifier involves 3 steps.

1. Creating a database of images of different persons.
2. Extract features of these images.
3. Using a machine learning algorithm to classify between different persons on the basis of these features.

3.1.1 Creating an Image Database

This database includes people who are to be identified later. Fig. 3.2 give a glimpse of the database of images that has been created.

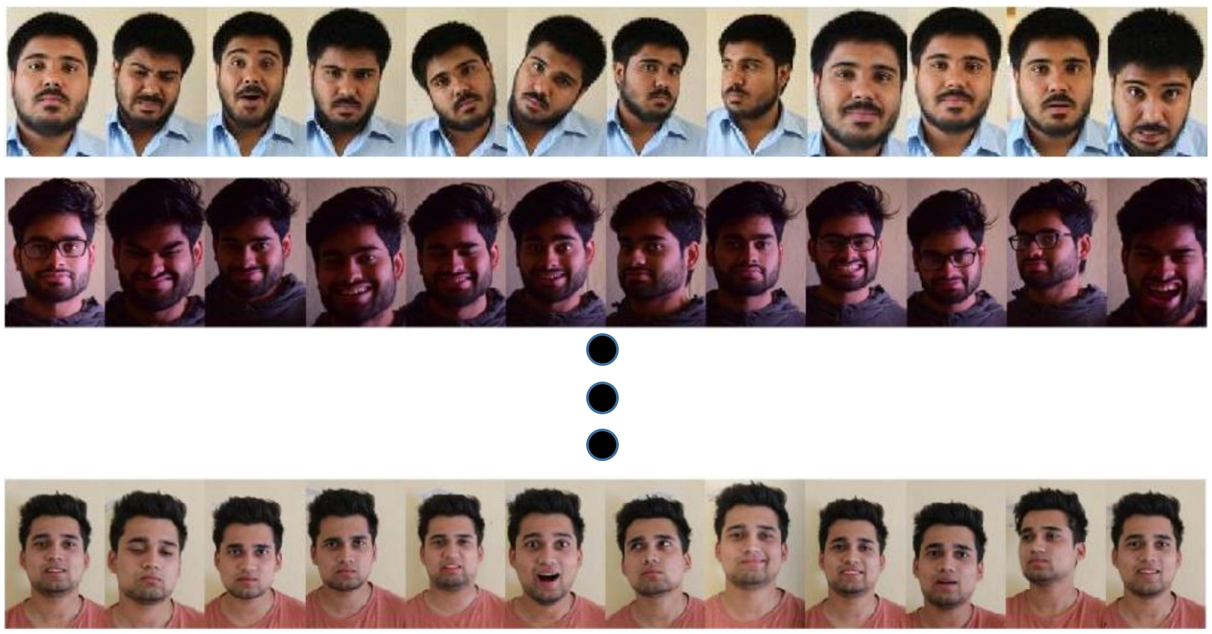


Fig. 3.2 A Sample display of database

As it can be seen from the database a person here has different facial expressions and orientations thus the features that are to be extracted will be different and of wide range thus giving a better scope for the training of the model.

Number of persons	7
No of images for each person	12
Dimensions of each image	562x375 pixels
Type of Image	RGB
Extension of Image	.JPG
Total number of images	84

Table 3.1 Details of Database

If the database is large it increases the complexity of computing features in terms of no of features and time but it also increases the efficiency of the system. Thus to create an efficient model for training creating an optimal database is necessary which is neither too big nor too small.

3.1.2 Extracting Features.

Feature extraction is used for classification and recognition of images. On the basis of these features, relevant information from the image can be extracted and can be used further for classification. There are numerous feature extraction methods available.

Criteria for choosing features given by Lippman are as follows:

a. Features must contain information that should be different for different classes.
b. Not be sensitive to irrelevant information as input
c. Limited in the count for efficient computation of discriminant function and restrict the amount of training data required.

As raw pixel information has no discriminative information to distinguish between different faces, thus to detect and recognize a face Feature Extraction is a critical and necessary step. Feature extraction can be done at different levels which are described in the spectrum is depicted in Fig. 3.3.

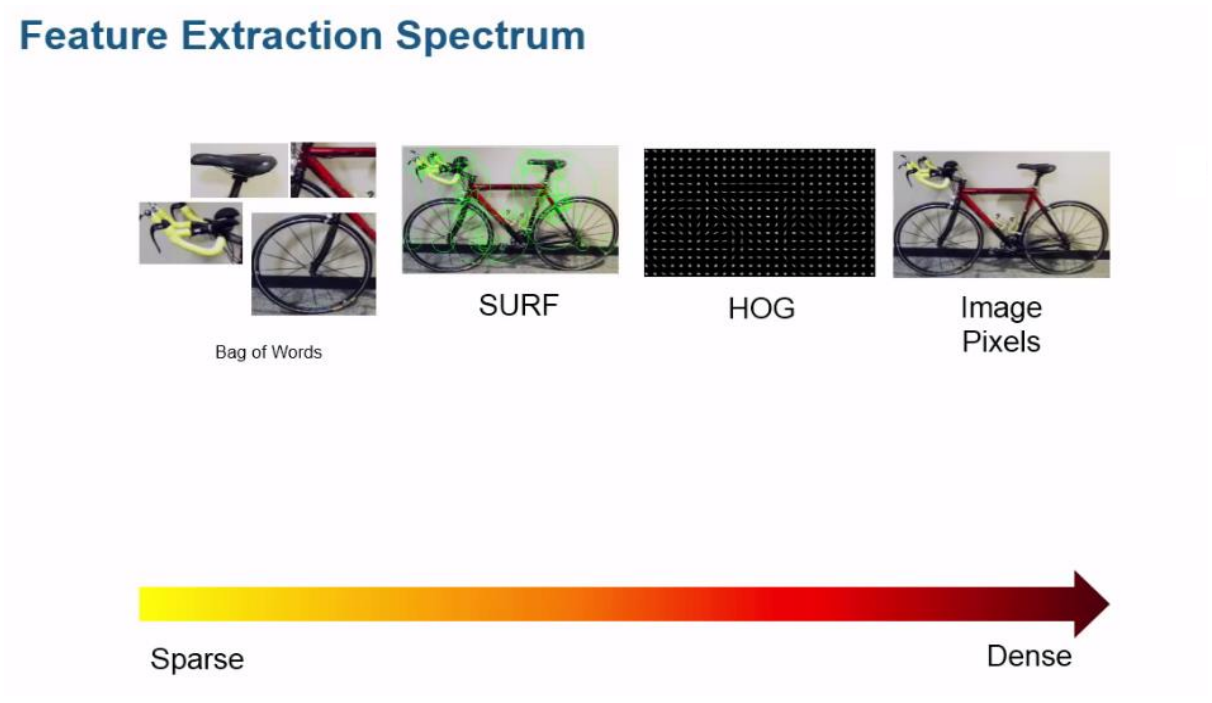


Fig. 3.3 Feature Extraction Spectrum [16]

Each pixel by itself is just a number to make sense of these pixels with the pixels around we relate the pixels with other pixels around them. On the basis of these pixels relations, various feature extraction methods have been arranged from left to right.

Bag of Visual Words (BoVW) downplays spatial information in the image and classifies on the basis of the histogram of the frequency of visual words. The set of visual words forms a visual vocabulary, which is constructed by clustering a large corpus of features.

Speed Up Robust Features (SURF) uses LoG (Laplacian of Gaussian) to detect keypoints in an image and extracts features around these key points similar to Scale Invariant Feature Transform(SIFT) but is faster than SIFT.

But our focus will be on **HoG (Histogram of Oriented Gradients) Feature [1]** extraction. Generally, Hog features are used to determine the shape of the objects but due to their high dimensionality of images in our database, they have worked reasonably well. Also, the test images that has been used for recognition have also been taken under similar conditions of the database that has been used to train the classifier.

Steps involved in calculating HoG Features:

1. Computing centered Vertical and Horizontal gradients.

   Centered $\quad f(x) = \lim_{h\to 0}\left(\frac{f(x+h)-f(x-h)}{2h}\right)$

2. Using filter masks Compute Gradient Magnitude and Orientation.

-1
0
1

-1	0	1

Filter Masks

   Gradient Magnitude $\quad s = \sqrt[2]{(sx)2 + (sy)2}$

   Unsigned Orientation $\quad \theta = \text{mod arctan}\left(\frac{sy}{sx}\right)$

3. Say for a 64x64 size image
   - Divide the image in 16x16 block with 50% overlap see Fig 3.4.

     7x7 = 49 blocks in total.

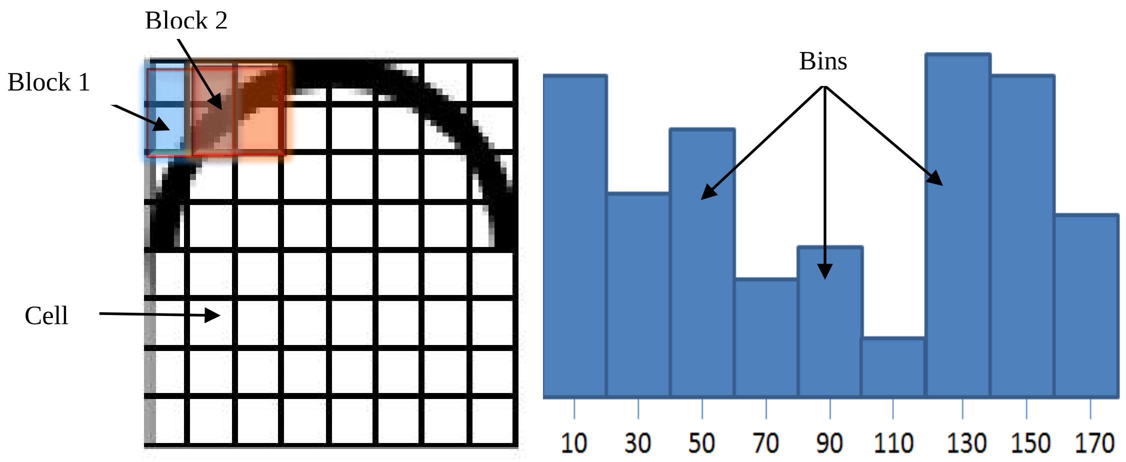


Fig. 3.4 Description of blocks, cells, and bins [18]

4. Each block has 2x2 cells see Fig 3.4 with size 8x8.
5. Quantize the gradients into 9 bins as shown in Fig3.4.
   - The vote is gradient magnitude.
   - Interpolate gradient values linearly by voting between nearest bin centers.
     - Example: if $\theta = 85^o$
     - Distance from nearest bin 70 and 90 are 15 and 5 degrees respectively.
     - Thus their votes are 5/20 = 1/4 and 15/20 = 3/4.
6. Concatenate Histograms and make it a 1D matrix.

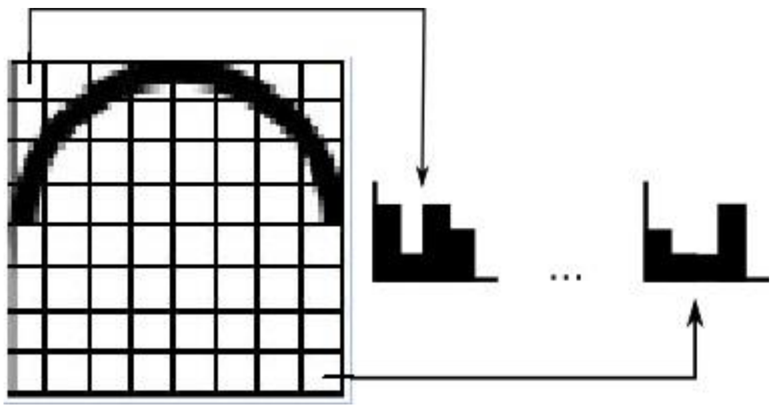


Fig. 3.5 Concatenation of Histogram

Cell size	[8 8]
Block size	[2 2]
Block overlap	50%
Number of bins	9
Input image	64x64x3
Signed Orientation	False

Table 3.2 Prerequisites for HoG feature calculation for Fig 5.3.

The formula for calculating HoG Features in Matlab.

BlocksperImage = floor value of [(size(Image)/Cell size – Block size)/(Block size – Block overlap) + 1]

Where floor takes floor value and size depicts the dimensions of the image rest of the other terms are mentioned in Table 3.2.

Number of HoG Features = BlocksperImage x Number of Bins x Block size

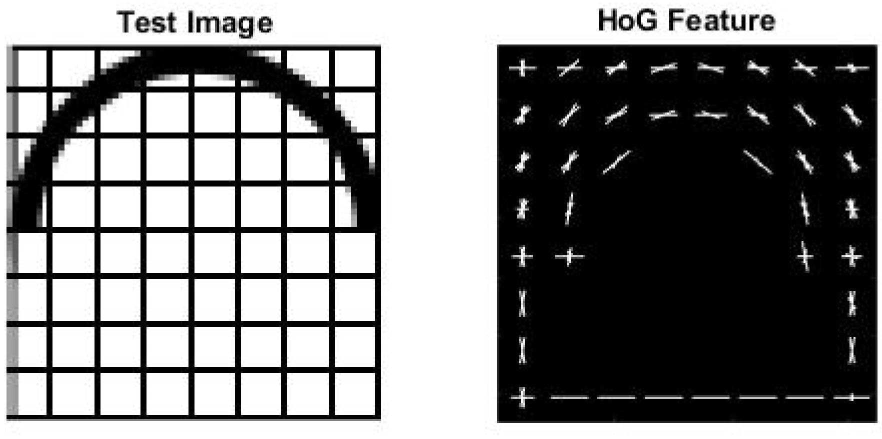


Fig 3.6 Visualizing Hog Features

Dimensionality of extracted HoG Features from the database has been depicted in the following Table 3.3.

Number of image sets	7
Number of images per set	12
Total no of test images	7x12 = 84
Dimensionality of HoG	111780
HoG Features net dimensionality	84x111780

Table 3.3 Dimensions of Hog Features from Database

HoG Features of a sample image from the database have been depicted in Fig. 3.7.

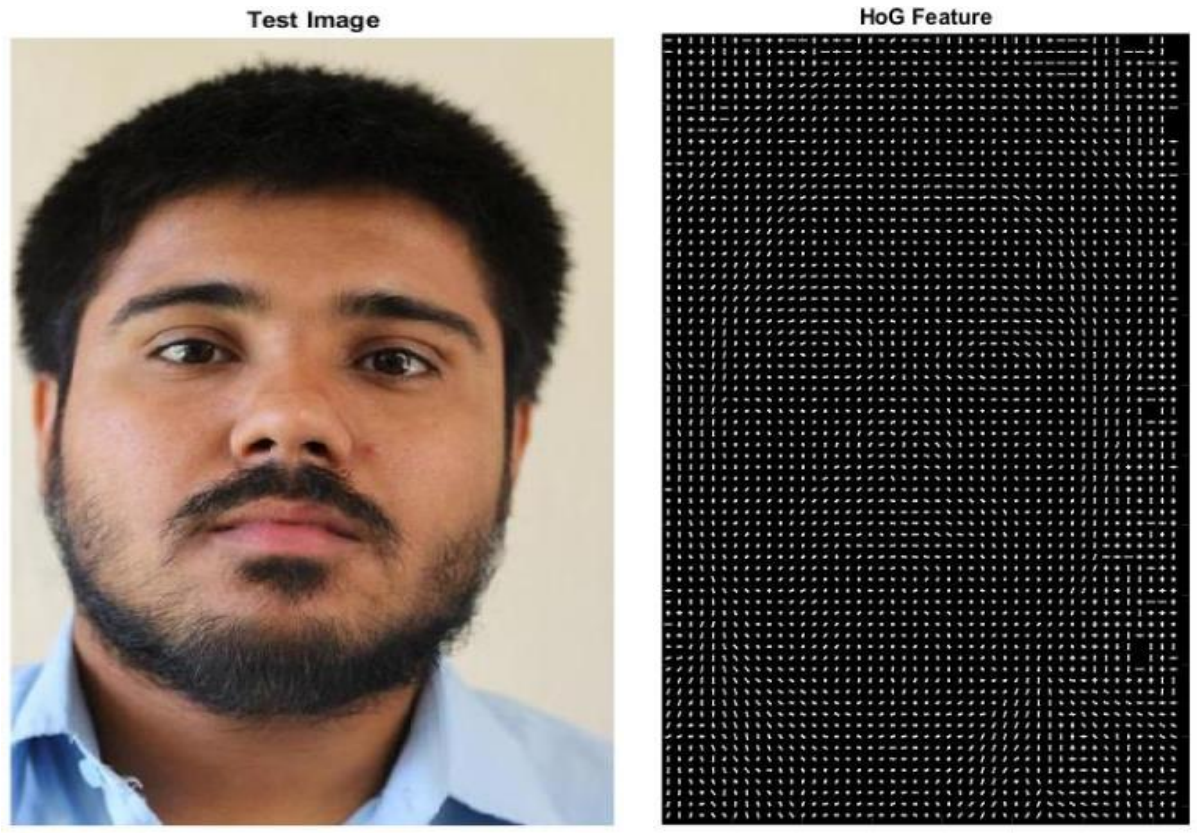


Fig 3.7 Visualizing Hog Features from Database

3.1.3 Learning or Modelling

After Extracting features then a classifier or model is created by machine learning algorithm by training the classifier from these extracted features. In simple words, on the basis these features (i.e. numbers) represented in the form of table where each set of features are passed along with their descriptor or labels (i.e. each person’s features are depicted as a class) as input to the machine learning algorithm for training which learns the regularities in the data and classifies different persons on the basis of these features. Later after creating the model we pass the features of Test Image for recognition to this classifier for predicting that these features belong to which class and hence person’s identity will be predicted. Prediction is discussed in section 3.2.2.

Machine learning algorithm here used is Error Correcting Output of Codes (ECOC) which classifies a multiclass problem as we are using more than two classes(i.e. persons) by reducing it to multiple binary classifiers such as “Support Vector Machines (SVM)”. To understand the multiclass ECOC classifiers we must first understand simple Binary Classification such as in support vector machines.

Binary classes mean we have exactly two classes. These classes are categorized in 2 categories.

1. Separable Data
2. Non-Separable Data

For separable data, Support Vector Machines rely on **Linear Classifiers**. Under this classification Support vectors are the points on the boundary of the classifier's margin and on the basis of these points, the classifier is created as shown in Fig 3.8.

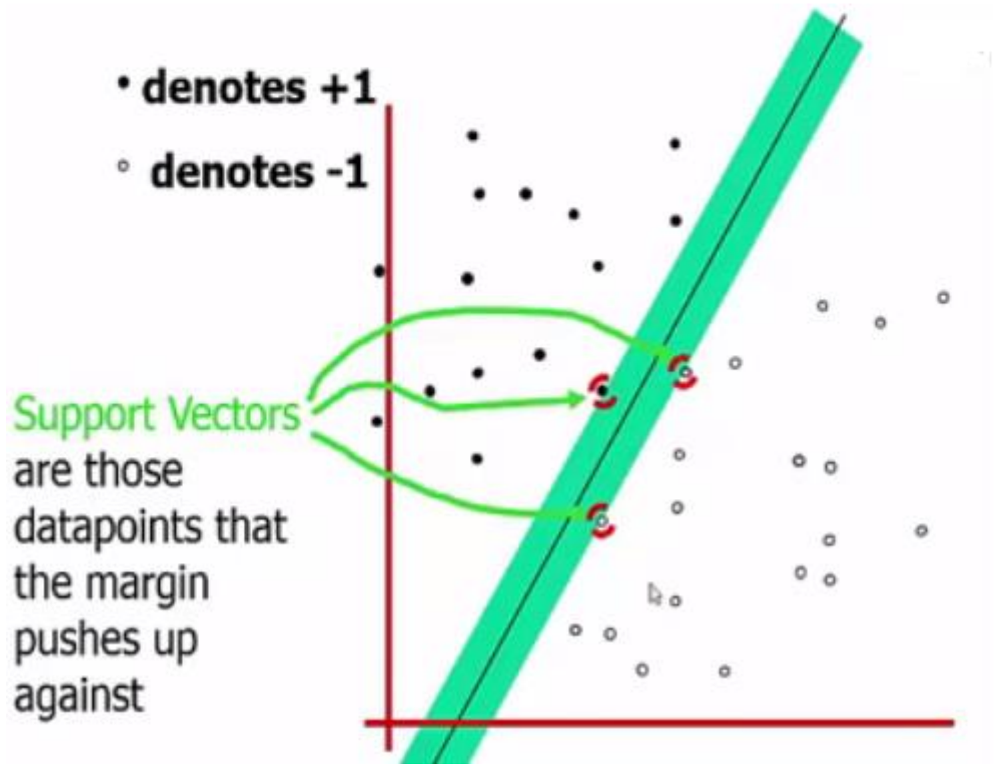
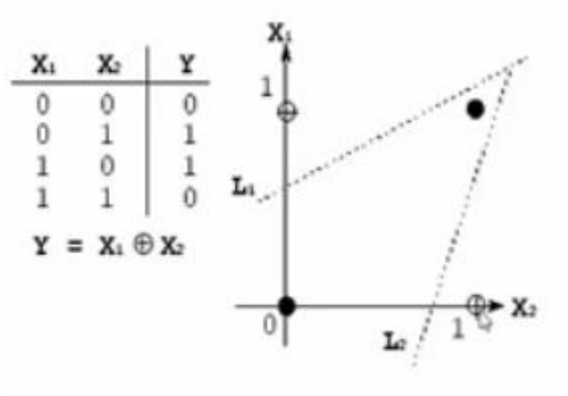

• denotes +1
∘ denotes -1
Support Vectors are those datapoints that the margin pushes up against

Fig.3.8 Linear Classification in SVM [17]

For non-separable data, Support Vector Machines rely on **Non-Linear Classifiers.** First, let us understand non-separable data. The simplest example of non-separable data is XOR Problem. As it can be seen from Fig.3.9 that no linear classifier can separate the points X_1 and X_2. Thus mapping is done differently for classifying these points as shown in Fig 3.10.



X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = X_1 \oplus X_2$

Fig. 3.9 XOR Problem.

Mapping (X_1, X_2) to (X_1, X_1X_2) for solving Xor Problem.

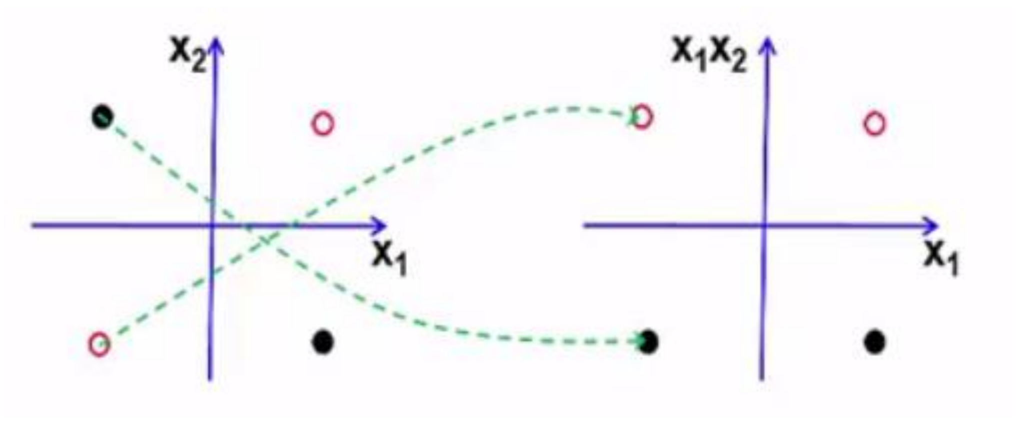


Fig. 3.10 Solving Xor Problem

A solution of every linearly non-separable data is not so simple as solving Xor problem so Hyperplane is used for classifying linearly non-separable data. The general idea is that an original input space can always be mapped onto a higher dimensional feature space where the data is linearly separable as shown in Fig 3.11 as

$$\Phi: x \longrightarrow \varphi(x)$$

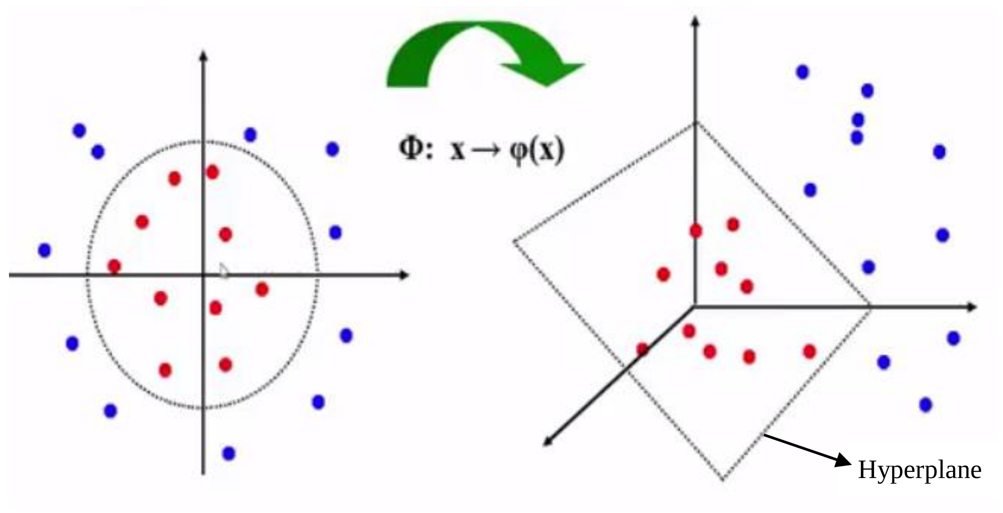


Fig. 3.11 Hyperplane for linearly non-separable data

As in the dataset, we have 7 persons thus multiclass class classification is used there are various methods available for multiclass classification like one-vs-all, one-vs-one reduction method, Naïve Bayes, Neural Networks, ECOC etc. As it is righteously said by George E.P. Box "All models are WRONG but some are USEFUL". Thus, ECOC is used as it gives better results for face recognition. ECOC is described as an ensemble method for multiclass class classification problem. ECOC combines many binary classifiers to solve multiclass problems. Table. 3.4 shows a 15-bit ECOC classifier with 10 classes.

Class	Code Word														
	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
0	1	1	0	0	0	0	1	0	1	0	0	1	1	0	1
1	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0
2	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1
3	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1
4	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1
5	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1
6	1	0	1	1	1	0	0	0	0	1	0	1	0	0	1
7	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1
8	1	1	0	1	0	1	1	0	0	1	0	0	0	1	1
9	0	1	1	1	0	0	0	0	1	0	1	0	0	1	1

Table. 3.4 ECOC Code Sample.

Each class is assigned a unique string called a Code Word. For each column one binary classifier is learned during training. For example in first column {0,2,4,6,8} is separated from {1,3,5,7,9} and similarly 15 classifiers are trained similarly to separate every class. For classification of a data point x as input, it is evaluated with all the 15 classifiers and a 15-bit string is generated finally, the class is chosen whose code word closest to x's output string as the predicted label.

As we can see from the above example that only 4 bits can distinguish the above classes but instead we have use 15 bits which are a lot more than required this is done for error correction. As some bits are redundant or error correcting bits which can tolerate error introduced due to finite training sample, poor choice of input features and flaws in training algorithm. As if the minimum hamming distance between the codes is d then code can correct at least floor value of $\left\lfloor \frac{d-1}{2} \right\rfloor$ single bit errors. As long as the errors are less than $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors can be corrected.

For the above mentioned code in Table 3.4 we can correct 3 errors out of 15 bits. Design of this coding matrix is of random type.

For the dataset, we have used the following Table 3.5 shows the parameters of the ECOC classification.

Number of Classes	7
Values inside coding matrix	-1,0,+1
Coding matrix dimensions	7x21
Number of classifiers trained	21
Number of inputs for training for each class	12x111780
No of inputs for prediction	111780

Table. 3.5 Details for ECOC classifier

Table 3.6 depicts the coding matrix of the ECOC classifier that has been created for identification of the person.

Class	**Code Word**																				
Abhinav	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Anmol	-1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Divyansh	0	-1	0	0	0	0	-1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
Rishav	0	0	-1	0	0	0	0	-1	0	0	0	-1	0	0	0	1	1	1	0	0	0
Shubham	0	0	0	-1	0	0	0	0	-1	0	0	0	-1	0	0	-1	0	0	1	1	0
Sudhanshu	0	0	0	0	-1	0	0	0	0	-1	0	0	0	-1	0	0	-1	0	-1	0	1
Tushar	0	0	0	0	0	-1	0	0	0	0	-1	0	0	0	-1	0	0	-1	0	-1	-1

Table 3.6 Coding matrix of the classifier

.

3.2 Face detection and Feature extraction of Test images

Using Voila-Jones algorithm [4] faces in the input frame are detected and these detected faces are saved in JPEG format after normalization i.e. converting each image in same dimensions as that of the existing database used for training the classifier.

3.2.1 Viola-Jones Algorithm for Detecting Faces

In MatLab vision.CascadeObjectDetector works on this algorithm for face detection. This algorithm basically involves four steps for detecting faces in an image.

- Haar Features
- Integral Image
- Adaboost
- Cascading

Haar Features used for face detection are of 3 types. Two-rectangle feature type (horizontal/vertical edge features) see Fig 3.12(a) & (b), three-rectangle feature (line features) type Fig 3.12(c) & (d), four-rectangle feature type Fig 3.12(e).

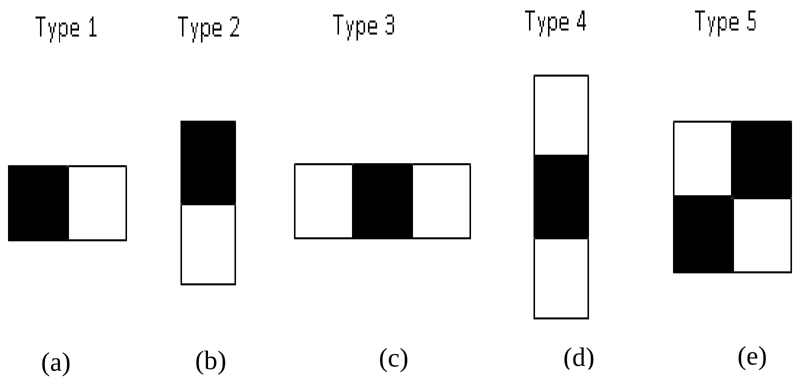


Fig 3.12 Haar Features

Haar features are used as convolution kernels where black part is assigned -1 and white part is assigned as +1. Utilizing a 24x24 pixel base detection window, with all the possible permutations of traversing horizontally and vertically and scaling we get a full set of features as 45,396 features which are too large for a 24x24 window in terms of computability. Thus we deal with this problem in later stages.

The Integral Image at location (x, y), in an image, is the sum of the pixel values above and to the left of (x, y), inclusive as shown in an example below in Fig 3.13.

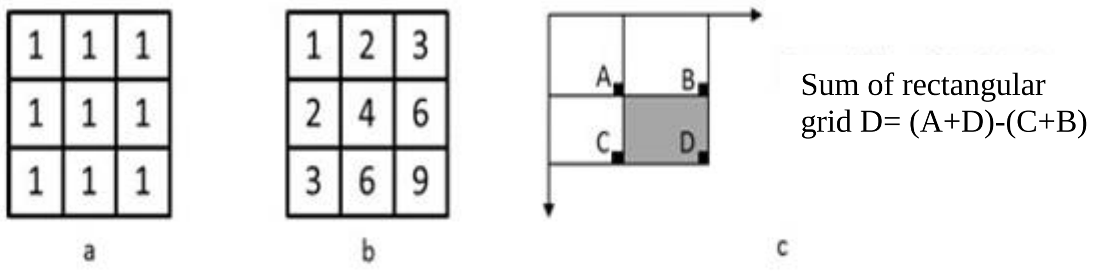


Fig 3.13 Sum of pixels in Integral Image **(a).** Original image **(b).** Image integral **(c).** Rectangular representation of image integral.

Integral image increases Computational efficiency as a single feature can be evaluated for any scale and at any location and it does not require a pyramid of scaling.

In **Adaboosting** discriminant features are found that are effective in detecting components of faces such as eye, nose etc. from Haar features and on the basis of these features we reduce the computability for a 24x24 window from 45,396 much lower by selecting these effective features. This is done by training through a large dataset and using machine learning to select these useful features. Adaboosting learning algorithm is given in Fig. 3.14.

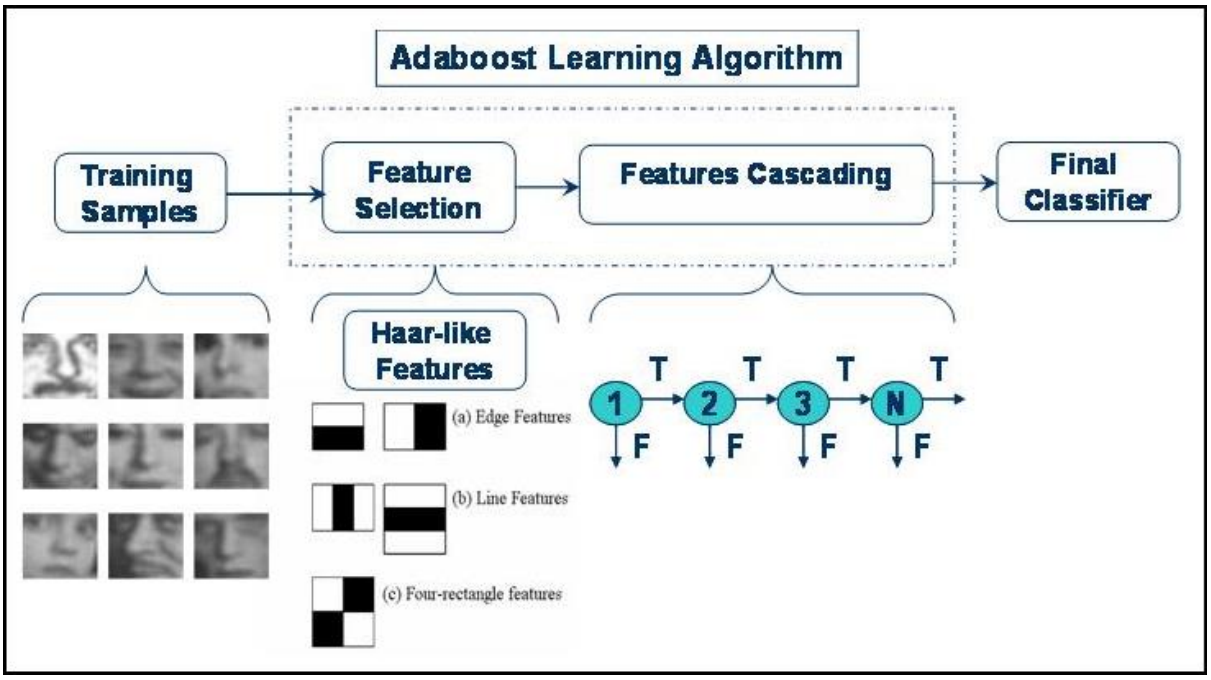


Fig 3.14 Adaboost Learning Algorithm.

Cascading is done with the help of decision trees this further enhances the computational ability. By moving one by one from one sub-window to another that if qualifies as whether a face is there it goes further for checking more of such sub-windows if the decision comes to be a no from a sub-window further checking is avoided as shown in Fig 3.10.

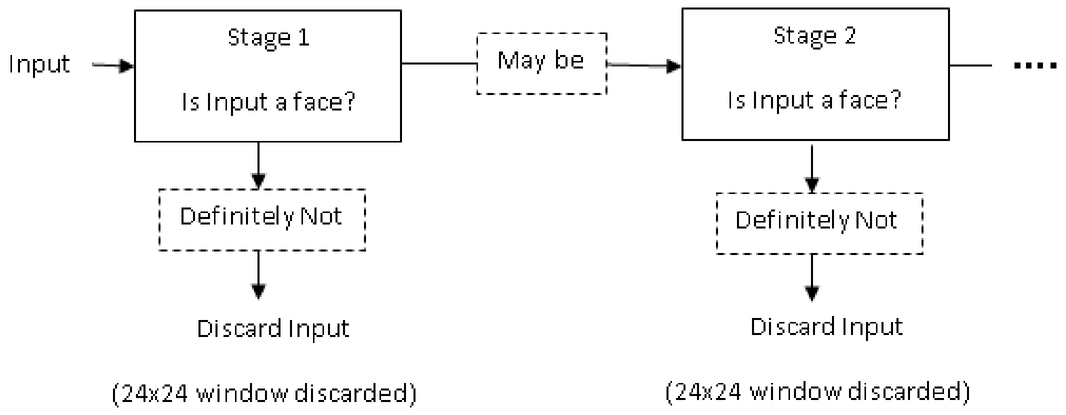


Fig 3.15. Cascading.

After faces are detected these faces are cropped out from the input frame in the form of image and are normalized to that of the images in the database that is constructed for the training of the classifier. These images are called Test Images which will further be used for recognition.

3.2.2 Feature Extraction, Recognition of Test Images and Display Attendance

Feature extraction is done similar to extracting HoG features as discussed in section 3.1.2 in detail of the Test Images. The Test Images for recognition are shown in Fig. 3.16.

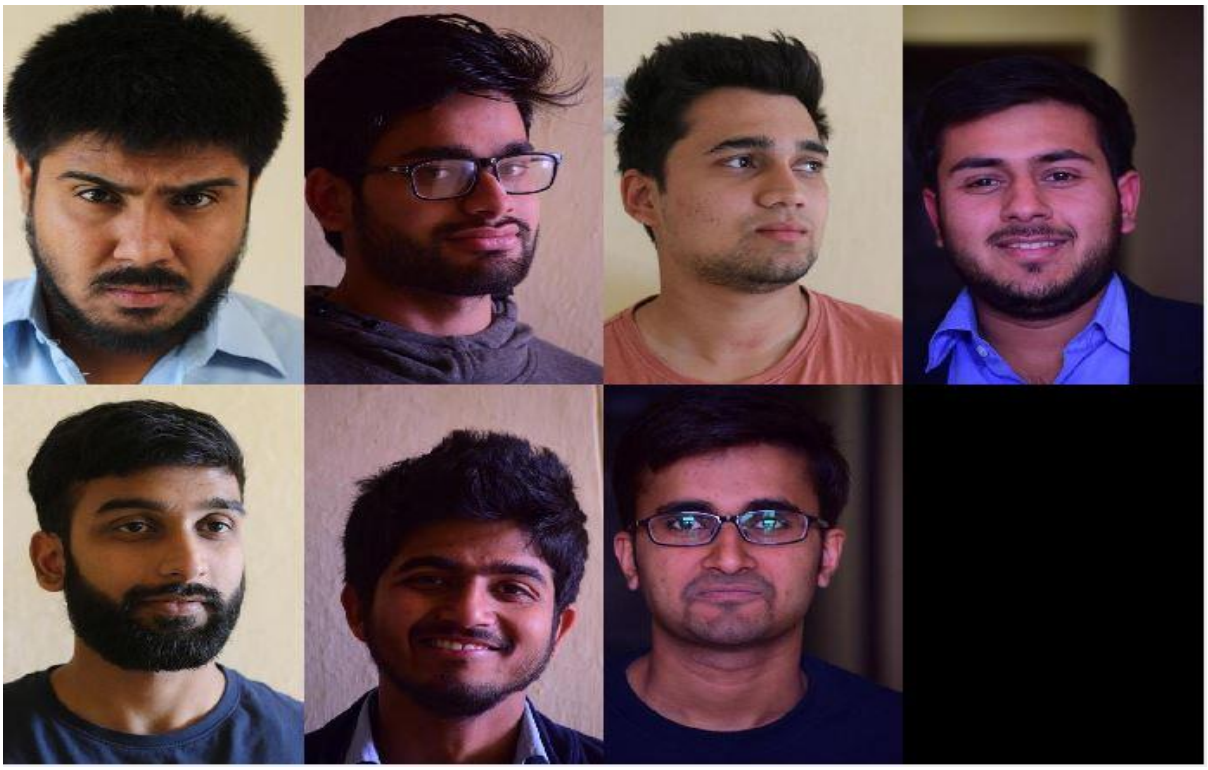
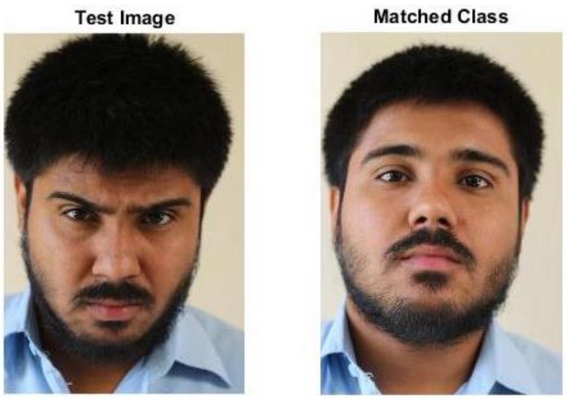
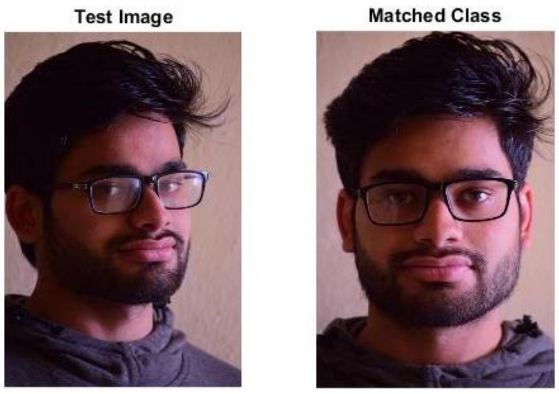


Fig.3.16 Test Images

These extracted features are passed to the trained Classifier and labels have been predicted [6] [7] [8] on the basis of these features as input to the Classifier. This prediction is done on the basis of minimizing the Kullback-Leibler divergence to estimate class posterior probabilities. Explaining mathematics of these prediction functions is beyond the scope of this thesis. But in layman terms, it can be understood that Kullback-Leibler gives a distance vector (in simple words a number) which does not satisfy triangle law but is used to calculate the posterior probability of the classes. The class which has the most probability is estimated as the predicted class and its label is returned. After labels have been predicted, this label has been mapped back to the training set and compared to the Database by the descriptor and compared with them and checked whether the predictor has predicted right by displaying the images of Test Image and Matched Class(from the database) side-by-side as shown in the Fig. 3.17.


Test Image
Matched Class

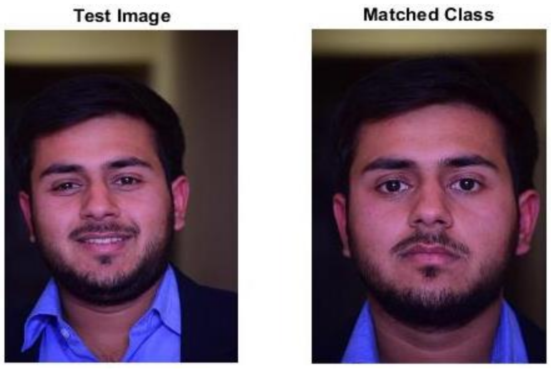
(a)


Test Image
Matched Class

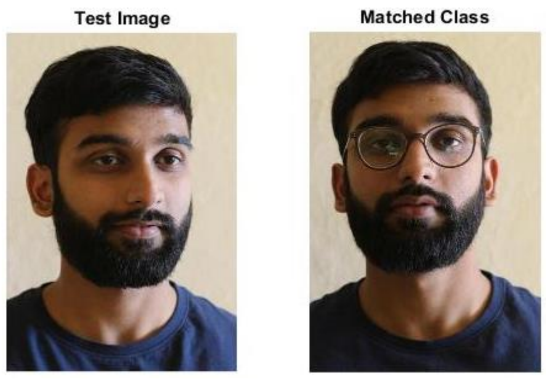
(b)

Test Image
Matched Class

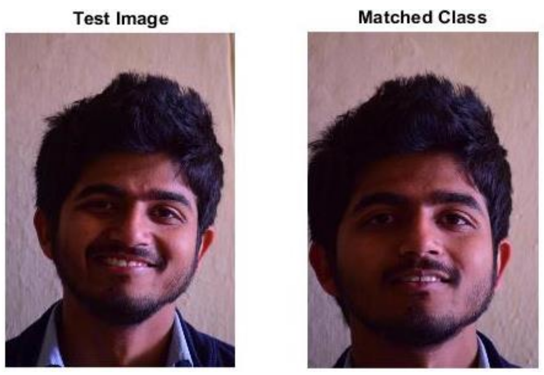
(c)


Test Image
Matched Class

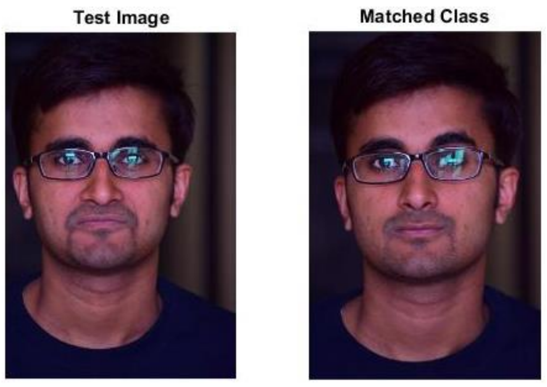
(d)



(e)



(f)

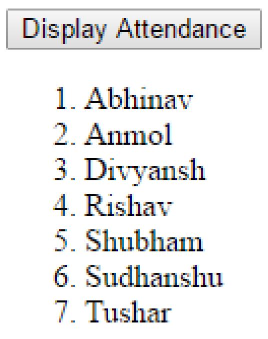


(g)

Fig 3.17 Checking Test Image with Matched Class

As it has been observed that all the images have been matched correctly thus completing the recognition part.

After recognition, the labels have been written in a text file with each label separated by a comma. This text file is displayed on a web page by clicking on the display attendance tab thus giving the names of the people identified from the text file. Hence showing their attendance as shown in Fig 3.18.



1. Abhinav
2. Anmol
3. Divyansh
4. Rishav
5. Shubham
6. Sudhanshu
7. Tushar

Fig 3.18 Displaying attendance on the web page

The code for chapter 3 has been put in an appendix.

4. Conclusion

In the project **"Facial Recognition and Automatic Attendance System",** the objective of taking attendance with a photograph has been accomplished for a class of 7 people. This has been done by detecting, cropping and normalizing faces in the form of images from an input frame. Then extracting its HOG Features which are given as input to the classifier and then recognize these faces by predicting from this classifier. The classifier has been created by training it from 12 images of each person for all 7 persons via extracting HOG Features of each image. Finally, the people who have been identified are displayed with their names, hence marking their attendance on the web page.

In this project due to the high dimensionality of HOG Features, Facial Recognition has been possible. Its precision for facial recognition can be improved by changing features extraction method from HOG to Dense SIFT (Scale Invariant Feature Transform) [9].

Also, its precision (accuracy of the system to identify faces for a larger database) can be improved if the size of the training set for training the classifier can be improved by increasing the number of images in training set for each person.

As the number of people to be identified is just 7 thus HOG features have worked out well in this case recognizing all 7 persons correctly in this Project.

Appendix

HTML CODE:

```
<html>
<head>
<title>Image Capture via pubnub</title>
<script src="jquery.js"></script>
</head>

<body>
<h1 style="text-align:center;color:red;">FINAL YEAR PROJECT</h1>
<h3 style="text-align:center;"> Hello.</h3>

<p style="text-align:center;">Click on <mark> CAPTURE </mark> to click image.</p>
<form style='text-align:center;' action='image_capture.php' method='post'>
<input name='msg' type='hidden' value='capture'/>
<input type="submit" value="CAPTURE"/>

</form>
<button onclick="displayNames();">Display Attendance</button>
<ol id="list">

</ol>

<label>
</body>
</html>

<script>
var displayNames = function(){

        $.get('image_database/list.txt', function(data) {
```

```
                    var listArray = data.split(',');
                    for(var l = 0; l < listArray.length - 1; l++){
                              $('#list').append('<li>' + listArray[l] + '</li>');
                    }

          });

}
</script>
```

PHP CODE:

```
<?php
require_once('libpubnub/autoloader.php');
use Pubnub\pubnub;
$config=['publish_key' => 'pub-c-ad35a3b1-81f2-4748-8806-3f47b441cf5a',
                     'subscribe_key' => 'sub-c-62181a90-4062-11e6-85a4-0619f8945a4f'
                     ];
$pubnub = new Pubnub($config);
$message = $_POST['msg'];
$pubnub -> publish("channel1",["action"=>"$message"]);
?>
```

PYTHON CODE:

```
from pubnub import Pubnub
from time import sleep
import picamera
import json

camera = picamera.PiCamera()
```

```
pubnub =Pubnub (publish_key = "pub-c-ad35a3b1-81f2-4748-8806-3f47b441cf5a",
            subscribe_key = "sub-c-62181a90-4062-11e6-85a4-0619f8945a4f")

def callback(message):
   print(message)
def _callback(message, channel):
   print(message)
   kk = json.dumps(message)
   kk=json.loads(kk)
   print(kk['action'])
   if(kk['action'] == 'capture'):
            camera.capture('projectimage.jpg')
      print('image successfully captured')

def _error(message):
   print(message)

pubnub.subscribe(channels="channel1", callback=_callback)
```

MATLAB CODE:

```
%% Importing images for training from database
close all;
clear all;
clc;
faceDatabase = imageSet('TempDatabase','recursive');

%% Extracting HOG Features for training set
trainingFeatures = zeros(size(faceDatabase,2)*faceDatabase(1).Count,111780);
featureCount = 1;
for i=1:size(faceDatabase,2)
   for j = 1:faceDatabase(i).Count
      trainingFeatures(featureCount,:) = extractHOGFeatures(read(faceDatabase(i),j));
```

```
        trainingLabel{featureCount} = faceDatabase(i).Description;
        featureCount = featureCount + 1;
    end
    personIndex{i} = faceDatabase(i).Description;
end

%% Create 7 class classifier using fitcecoc
faceClassifier = fitcecoc(trainingFeatures,trainingLabel);

%% person to detect from raspberry pi
queryFace = imread('projectimage3.jpg');
imshow(queryFace);
title('Query Image');

%% Creating a detector object
faceDetector = vision.CascadeObjectDetector;
%Detect Faces
bboxes = step(faceDetector, queryFace);



%% Annotation of detected faces
queryFaceDetected = insertObjectAnnotation(queryFace, 'rectangle', bboxes, 'Face');
figure, imshow(queryFaceDetected), title('Detected face');



%% cropping and normalizing of extracted Faces
 for i = 1:size(bboxes,1)
 J= imcrop(queryFace,bboxes(i,:));
 scaleFactor = 2000/size(queryFace,1);
 J = imresize(J,scaleFactor);
 figure;imshow(J); title('normalized face')
 imsave
 end
```

```
%% Test Images from Test Set and extracting its HOG Features
Testimage = imageSet('TestImages');
displayimage=1;
fileID= fopen('list.txt','w');
fclose(fileID);
for person=1:Testimage(1).Count %use Testimage(1).Count intead of 7
   queryImage = read(Testimage,person);
   queryFeatures = extractHOGFeatures(queryImage);
   personLabel = predict(faceClassifier,queryFeatures);
   % Mapping back to the training for identification and displaying image
   booleanIndex = strcmp(personLabel, personIndex);
   integerIndex = find(booleanIndex);
   figure;
   subplot(1,2,displayimage);imshow(queryImage);title('Query Face');
   subplot(1,2,displayimage+1);imshow(read(faceDatabase(integerIndex),1));title('Matched
Class');
   displayimage=1;
   % Generating a text file to check who is present
   A = personLabel(1:1);
   Y = A{:};
   fileID= fopen('list.txt','a');
   fprintf(fileID,'%s,',Y);
   fclose(fileID);
end
```

REFERENCES

[1] Dalal, N. & B. Triggs. (June 2005). "Histograms of Oriented Gradients for Human Detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 886–893, San Diego, 20-25 June 2005, California, IEEE Publishing.

[2] Thomas G. Dietterich & Ghulum Bakiri. (1995). "Solving Multiclass Learning Problems via Error-Correcting Output Codes", *Journal of Artificial Intelligence,* Vol. 2, pp. 263-286.

[3] Nobuhiko Yamaguchi & Naohiro Ishii. (2002). "Constructing Error Correcting Output Coding Classifiers", *Proceedings of the 9th International Conference on Neural Information Processing* (ICONIP'OZ), Vol. 5, pp. 2635 – 2639, Singapore, 18-22 Nov. 2002, Singapore, IEEE Publishing.

[4] Viola Paul & Michael J. Jones. (2001). "Rapid Object Detection using a Boosted Cascade of Simple Features", *Proceedings of the 2001 IEEE computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1, pp.511–518, Kauai, 8-14 Dec. 2001, Hawaii, IEEE Publishing.

[5] Escalera, S., O. Pujol, & P. Radeva. (2008). "Separability of ternary codes for sparse designs of error-correcting output codes." *International Conference on Pattern Recognition*, Tampa, 8 - 11 Dec. 2008, Florida, IEEE Publishing.

[6] Allwein, E., R. Schapire, & Y. Singer. (2000). "Reducing multiclass to binary: A unifying approach for margin classifiers." *Journal of Machine Learning Research*, Vol. 1, pp. 113–141.

[7] Wu, T. F., C. J. Lin, & R. Weng. (2004). "Probability Estimates for Multi-Class Classification by Pairwise Coupling." *Journal of Machine Learning Research*. Vol. 5, pp. 975–1005.

[8] Zadrozny, B. (2001). "Reducing Multiclass to Binary by Coupling Probability Estimates." *NIPS: Proceedings of Advances in Neural Information Processing Systems.* Vol.14, pp. 1041–1048, Vancouver, 3 – 8 Dec 2001, British Columbia.

[9] Karen Simonyan, Omkar M. Parkhi, Andrea Vedaldi & Andrew Zisserman. (2013) "Fisher Vector Faces in the Wild." *24th British Machine Vision Conference,* University of Bristol, 9 - 13 Sept. 2013, Bristol.

[10] Jyelton (2014, July 23). Calculating Capacitance for a power supply circuit. Retrieved from https://electronics.stackexchange.com/questions/122413/how-to-calculate-the-values-of-capacitors-for-5v-dc-power-supply.

[11] Eben Upton (2016, April 25). NEW 8-MEGAPIXEL CAMERA BOARD ON SALE AT $25. Retrieved from https://www.raspberrypi.org/blog/new-8-megapixel-camera-board-sale-25/

[12] Canon Eos 700d EOS Digital SLR and Compact System Cameras (2014) Retrieved from http://www.canon.co.uk/for_home/product_finder/cameras/digital_slr/eos_700d/

[13] Mr. Excuisite, (2017, March 20). Selling of Raspberry pi. Retrieved from http://www.journaldugeek.com/2017/03/20/plus-de-125-millions-de-raspberry-pi-ont-ete-vendu-soit-presque-autant-que-le-commodore-64/

[14] Boxed processor Intel Core i5. Retrieved from http://www.conrad.com/ce/en/product/1226048/Boxed-processor-Intel-Core-i5-i5-4690K-4-x-35-GHz-Quad-Core-PC-base-Intel-1150-88-W

[15] Karamjeet (2016, November 5). finalproject app created on pubnub. Retrieved from https://admin.pubnub.com/#/user/286606/account/286606/app/311786/key/198711/console

[16] Avi Nehemiah (2014, November 13). Face Recognition with MATLAB. Retrieved from https://www.mathworks.com/videos/face-recognition-with-matlab-98076.html

[17] Dr. Mubarak Shah, UCF CRCV (2012, September 19). Retrieved from https://www.youtube.com/watch?v=715uLCHt4jE&list=PLd3hlSJsX_ImKP68wfKZJVIPTd8Ie5u-9 %computer vision lectures

[18] Chris McCormick (2013, May 9). Hog Person Detector Tutorial. Retrieved from http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
LEARNING RESOURCE CENTER

PLAGIARISM VERIFICATION REPORT

Date: 2/5/2017

Type of Document (Tick): Thesis | M.Tech Dissertation/ Report | [✓ B.Tech Project Report] | Paper

Name: KARAMJEET SINGH BAKSHI Department: ECE

Enrolment No. 131038 Registration No. ____

Phone No. 9816927846 Email ID. karamjeetsinghbakshi007@gmail.com

Name of the Supervisor: Salman Raju Talluri

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): FACIAL RECOGNITION AND AUTOMATIC ATTENDANCE SYSTEM

Kindly allow me to avail Turnitin software report for the document mentioned above.

[illegible]
(Signature)

FOR ACCOUNTS DEPARTMENT:

Amount deposited: Rs. 300 Dated: 2/5/2017 Receipt No. CRV1705/12
(Enclosed payment slip)

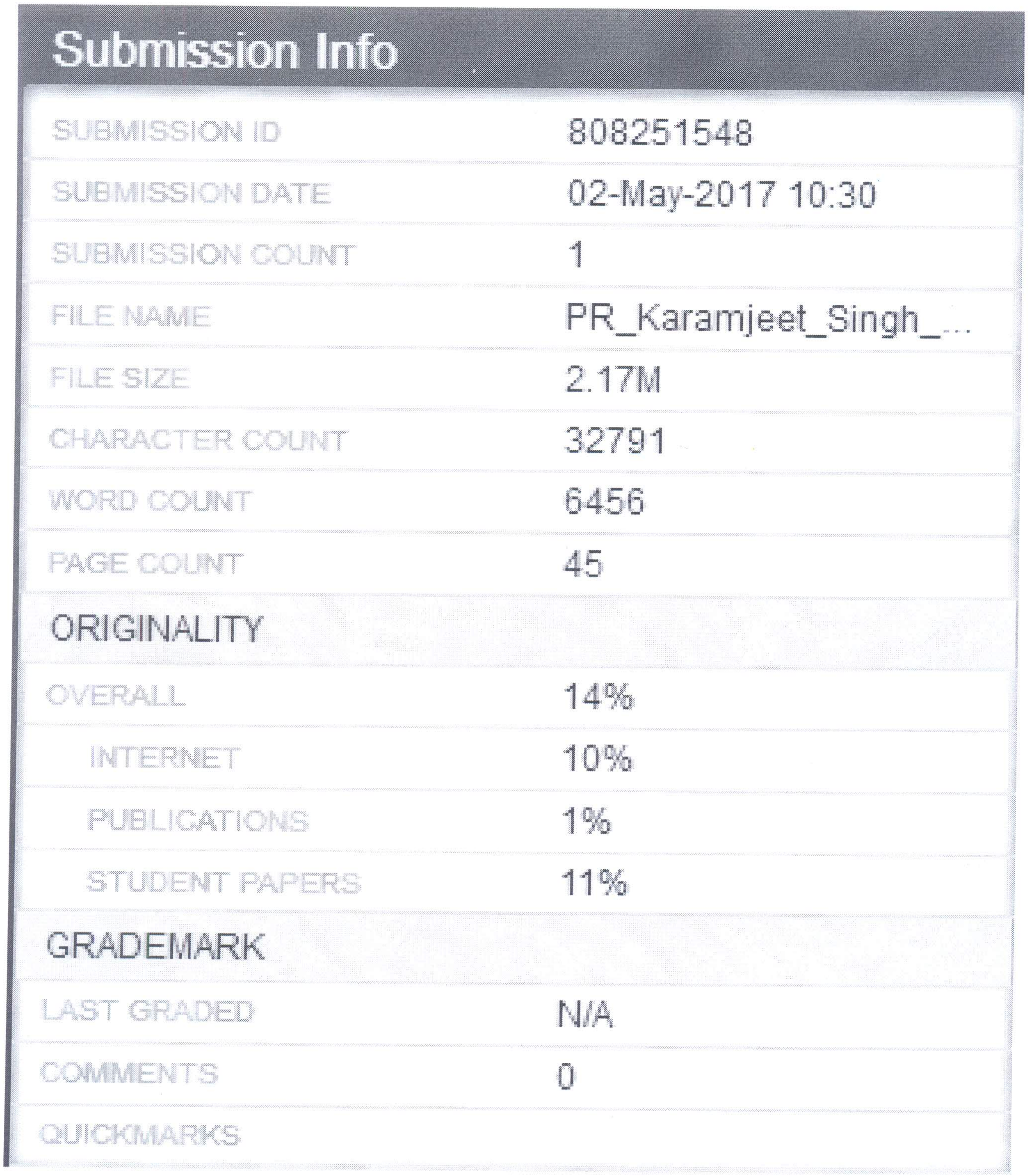
[illegible]
(Account Officer)

FOR LRC USE:
The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Report delivered on	Similarity Index in %	Submission Details	
9:00 AM 2/5/2017	10:00 AM 2/5/2017	14%	Word Counts	6456
			Character Counts	32791
			Page counts	45
			File Size	2.17M

[illegible]
Checked by
Name & Signature

[illegible]
Librarian



Submission Info

SUBMISSION ID	808251548
SUBMISSION DATE	02-May-2017 10:30
SUBMISSION COUNT	1
FILE NAME	PR_Karamjeet_Singh_...
FILE SIZE	2.17M
CHARACTER COUNT	32791
WORD COUNT	6456
PAGE COUNT	45
ORIGINALITY	
OVERALL	14%
INTERNET	10%
PUBLICATIONS	1%
STUDENT PAPERS	11%
GRADEMARK	
LAST GRADED	N/A
COMMENTS	0
QUICKMARKS	

[illegible]