FACE RECOGNITION USING OPEN COMPUTER VISION AND MINI SMART PHONE

Group Members-

1-Vivek Kumar Suvam (1901500100113)
2-Vinayak Pandey (1901500100112)
3-Nishit Bansal (190150010062)
4-Ritik Rathore (190150010080)

Section-B|Semester-3|Year-II

Abstract

Identifying a person with an image has been popularised through the mass media. However, it is less robust to fingerprint or retina scanning. This report describes the face detection and recognition mini-project undertaken for the visual perception and autonomy module at iilm-cet. It reports the technologies available in the Open-Computer-Vision (Open CV) library and methodology to implement them using Python. For face detection, Haar-Cascades were used and for face recognition Eigenfaces, Fisherfaces and Local binary pattern histograms were used. The methodology is described including flow charts for each stage of the system

The mini smartphone lets user, perform various tasks like booking a room in a hotel, booking a taxi, perform mathematical operations using a calculator, play rock paper scissors with the computer and lastly, view calendar of any year.
we've used a randomize function in hotel, taxi and rock paper scissors programs to assign room no. , driver's name, car's number, driver's mobile number and computer's moves in the game.
we've used nested switch case in this whole program to create menu within a menu.
we've also used function prototyping in the program to make it easy for me and the viewer to understand the code.
we've used the Object oriented programming (OOPS) to make the code shorter and easy to understand.

Introduction

The following document is a report on the mini project for visual perception and autonomy. It involved building a system for face detection and face recognition using several classifiers available in the open computer vision library (Open CV). Face recognition is a non-invasive identification system and faster than other systems since multiple faces can be analysed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision "whose face is it?" , using an image database. In this project both are accomplished using different techniques and are described below. The report begins with the mentioning of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described.

Face Detection using Haar-Cascades

A Haar wavelet is a mathematical function that produces square-shaped waves with a beginning and an end and used to create box shaped patterns to recognise signals with sudden transformations. An example is shown in figure-1. By combining several wavelets, a cascade can be created that can identify edges, lines and circles with different colour intensities. These sets are used in Viola Jones face detection technique in 2001 and since then more patterns are introduced [10] for object detection as shown in figure-1.

To analyse an image using Haar cascades, a scale is selected smaller than the target image. It is then placed on the image, and the average of the values of pixels in each section is taken. If the difference between two values pass a given threshold, it is considered a match. Face detection on a human face is performed by matching a combination of different Haar-like-features. For example, forehead, eyebrows and eyes contrast as well as the nose with eyes as shown below in figure a single classifier is not accurate enough. Several classifiers are combined as to provide an accurate face detection system as shown in the block diagram below in figure-3.

Figure 1: A Haar wavelet and resulting Haar-like features.

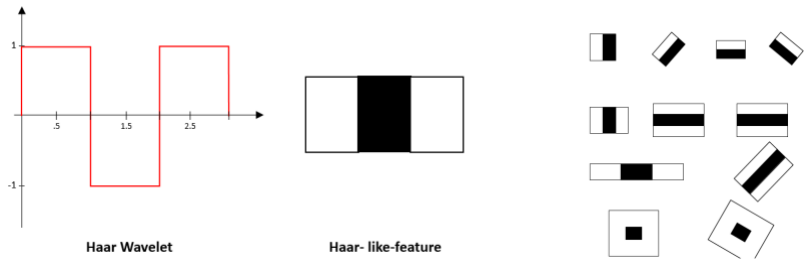
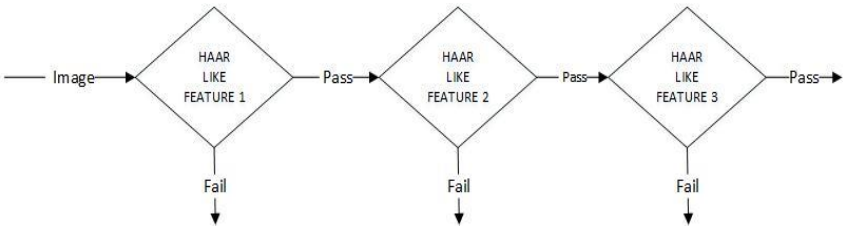


Figure 2: Several Haar-like-features matched to the features of face.

Figure 3: Haar-cascade of flow chart



Face Recognition

The following sections describe the face recognition algorithms Eigenface, Fisherface, Local binary pattern histogram and how they are implemented in Open CV.

1 Eigenface

2 Fisherface

3 Local Binary Pattern Histogram

Methodology

Below are the methodology and descriptions of the applications used for data gathering, face detection, training and face recognition. The project was coded in Spyder IDEs.

Face Detection

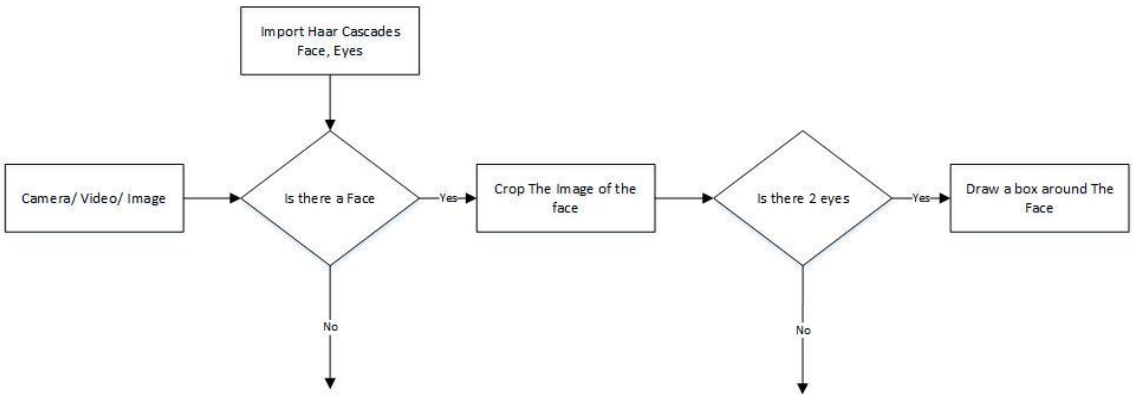


Figure 8: The Flow chart of the face detection application

Face Recognition Process

1. Collecting images IDs
2. Extracting unique features, classifying them and storing in XML les
3. Matching features of an input image to the features in the saved XML les and predict identity

Collecting the image data

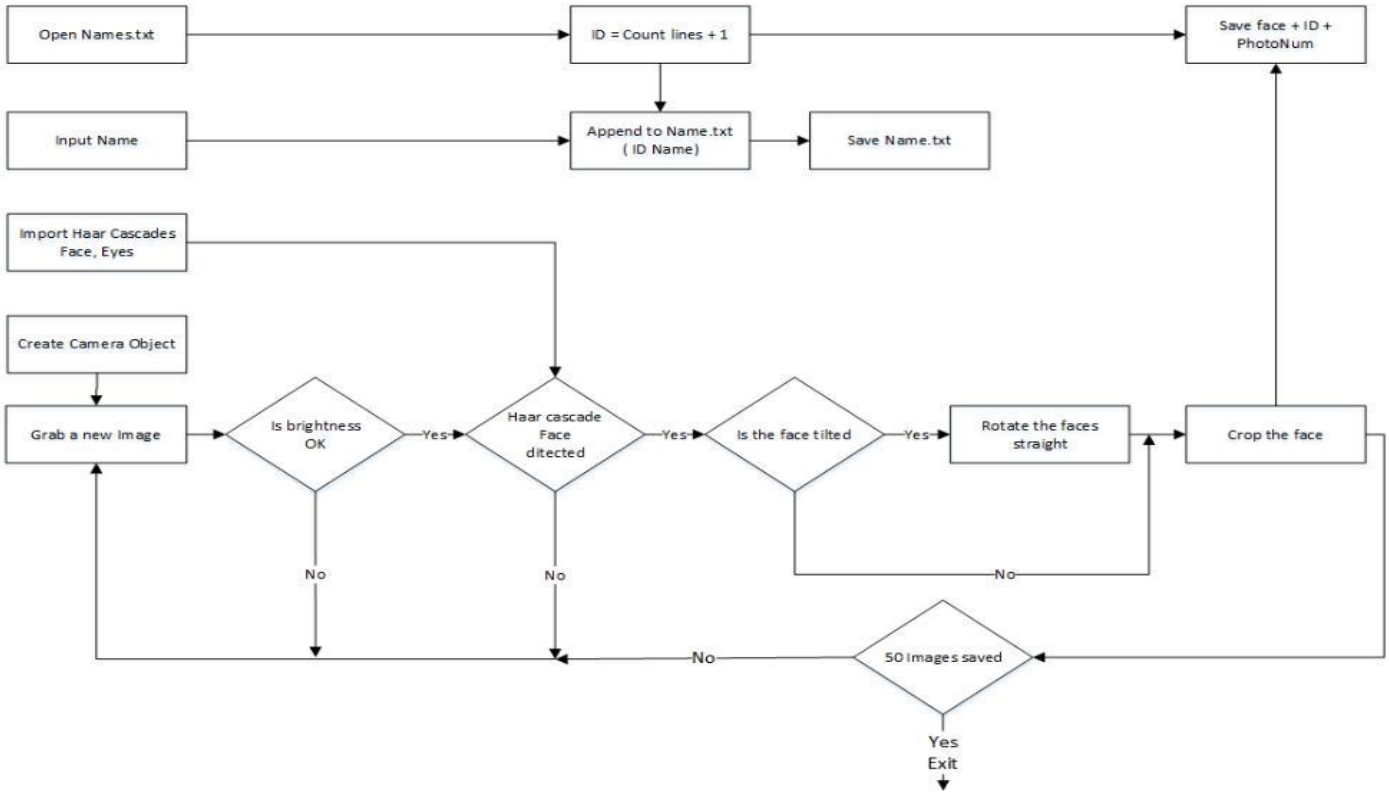


Figure 9: The Flowchart for the image collection

Training the Classifiers

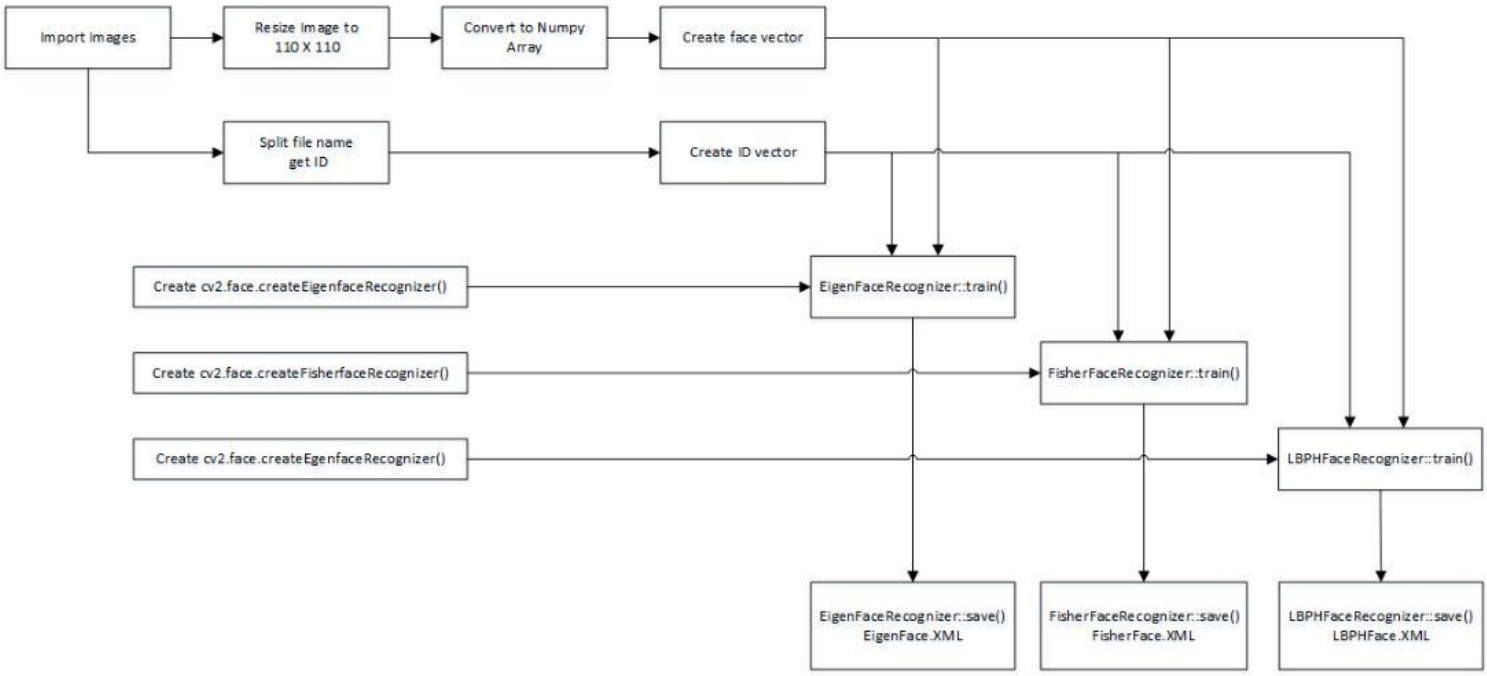


Figure 10: Flowchart of the training application

The Face Recognition

Face recogniser object is created using the desired parameters. Face detector is used to detect faces in the image, cropped and transferred to be recognised. This is done using the same technique used for the image capture application. For each face detected, a prediction is made using Face Recognizer.predict() which return the ID of the class and confidence. The process is same for all algorithms and if the confidence is higher than the set threshold, ID is -1. Finally, names from the text with IDs are used to display the name and confidence on the screen. If the ID is -1, the application will print unknown face without the confidence level. The flow chart for the application is shown in figure 11.

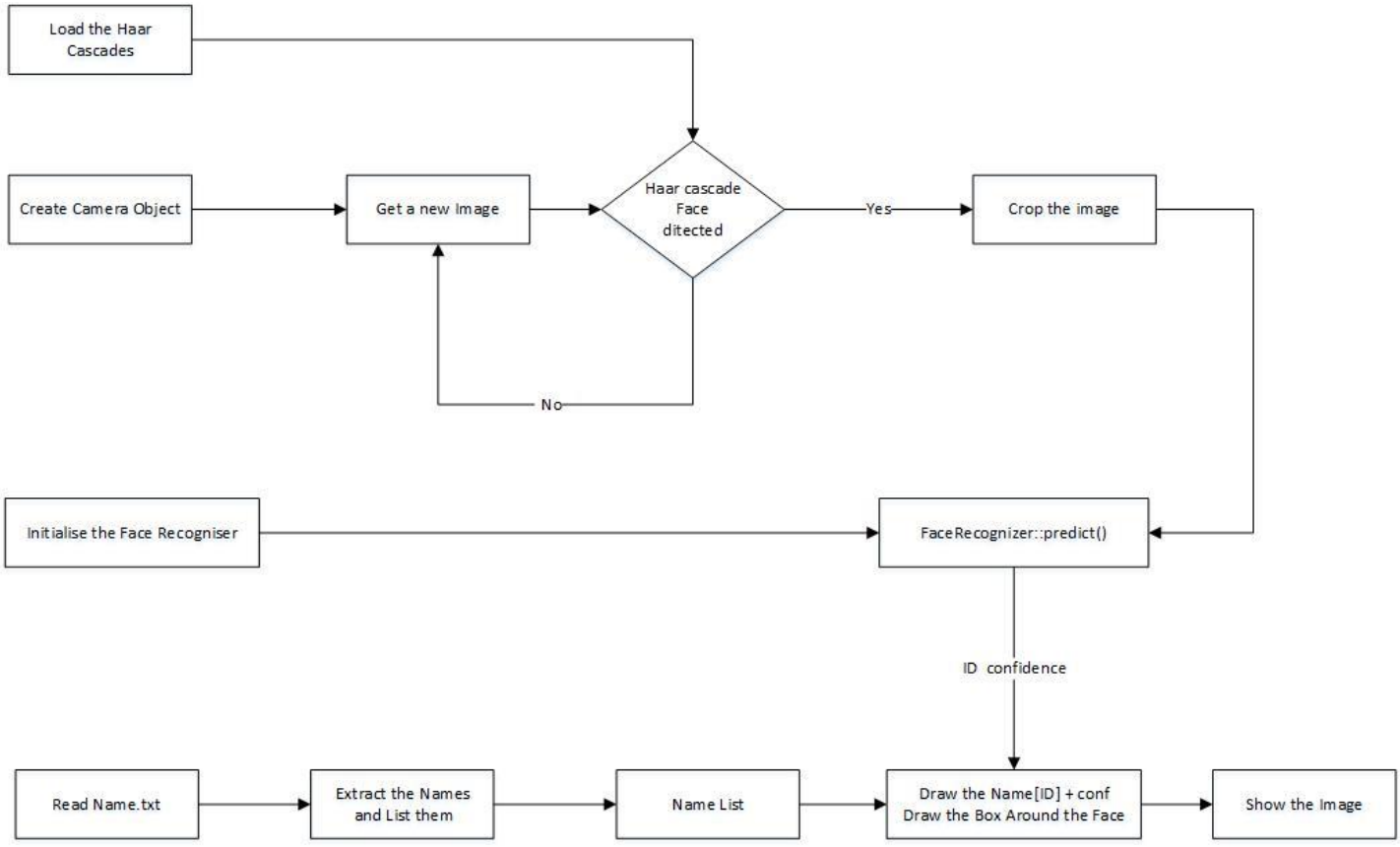


Figure 11: Flowchart of the face recognition application