**FACE RECOGNITION**

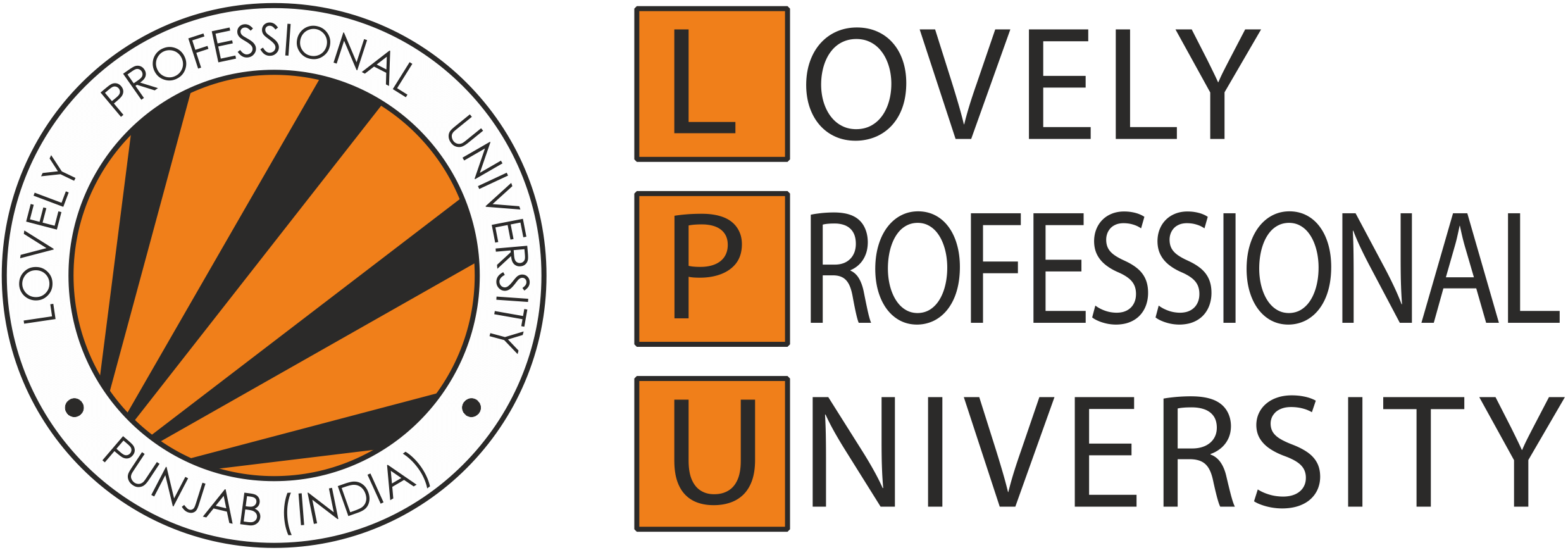
END TERM REPORT

*By*

Nosang, Mohit, Sourabh ad Vinay

(Section: K18JF)

(Roll Number(s): \_\_\_)



**Department of Intelligent Systems**

**School of Computer Science Engineering**

**Lovely Professional University, Jalandhar**

April, 2020

**Student Declaration**

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

<>

Mohit Ranjan

<>

<>

Sourabh Gautam

<>

Nosang Subba

<>

<>

Vinay Kumar

<>

<>

Place

Date

BONAFIDE CERTIFICATE

Certified that this project report “………. TITLE OF THE PROJECT……………..” is the Bonafide work of “…………..NAME OF THE CANDIDATE(S).…………” who carried out the project work under my supervision.

<>

<>

<>

<>

<>

**TABLE OF CONTENTS**

**TITLE PAGE NO.**

**1. Introduction ……………………………………………………................. 5**

**1.1 Idea....................................................................................6**

**1.1.1 Using Haar Cascade………….............................................. 3**

**1.1.2 Using Dlib(C++)................................................................5**

**2 About the project…............................................................. 7**

**2.1 Motivation…..................................................................... 8**

**2.2 Description........................................................................... 9**

**2.3 Data Flow diagram……………………………………………………………. 10**

**2.4 Implementation……………………………………………………………….. 11**

**2.5 SWOT Analysis ………………………………………………………………..12**

**INTRODUCTION**

**Idea**

We wanted to create a simple Face recognition that detects the face and displays the detail of the person such as name. We had to implement an algorithm to match faces in our database for this to happen.

To implement it we had to research about various methods to create such application. There were a few ways by which we could do this.

* **Using Haar cascade:** Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection using a Boosted Cascade of Simple Features. Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.
* **Using Dlib (C++ library):** Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high-performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.

To ease up our workload and complexity we chose the first method i.e haar cascade method. Most of the resources on internet are with this method only. So, to complete and understand this project we had to choose first method.

**About Project**

The project is about “Face Recognition System” using OpenCV (Open Source Computer Vision Library) in Python, Haar Cascade for object detection. LBPH (Local Binary Pattern Histogram) algorithm in python. There is a driver program “video.exe” which redirects the user to the camera hardware of the system. It will detect the face in the camera and if the face data is inside the database, it will display the name of the person above the frame as in the database. It captures and crops the image and matches it using LBPH algorithm on real time and displays the name of person in each frame.

The project is close to deep learning and we were supposed to learn the basics of deep learning and the advanced level of python. The project face recognition has helped us to learn about various libraries that are being used to create different type of input programs, we were able to learn about the basic file handling and encryption. This project has helped us to explore the ideas behind various AI and how their algorithms were thought. We were able to understand the logic behind China’s world’s best surveillance system.

The best thing we learned from this project was the teamwork and here is how our work was divided.

|  |  |
| --- | --- |
| Mohit Ranjan  11808170  Roll no. 45 | Created the report file and helped Nosang in finding resources. Wrote the driver code along with Sourabh. |
| Nosang Subba  11808163  Roll no. 60 | Collected the resources, Wrote the training program for the AI and helped Mohit in Report. Also finalised all the programs. |
| Sourabh Gautam  11808142  Roll no. 42 | Helped Mohit in report and wrote code for driver program along with Mohit. Helped Nosang with tutorial videos. |
| Vinay Kumar  Roll no. 44 | Helped others |

**Motivation**

Creating an AI is one of the most productive tasks these days. From SIRI to self-driving cars, the production of AI is increasing rapidly. Even in our smartphone we use lot of Artificial intelligence like optical text recognition, Face recognition, AI filters in camera, etc. Today in the era of gargantuan amount of data that is generated by both human and machines, outpaces the humans’ ability to contain, absorb, or interpret and make complex decisions based on the data. For example, most human can figure out the way not to lose in a game of tic-tac toe even if there are 255,168 moves. An AI is efficient at calculating all those moves and arrive at best decision.

An AI can be used to detect fraud in credit card transactions or finding a registered criminal from CCTV footage if there are too many faces and less time to analyse the faces and match them to the criminal database. It can also be used at the traffic cams to detect the vehicle number plate and find details from the transportation database. For example, China has developed an AI for mass surveillance. It is the largest monitoring network in the world. The government implemented this mechanism of control that encompassed the entire nation and its people. This surveillance system is the four times the whole US government camera system. It has the face data of all the registered people of china and even of those people who come to visit China on their passport. 626 million cameras at every square and road detect about half a million faces every month. They even released a real time citizen tracking mobile application.

Learning from their success our goal is to learn and create such or better surveillance system for our country.

**Description**

**Modules:**

The code is too short to be modularised, but we have made it in a way that it can be understood easily.

Training:

We have used LBPH (Local Binary Pattern Histograms) algorithms for training purpose, which is provided by the Open CV library, to do so we need to use data set with facial images of the people we want the program to recognize. We need to also set an ID for each image, so the algorithm can recognize the face in the image and give a suitable output. The LPBH algorithm creates an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. Now , using the generated image the algorithm uses grid x and grid y to divide the image into multiple grids, then the histogram of each region is extracted then these histograms are concatenated to form a bigger histogram, this concatenated or final histogram represents the characteristics of the original image. The data is stored in a YAML file.

Face recognition:

For this step the algorithm should be initially trained and the YAML file is imported as it contains all the data. Input image is provided, and we perform the LPBH algorithm on the new image and create a histogram of this image and it is then compared to the histogram stored the dataset and histogram with the closest match is returned with the name.

**Data Flow diagram:**

Cascade classifier

Use LBHP recognizer to recognize the face and share in file

Share the labels of image in pickle file

Detect faces from the NumPy array

Share the greyscale images in the form of NumPy array

Resize the text

Show label

Face recognized

Use the LPBH recognizer

Cascade classifier

Give labels and IDs to all the images present in the directory

Convert it into greyscale images

Face recognition training

Make a rectangle around faces

Face Detected

Detect faces

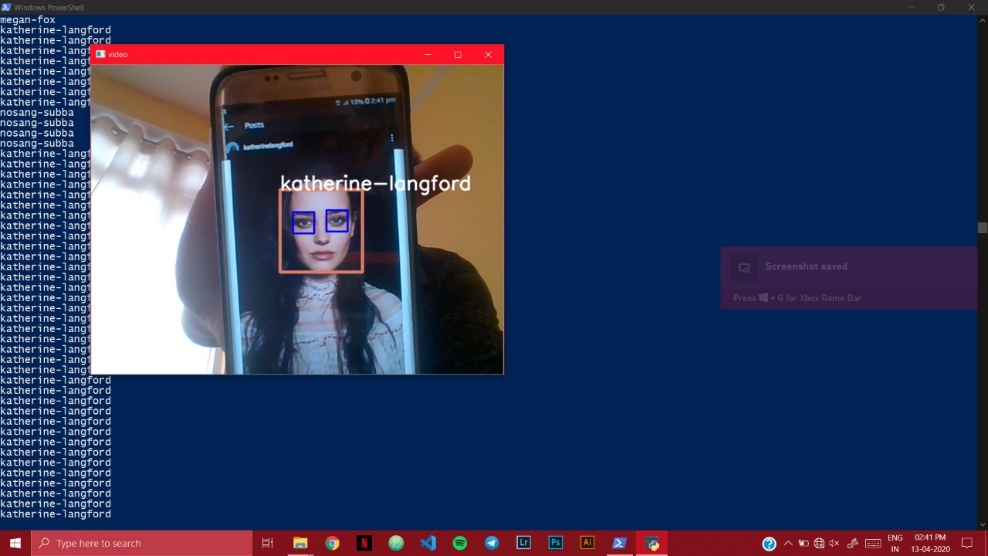
Convert colour images to greyscale

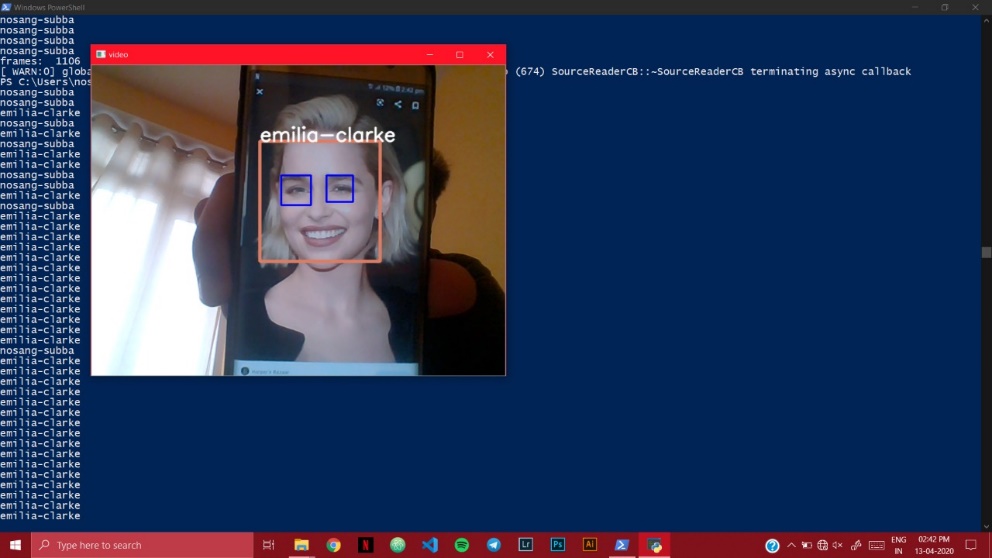
Open camera and acquire images from it

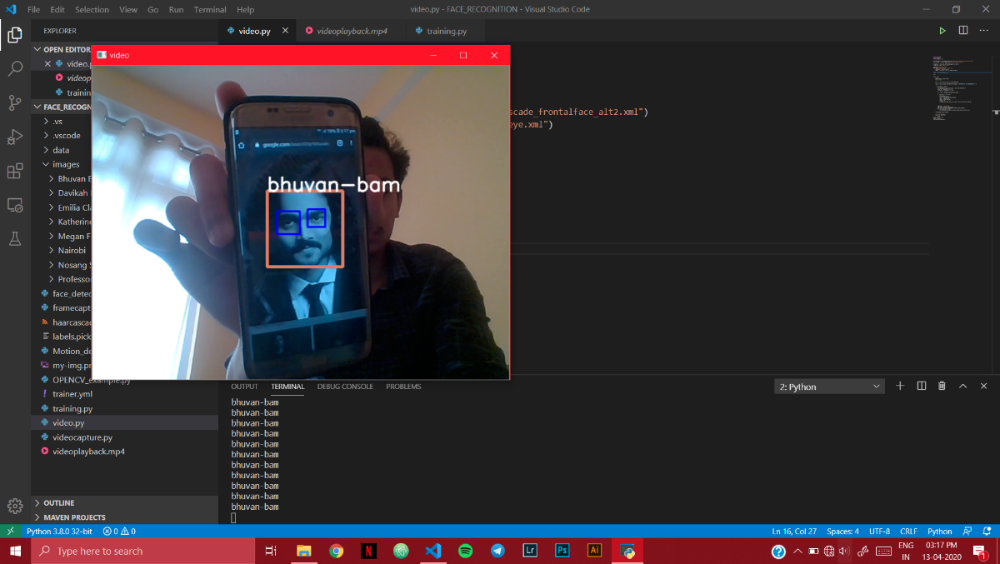
**Implementation**

Our expectations were almost similar to our results, although we faced some accuracy issues since we weren’t using deep learning method.

After running the “video.py” driver code, we had the following response:







**SWOT Analysis**

**Strength**

* Code is precise and short
* Code is easy to understand
* Complexity of code is lesser than other methods used.
* Can recognize various faces kept in database.

**Weakness**

* Can’t recognize faces precisely
* Poor user interface
* Sometimes confuses nose with eyes
* If there’s no face data of the user, It will identify user as some other face in database

**Opportunities**

* Can use this software at zoo entrance
* Can be used as auto sanitiser spray.
* A face detector in camera .
* Augmented reality.

**Threats**

* This can be replaced by an ultrasonic sensor.
* Not secure, can be risky to use at bigger levels
* This AI simply works on database, cannot learn (for example it can’t)
* Not easy to manage the database.

So, basically this particular type of face recognition system can be used at smaller level where only human entry is needed and there’s no big risk. Since this program can detect faces from any type of optical source, it can detect faces from a photograph too, making it the least secure security device. But there are many non-risky things like sanitiser spray for corona virus pandemic can be used to automatically spray sanitizers slowly from the mechanical nozzle after detecting a human face at screen. It is a low budget miracle for small things.