School of Electronics and Communication Engineering

Final Year B. Tech.

MINI PROJECT LOGBOOK

Academic Year 2020 -2021 Trimester-VIII

Name of Student : Sharaan Thayanithi

PRN Number : S1032181645

Batch :A1

Project guide :- Mrs. Shweta Pawar

S. No. 124, Paud Road, Kothrud, Pune-411038 Maharashtra, India.
Website: www.mitwpu.edu.in

Rules & Regulations

1. All students must enter the correct information in the Logbook.
2. All the entries in the mini project Logbook must be verified by the concerned project guide.
3. Student must report to their respective guide and batch coordinator for mini project in the allocated time as per the time table.
4. Submit soft and hard copies of Synopsis, Mini Project report as per the given format.
5. Project Logbook must be brought at the time of interaction with guide, review presentation & Final project exam.
6. Changes, if any, must be countersigned by the concerned project guide.
7. For project exhibition / seminar / project evaluation all the students must report 15 minutes before the schedule.
8. For queries, if any, contact your project guide and batch coordinator.
9. This booklet must be submitted to Guide / Batch Coordinator or the Head of School.

MIT World Peace University, Pune

School of Electronics and Communication Engineering

Project Report Contents

Sr. No.	Contents	Percentage Contents of Report	Remarks	Page No:
1	Weekly Chart Activity	5 %	Detail Index	4,5
	Cover Page			6
	Certificate			9
	Acknowledgement			10
	Abstract			11
	List of Figure			
	List of Tables			
2	**Chapter 1**	10 %	Background and Literature Review	
	Introduction			
	Review of the related Literature			
3	**Chapter 2**	10 %	Present work related the given topic, limitations, solution given, scope of the statement and objectives.	
	Basic Concepts			
	Scope and Objectives of Project			
4	**Chapter 3**	20 %	Explanation about the project	

	Methodology Output and Result Analysis Accuracy			
	Libraries Used			
	Software and Hardware Requirements			
6	**Chapter 4**	15 %	Result Sheet and Conclusion	
	Conclusion			
	Future Scope			
7	References		References and Datasheets referred	
	Datasheets			

Weekly Chart Activity

Week	Date	Activity Planned/Completed	Sign of the Students	Remarks and Sign of Guide	Remarks and Sign of Batch Coordinator
1.	06/11/2020	Introduction to Mini Project Subject ,Marking Schemes and Pattern of Course			
2.	27/11/2020	Discussion about the general Guidelines, Methodology Midterm Evaluation Scenario w.r.t this Course			
3.	04/12/2020	Discussion about **shortlisting 3 topic** related to AI, ML or Data Science of our choice and how to go about to make the PPT slides of those topics			
4.	11/12/2020	Presenting the PPT (containing basic introduction and methodology) of those 3 shortlisted topics to the Batch Co-ordinator and select any1 suitable topic from it			
5.	18/12/2020	Discussion, by Batch coordinator, about the format of writing **Synopsis** and the topics it should include in it w.r.t our shortlisted topic			
6.	01/01/2020	Presenting the Synopsis to the Batch Co-coordinator and making necessary changes, if told to do so			

Week	Date	Activity Planned/Completed	Sign of the Students	Remarks by Guide And Sign of Guide	Remarks and Sign of Batch Coordinator
7	15/01/2021	Discussion, by Batch Co-ordinator, about the format for preparing **Literature Survey**			
8	22/01/2021	Presenting our prepared Literature Survey w.r.t our final topic to our Batch Co-ordinator and discussing the status/improvement of our project w.r.t its code and output after the MidTerm Evaluation			
9	29/01/2021	Discussion, by Batch Co-ordinator about format of writing a **End Term Report** followed by showing a Sample Report made by a previous year's student			
10	05/02/2021	Discussion by Batch Co-ordinator about the Final Exam Presentation Scenario and pre-presenting it on 10/02/2021 to discuss any changes in report			
11	10/02/2021	Pre-Presenting End Term Report and make necessary changes			
12	12/02/2021	**Final/End Term Mini Project Evaluation** FOR TY B-TECH Trimester II with PPT and Project Report			

TY B. Tech. Trimester VIII MINI PROJECT REPORT

On

Vigilante: AI-based Real-Time Driver Safety Monitoring System

SUBMITTED BY,

Sharaan Thayanithi (S1032181645)

Batch –A1, Roll Number - PA 53

PROJECT MENTOR,

Prof. Shweta Pawar

BATCH CO-ORDINATOR

Prof. Shweta Pawar

Year: 2020 - 2021

School of Electronics and Communication Engineering

MIT World Peace University, Pune

School of Electronics and Communication Engineering
MIT World Peace University
Pune

CERTIFICATE

This is to certify that the B. Tech. Mini Project entitled

(Vigilante: AI-based Real-Time Driver Safety Monitoring System)

work has been carried out successfully by

Sharaan Thayanithi (S1032181645)

during the Academic Year 2020 – 2021 in partial fulfillment of their course of Mini Project for Final Year Electronics and Communication Engineering as per the guidelines prescribed by the MIT World Peace University, Pune

Project Guide	Batch Coordinator	Mini Project Coordinator	Head of School
(Mrs. Shweta Pawar)	(Mrs. Shweta Pawar)	Dr. Trushita S. Chaware	Dr. Vinaya Gohokar

ACKNOWLEDGEMENT

Apart from the efforts of me the success of any dispersion depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this Report.

I wish to express my deep sense of gratitude to my internal guide and Batch Coordinator of Mini Project Subject of Trimester VIII, **Mrs. Shweta Pawar**, for the valuable guidance and useful suggestions, by encouraging me to the highest peak, which help me in completing the project, work in time.

Finally at importantly, I would like to extend my heartfelt thanks to my beloved parents for the blessings my family, my friends, classmates and colleagues for their help and wishes for the successful completion of this report .

ABSTRACT

In this project, a **Real-Time Drowsiness and Yawn Detector application/software** is designed that calculates the width of of the person's eyes and mouth/lip (through WebCam) to determine whether the person is drowsy or yawning or not or both simultaneously, without one affecting the execution of the other.

As one of the most successful applications of image analysis and understanding face recognition has recently gained significant attention especially during the past several years There are at least two reasons for such a trend the first is the wide range of commercial and law enforcement applications and the second is the availability of feasible technologies after 35 years of research Moreover recent significant advances in multimedia processing has also helped to advance the applications of face recognition technology.
Though tracking and recognizing face objects is a routine task for humans building such a system is still an active research Among many proposed face recognition schemes image based approaches are possibly the most promising ones However the 2D images/patterns of 3D face objects can dramatically change due to lighting and viewing variations Hence illumination and pose problems present significant obstacles for wide applications of this type

I have also designed and implemented an automated **Real-Time Face Recognition** system that recognizes the name of the person within the marked bounding box, and list down his record in the mentioned excel file, as an entry in excel, by taking the input from the WebCam. That excel record would contain his Full Name and the Time that person has been scanned by the WebCam. This would be done by extracting Facial Landmarks of the person, process it and show the closest possible match for it, within the bounding.

LIST OF FIGURES

CHAPTER 1

Introduction

Driver Drowsiness and Yawn Detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy, by detecting the state of his eyes or mouth Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads.

Some of the current systems learn driver patterns and can detect when a driver is becoming drowsy or yawning.

Various technologies/methods that are used to successfully detect driver drowsiness and/or yawning state, in real-time, are:

Steering pattern monitoring :

Primarily uses steering input from electric power steering system. Monitoring a driver this way only works as long a driver actually steers a vehicle actively instead of an automatic lane-keeping system.

Vehicle position in lane monitoring :

Uses lane monitoring camera. Monitoring a driver this way only works as long a driver actually steers a vehicle actively instead of an automatic lane-keeping system.

Driver Eye and Mouth/Lip monitoring :

Uses computer vision to trace facial landmarks of the driver's face, observe it and extract his/her eye & mouth landmark to detect Drowsiness and/or Yawn , by using either a built-in camera, Car's DashCam or on mobile devices.

Physiological measurement :

Requires body sensors to measure parameters like brain activity, heart rate, skin conductance, muscle activity, dopamine & adrenaline level, blood oxygen level etc.

In this Project, we are making use of Driver Eye and Mouth/Lip monitoring technology for sucessful detection.

Road traffic safety refers to the methods and measures used to prevent road users from being killed or seriously injured. Typical road users include pedestrians, cyclists, motorists, vehicle passengers, horse riders, and passengers of on-road public transport (mainly buses and trams).

This Prototype Detector System is responsible not only for the Driver's safety but also ensures safety of road users, by warning the driver about his actions beforehand, thereby trying to avoid any mishap.

A **Face Recognition System** is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image. Because computerized facial recognition involves the measurement of a human's physiological characteristics facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless process.[1] Facial recognition systems have been deployed in advanced human-computer interaction, video surveillance and automatic indexing of images.[2] They are also used widely by law enforcement agencies.

Face recognition systems **capture an incoming image from a camera device** in a two-dimensional or three-dimensional way depending on the characteristics of the device.

In this comparison of faces, it **analyses mathematically** the incoming image without any margin of error and it **verifies that the biometric data matches** the person who must use the service or is requesting access to an application, system or even building.

Real-time face detection in video footage became possible in 2001 with the Viola–Jones object detection framework for faces.[17] **Paul Viola and Michael Jones** combined their face detection method with the Haar-like feature approach to object recognition in digital images to launch AdaBoost, the first real-time frontal-view face detector.[18] By 2015 the Viola-Jones algorithm had been implemented using small low power detectors on handheld devices and embedded systems. Therefore, the Viola-Jones algorithm has not only broadened the practical application of face recognition systems but has also been used to support new features in user interfaces and teleconferencing.

First face detection is used to segment the face from the image background. In the second step the segmented face image is aligned to account for **face pose, image size and photographic properties, such as illumination and gray scale**. The purpose of the alignment process is to enable the accurate **localization of facial features** in the third step, the **facial feature extraction**. Features such as eyes, nose and mouth are pinpointed and measured in the image to represent the face. The so established feature vector of the face is then, in the fourth step, matched against a database of faces, which then shows the recognized face.

Review Of Literature

[1] **Authors:** J.P Hespanha, D.J. Kriegman and P.N Behlushmer
Published in: IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 19, Issue: 7, Jul 1997)

Page(s): 711 – 720
Date of Publication: Jul 1997
INSPEC Accession Number: 5661535
DOI: 10.1109/34.598228
Publisher: IEEE

Rather than explicitly modeling this deviation, we linearly project the image into a subspace in a manner, which discounts those regions of the face with large deviation. Our projection method is based on Fisher's linear discriminant and produces well separated classes in a low-dimensional subspace, even under severe variation in lighting and facial expressions. The Eigen face technique, another method based on linearly projecting the image space to a low dimensional subspace, has similar computational requirements.

[2] **Authors:** S.Romdhani, P.Torr, B. Scholkopf, A. Blake

Published in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001

Date of Conference: 7-14 July 2001
Date of Publication: 07 August 2002

INSPEC Accession Number: 7024346
DOI: 10.1109/ICCV.2001.937694

Publisher: IEEE

A non-linear support vector machine is used to determine whether or not a face is contained within the observation window. The non-linear support vector machine operates by comparing the input patch to a set of support vectors (which can be thought of as face and anti-face templates). Each support vector is scored by some nonlinear function against the observation window and if the resulting sum is over some threshold a face is indicated.

[3] **Authors:** Pooneh.R. Tabrizi, Reza. A. Zaroofi

Published in: 2008 First Workshops on Image Processing Theory, Tools and Applications

Date of Conference: 23-26 November 2008
Date of Publication: 09, January 2009

INSPEC Accession Number: 10454541
DOI: 10.1109/IPTA.2008.474385

Publisher: IEEE

Drowsiness detection is vital in preventing traffic accidents. Eye state analysis - detecting whether the eye is open or closed - is critical step for drowsiness detection, by proposing an easy algorithm for pupil center and iris boundary localization and a new algorithm for eye state analysis, which we incorporate into a four step system for drowsiness detection: face detection, eye detection, eye state analysis, and drowsy decision. This new system requires no training data at any step or special cameras. Our eye detection algorithm uses Eye Map, thus achieving excellent pupil center and iris boundary localization results on the IMM database. Our novel eye state analysis algorithm detects eye state using the saturation (S) channel of the HSV color space.

[4] **Authors:** Belal Alshaqaqi, Mokhtar Keche, Meriem Boumehed & Abdelaziz Ouamri

Published in: 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)

Date of Conference: 12-15 May 2013
Date of Publication: 19 September 2013

INSPEC Accession Number: 13779602
DOI: 0.1109/WoSSPA.2013.6602353

Publisher: IEEE

Advanced Driver Assistance System (ADAS) is presented to reduce the number of accidents due to drivers fatigue and hence increase the transportation safety; this system deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence. We propose an algorithm to locate, track, and analyze both the drivers face and eyes to measure PERCLOS, a scientifically supported measure of drowsiness associated with slow eye closure.

CHAPTER 2

Basic Concepts

Before going in details of the project, some basic concepts used in this project are to be known. These are discussed in this chapter.

2.1 Haar Cascade Classifiers

Haar Cascade Classifiers are made up of **Haar-like features of** are digital image features used in object recognition and uses "**Integral Image**" concept to compute features detected by it. An **Integral Image** Concept is where each pixel represents the cumulative sum of a corresponding input pixel with all pixels above and left of the input pixel. These are tools that owe their name to their intuitive similarity with Haar wavelets and were used in the first real-time face detector. These are pre-trained face-recognition and identification models with the **extension - .xml**, which has read and analyzed the location of different sensory organs/parts, i.e.: the features of a human face (eyes, nose, mouth, jawline, forehead) into pixels in image and convert then into rectangle by function.

Haar Cascade Classifiers was accompanied with the "**Viola–Jones" object detection framework/method** proposed in 2001 by Paul Voila and Michael Jones, and are a part of the **OpenCv library** of Python Language.

Basically, it considers **adjacent rectangular regions** at a specific location in a detection window. Each feature is a single value obtained by **subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle**. This difference is then used to categorize subsections of an image. For example, with a human face, it is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

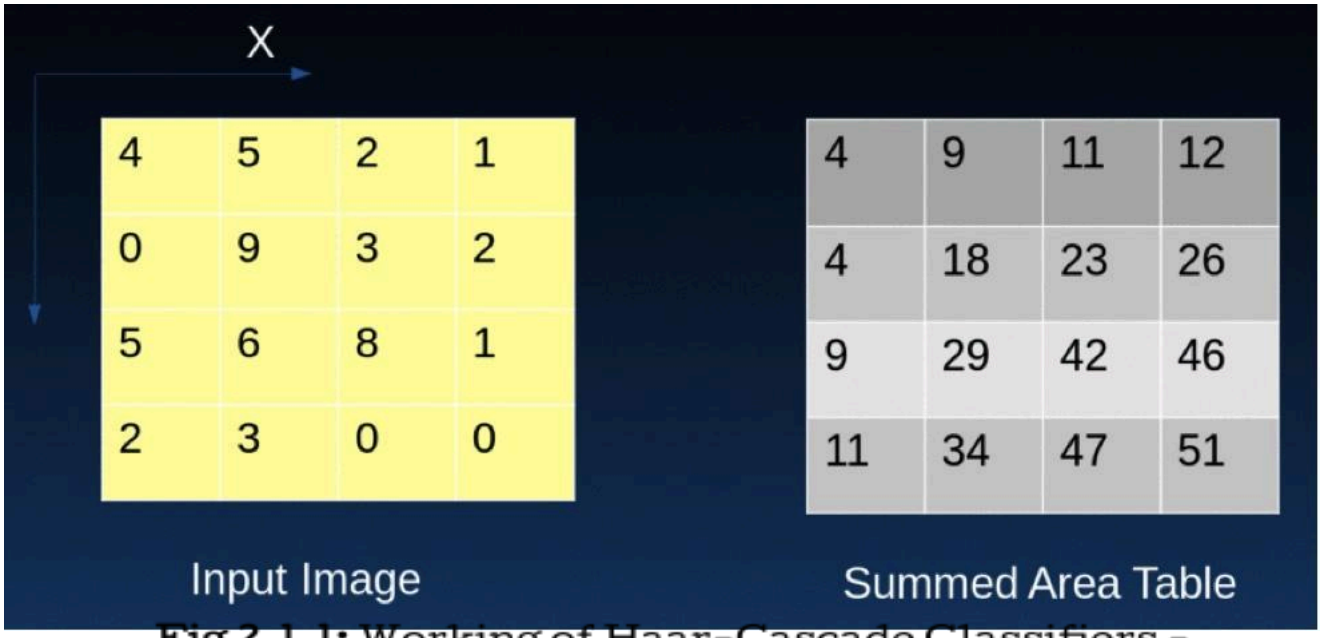


Fig 2.1.1: Working of Haar-Cascade Classifiers – how it calculates **Summed Area Matrix** from I/P image pixel values

In the detection phase of the Viola–Jones object detection framework, a window of the target size is moved over the input image, and for each subsection of the image the Haar-like feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because such a Haar-like feature is only a weak learner or classifier, they use an algorithm known as **Ada-boost Learning Algorithm**, to thereby strengthen its final output of recognized face.

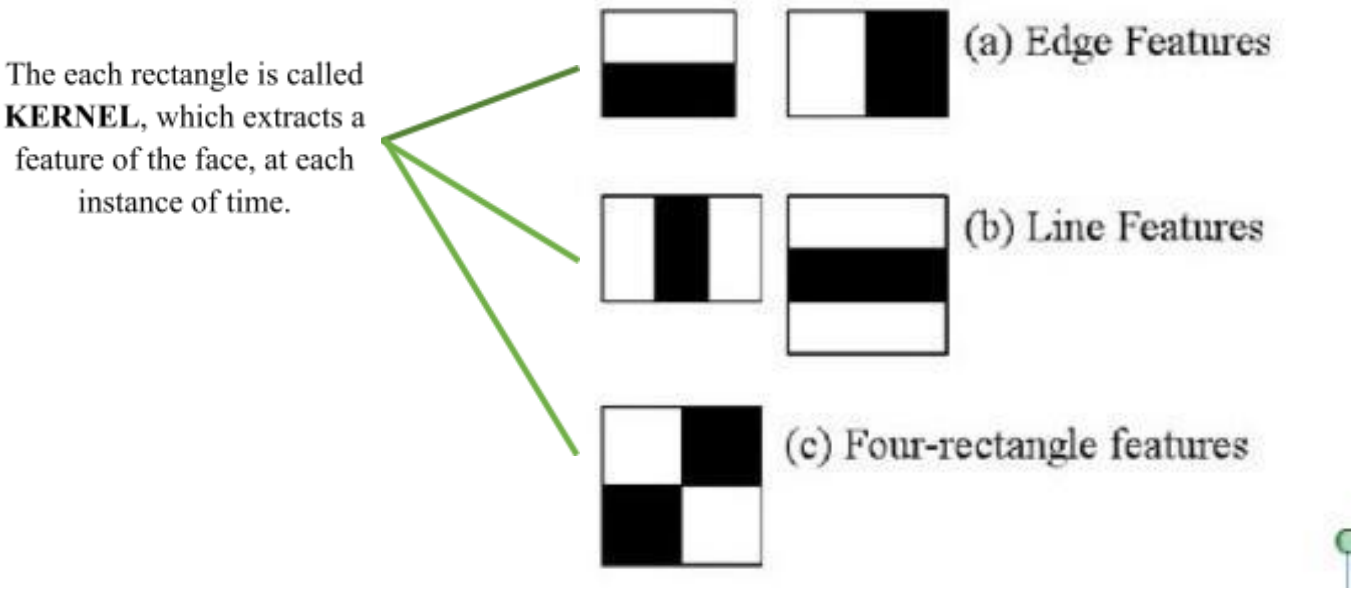


Fig 2.1.2: Kernel Generated by Haar Cascade Classifier for Feature Extraction

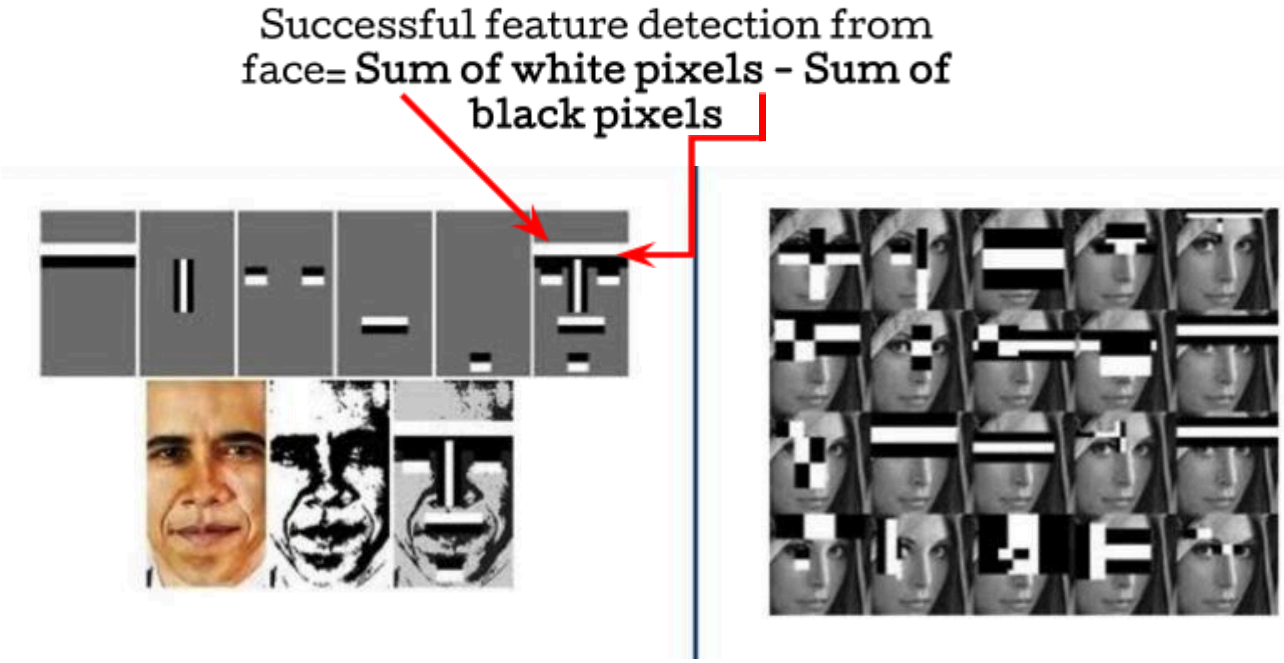


Fig 2.1.3: Feature-Extraction mechanism from the human face using the Kernel of different size

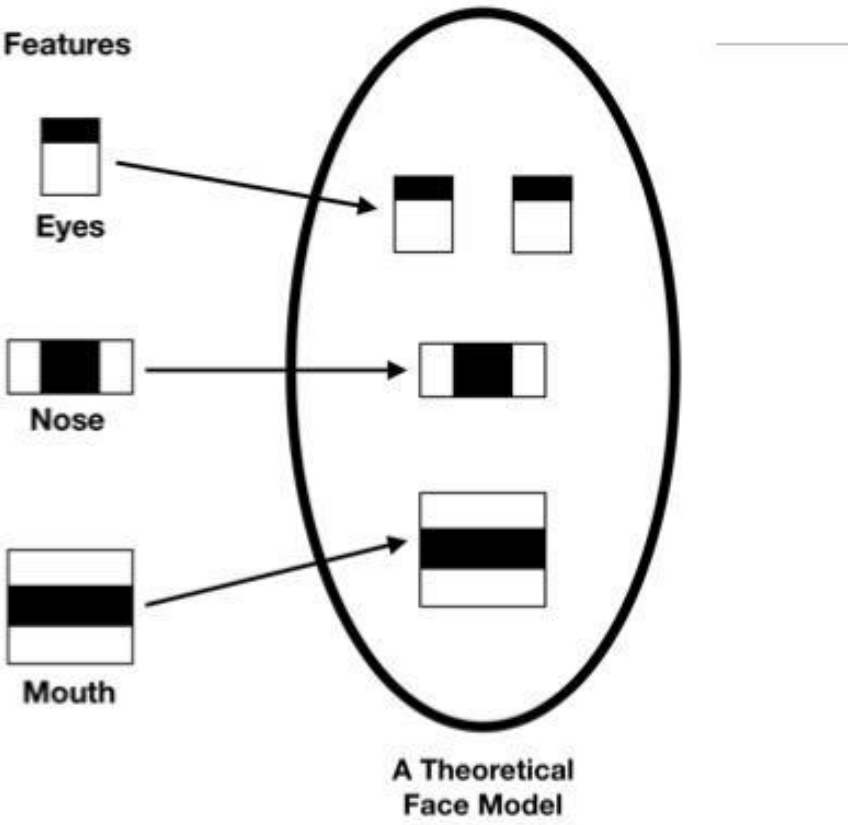


Fig 2.1.4: Feature Matching of Kernel with respect to a Human's Face

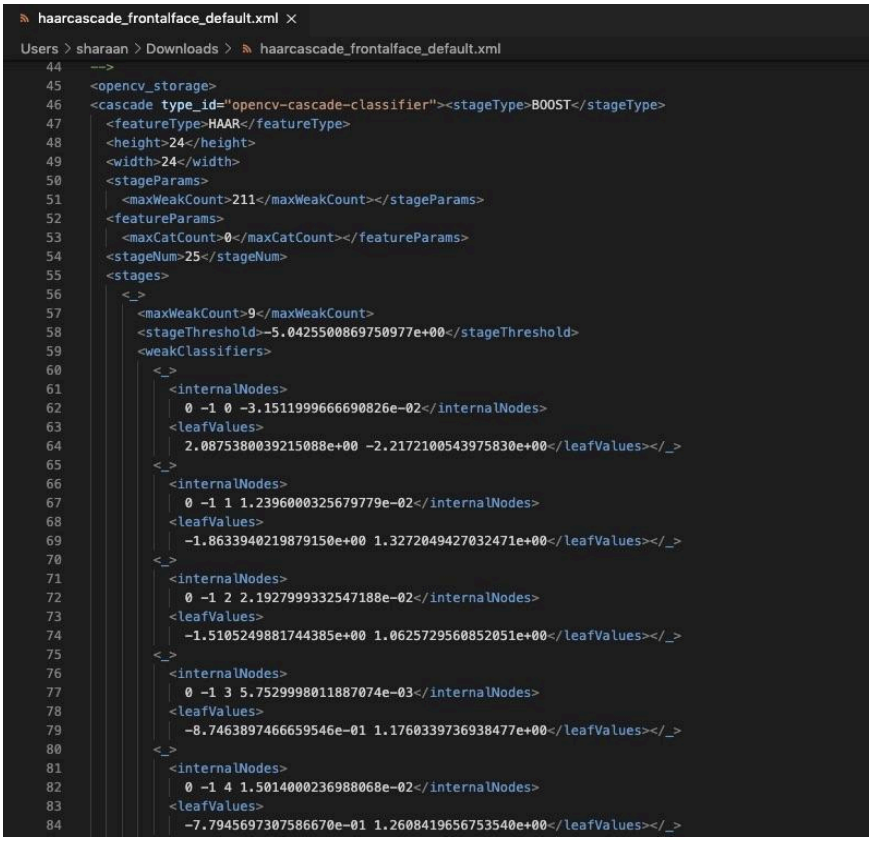


Fig 2.1.5: The haarcascade_frontalcatface.xml file

The concept of a **Cascade of Classifiers** is that instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window, which passes all stages, is a face region.

A Haar Cascade Classifier is a **machine learning-based approach**, that it initially needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. They are just like our convolutional kernel.

- In our project, to detect the front part (2D Visualization) of human's face, we make use of **haarcascade_frontalface_default.xml**

Other features included by Haar-Cascade Classifiers are:

- haarcascade_eye .xml - to detect and locate the **human eyes**
- haarcascade_fullbody .xml - to detect and locate **full body** of a human
- haarcascade_lowerbody .xml - to detect and locate the human's body below waist
- haarcascade_frontalcatface.xml - to detect the **front part a cat's face**
- haarcascade_smile.xml – to detect & extract **human's smile** from his face

2.2 Ada-Boost Algorithm

The Ada-Boost learning Algorithm -short for Adaptive Boosting, is the main/core algorithm, primarily used in Haar-Cascade Classifiers. It is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire.

It is an **ensemble learning technique (meta-learning)** that attempts to create a strong classifier from a number of weak classifiers, best used to boost the **performance of decision trees** on binary classification problems. It uses iterative approach to learn from its mistakes and previously executed weak classifiers and turns them into strong ones.

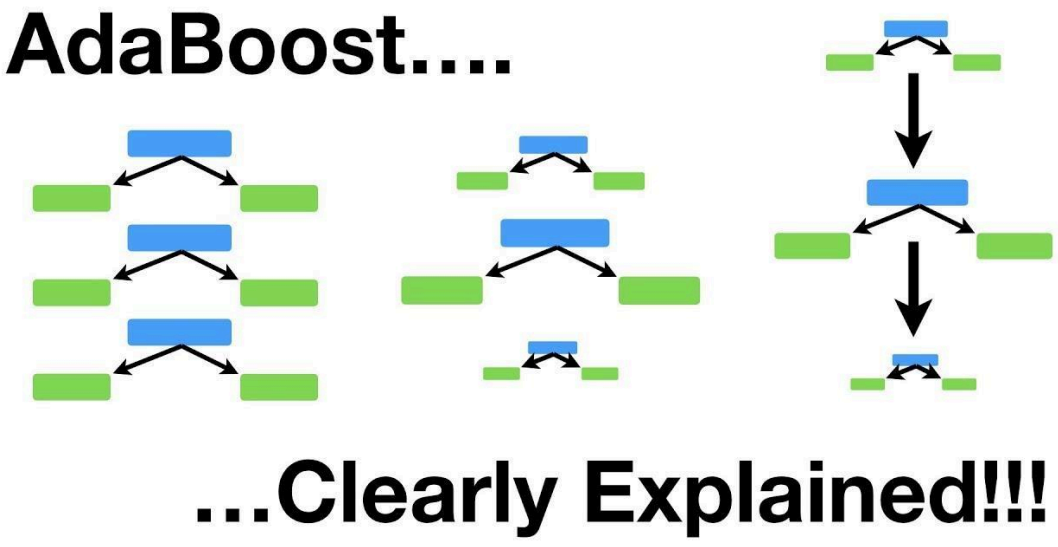


Fig 2.2.1: Ada-Boost uses Decision-Trees

When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative **'hardness'** of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

The output of the other learning algorithms i.e.: weak learners' is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

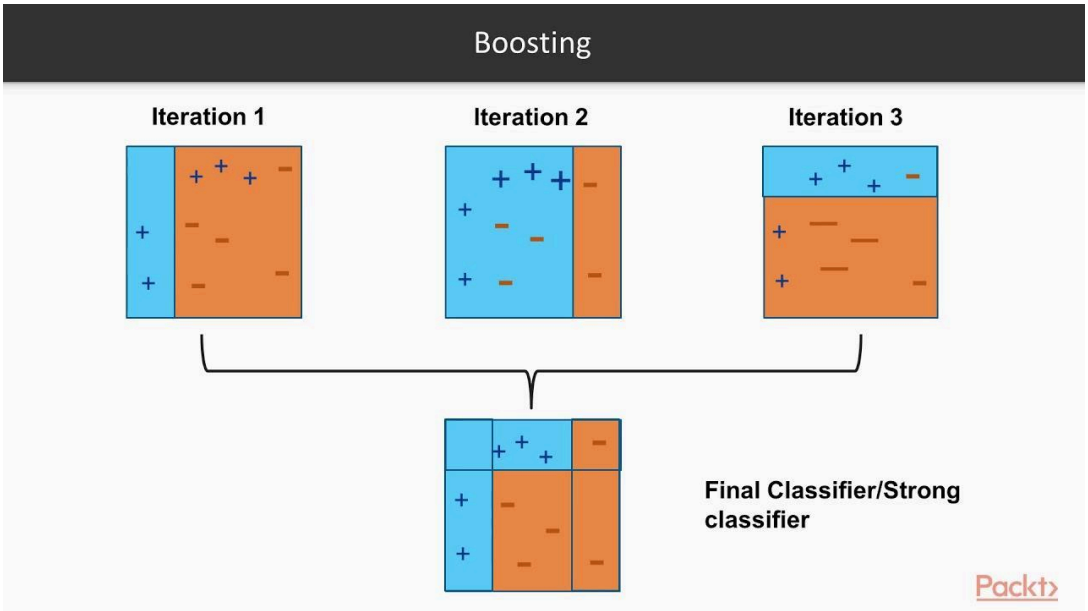


Fig 2.2.2: Mechanism of Ada-Boost algorithm

Ada-Boost, in some problems it can be less susceptible to the over-fitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Ada-Boost Algorithm is used in **classification** and not in regression models.

2.3 Shape-Predictor Tool

Just like Haar Cascade Classifiers, **Shape-Predictor** is a pre-trained and modeled tool used to predict & physically localize the facial landmarks of a human face, by converting and analyzing each part as a into a shape. The parts detected by this tool include the human's eyebrows, eyes, nose, mouth, lip and jawline.Shape_predictor () tool takes in an image region containing some object and outputs a set of point locations that define the pose of the human's face.

It's a type of Landmark's facial detector with pre-trained models, the Dlib is used to estimate the location of **68 coordinates (x, y)** that map the facial points on a person's face like image below. It is mostly saved as a markup fie with the **extension- .dat**

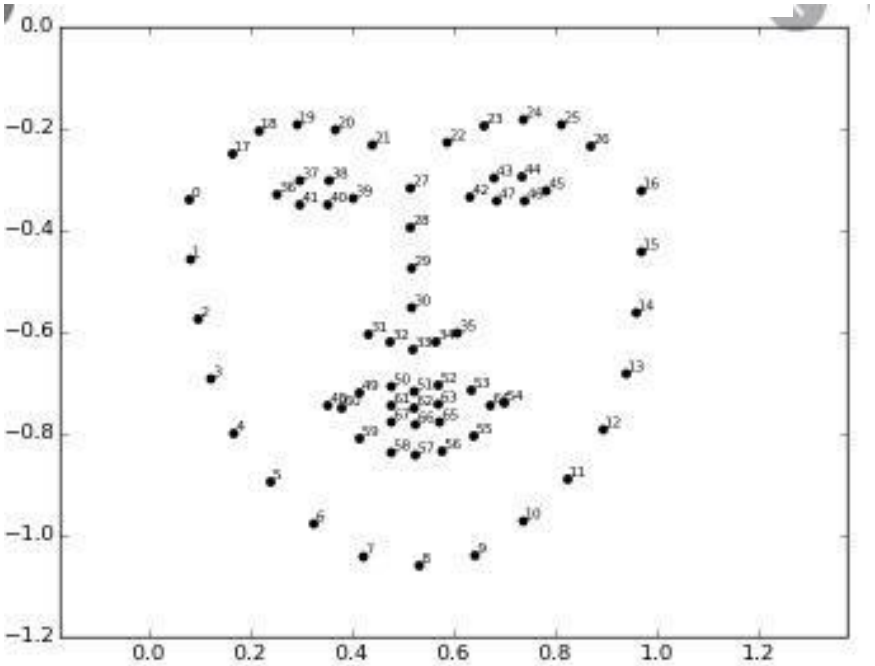


Fig 2.3.1: The 68 point's landmark traced w.r.t the (x, y) Co-ordinates

A file with the .dat file extension is a generic data file that stores specific information relating to the program that created the file.

Unlike Haar Cascade Classifiers, Shape-Predictor Tool belong to the **Dlib library** of Python Language. Dlib is a general-purpose cross-platform software library written in the programming language C++. It is open-source software released under a Boost Software License, in 2002, designed to implement a variety of machine learning algorithms, including classification, regression, clustering, data transformation, and structured prediction.

The Shape-Predictor tool points are identified from the pre-trained model where the iBUG300-W dataset was used.

- In our project, to detect facial landmarks/regions, we make use of **shape_predictor_68_face_landmarks.dat**

The following Human facial regions parts are essentially tried to localize, label & traced by the Shape Predictor, (corresponding to its numbered marking from the image below), as its output:

- ✔ Mouth
- ✔ Right eyebrow
- ✔ Left eyebrow
- ✔ Right eye
- ✔ Left eye
- ✔ Nose
- ✔ Jaw

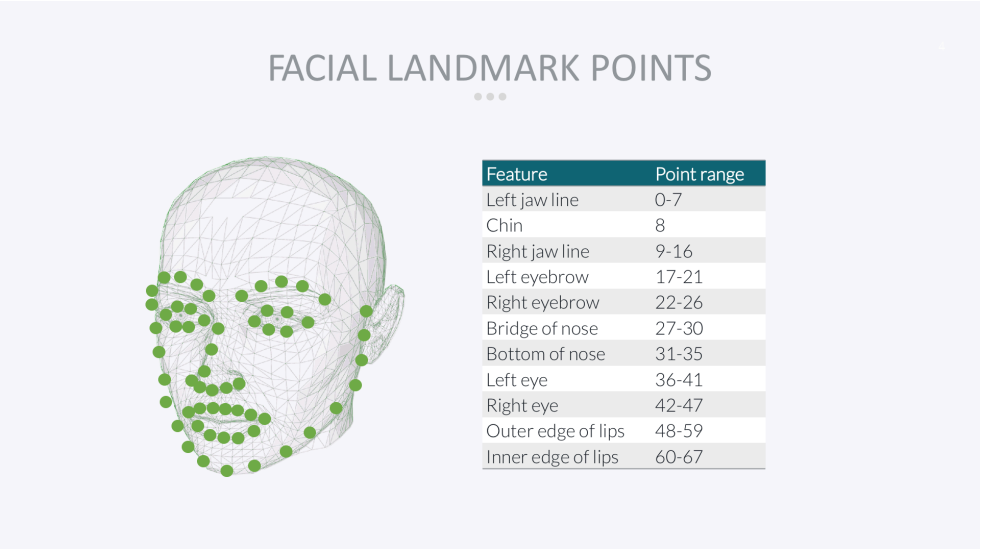


Fig 2.3.2: 3-D Representation of how 68-points facial landmarks are detected

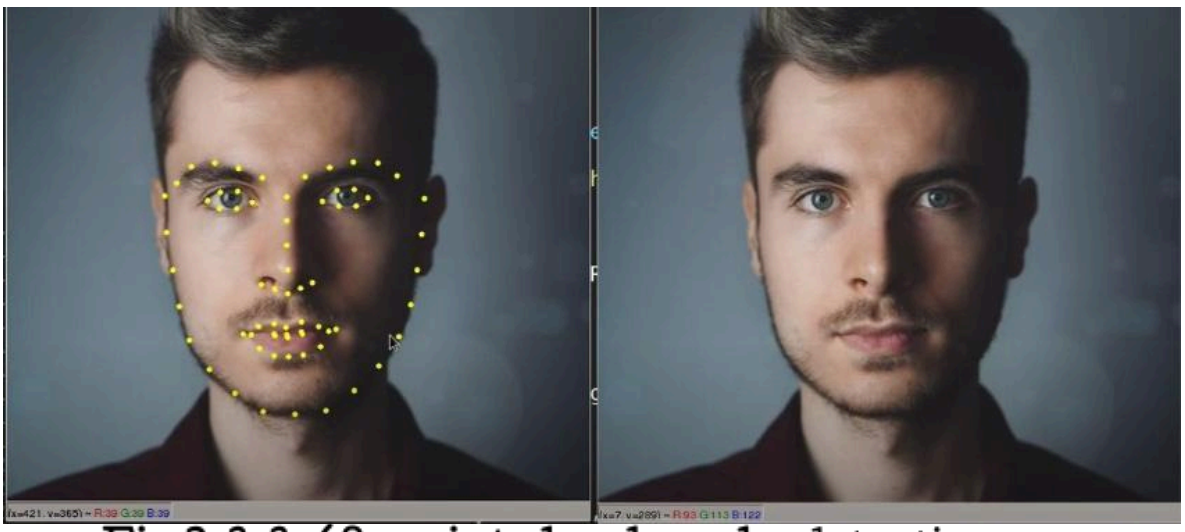


Fig 2.3.3: 68-points landmarks detection on an actual Human face

2.4 Convex Hull

In geometry, a **Convex hull** of a shape is the smallest convex set that contains it, that can be defined either as the intersection of all convex sets containing a given subset of a Euclidean space, or equivalently as the set of all convex combinations of points in the subset. For a bounded subset of the plane, the convex hull may be visualized as the shape enclosed by a rubber band stretched around the subset.

Basically, **Convex hull** is a structure used to connect closely spaced adjacent points in order to create a desired boundary of our requirement and interest.

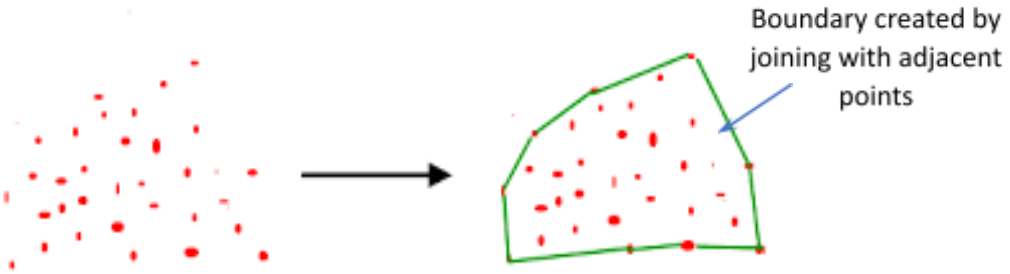


Fig 2.4.1: Basic Convex Hull Structure

They can be solved in time for two or three dimensional point sets, and in time matching the worst-case output complexity given by the upper bound theorem in higher dimensions.

In my project, I have used Convex Hull concept to connect adjacent points of traced width/distance between 2 eyelids of an eye, for both eyes (for drowsiness detection) and the gaping width/distance of mouth/lips (for yawn detection)

The terminology used for the former is called **Eye-Aspect-Ratio (EAR)** and for the latter, it is simply recognized/termed as **Gaping Width** or **Lip Width.**

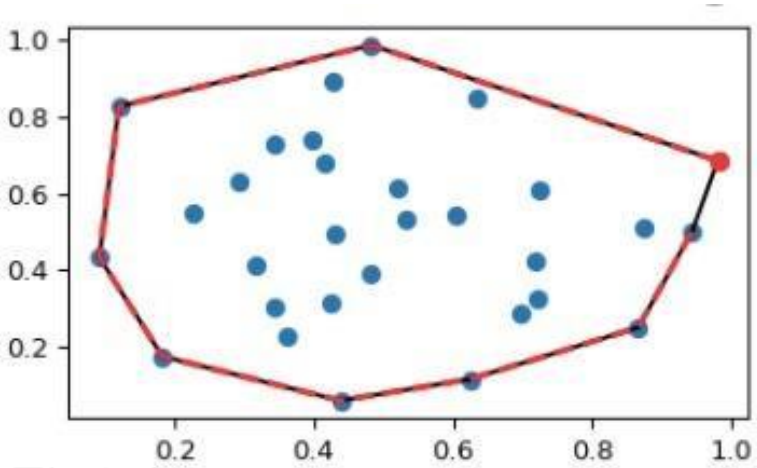


Fig 2.4.2: Convex Hull generated in Python by scipy.spatial package

Problem Statement

Despite the successes of many systems based on the FERET test, many issues remain to be addressed among those issues the following two are prominent for most systems:-

1) The Illumination problem 2) the Pose problem

Fig 2.5: The Illumination problem

The illumination problem is illustrated below in figure where the same face appears differently due to the change in lighting More specifically, the changes induced by illumination could be larger than the differences between individuals, causing systems based on comparing images to misclassify the identity of the input. For example, the popular Eigen subspace projections used in many systems as features have been analyzed under illumination. The conclusions suggest that significant illumination changes cause dramatic changes in the projection coefficient vectors, and hence can seriously degrade the performance of subspace-based methods.

The pose problem is illustrated in below image, where the same face appears differently due to changes in viewing condition. Moreover, when illumination variation also appears in the face images the task of face recognition becomes even more difficult. In the analysis and classification of various pose problems

are performed using a reflectance model with varying albedo Using such a model the difficulty of the pose problem can be assessed and the efficacy of existing methods can be evaluated systematically.

Fig 2.6: The Pose and (illumination) problem

For example, the pose problem has been divided into 3 categories:
1) The simple case with small rotation angles 2) The most commonly addressed case when there are a set of training image pairs (frontal and rotated images), and 3) The most difficult case when training image pairs are not available and illumination variations are present

Solving the illumination problem:-

As a fundamental problem in image understanding literature, illumination problem is generally quite difficult and has been receiving consistent attentions For face recognition, many good approaches have been proposed utilizing the domain knowledge, i.e. all faces belong to one face class These approaches can be broadly divided into four types :-

1) Heuristic methods including discarding the leading principal components

2) Image comparison methods where various image representations and distance measures are applied

2) Class based methods where multiple images of one face under a fixed pose but different lighting conditions are available, and model based approaches where 3D models are employed.

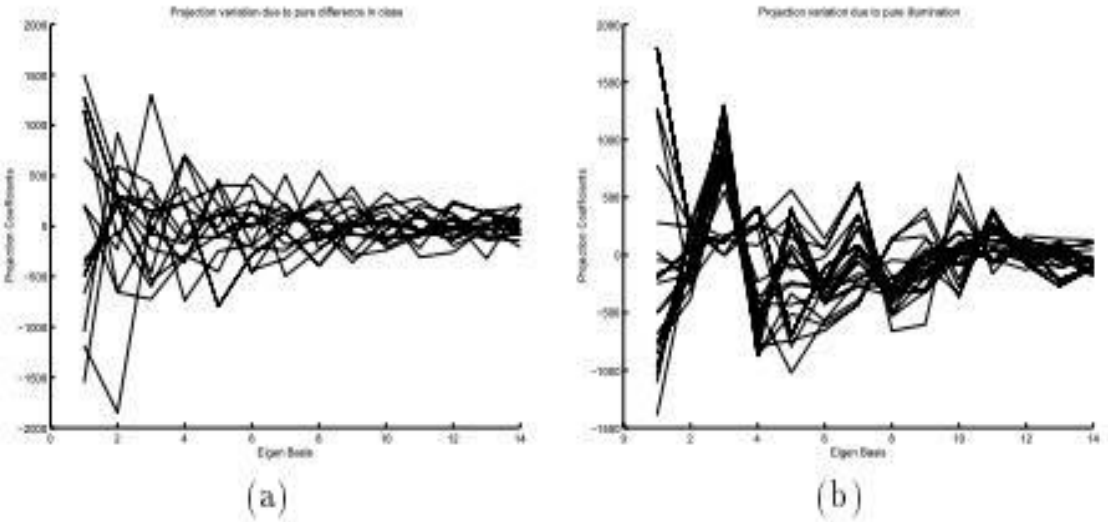
Solving the Pose problem:-

Researchers have proposed various methods to handle the rotation problem Basically they can be divided into three classes: -

1) Multiple images based methods when multiple images per person are available
2) Hybrid methods when multiple training images are available during training but only one database image per person is available during recognition

3) Single image/shape based methods when no training is carried out.

The third approach does not seem to have received much attention.



(a) (b)

Fig 2.7: Change of projection vectors due to a) Class variation and b) Illumination change

Scope and Objective of the Project

For Drowsiness and Yawn detection:

Automotive population is increasing exponentially in the country. The biggest problem regarding the increased traffic is the raise in number of road accidents. Road accidents are undoubtedly a global menace in our country. The global status report on road safety published by the World Health Organization (WHO) identified the major causes of road accidents are due to driver errors and carelessness.

Using **Drowsiness and Yawn detection system**, driver and road users' safety can be implemented in normal cars also especially during Night driving, Driver sleepiness and his alcoholism (which are the key players in accident scenario), thereby warning/alerting the driver beforehand, to make and take necessary actions/measures to prevent any accident, mishap or crash to occur due to his/her carelessness, which would take a toll of lives of people(s).

Objective:

- ✔ Driver drowsiness detection is a **car safety technology**, which helps to save the life of the driver by preventing accidents when the driver is getting drowsy.
- ✔ The main objective is to first design a system to detect driver's drowsiness by continuously monitoring **retina of the eye**.
- ✔ The system works in spite of driver wearing spectacles and in various lighting conditions.
- ✔ To alert the driver on the detection of drowsiness by using buzzer or alarm.
- ✔ **Speed of the vehicle** can be reduced.
- ✔ **Traffic management** can be maintained by reducing the accidents

For Face Recognition:

In today's networked world, the need to maintain the security of information or physical property is becoming both increasingly important and increasingly difficult. From time to time we hear about the crimes of credit card fraud, computer break-in's by hackers, or security breaches in a company or government building. Recently, technology became available to allow verification of "true" individual identity, the "**biometrics**". Biometric access control are automated methods of verifying or recognizing the identity of a living person on the basis of some physiological characteristics, such as fingerprints or facial features, or some aspects of the person's behavior, like his/her handwriting style or keystroke patterns. Since biometric systems identify a person by biological characteristics, they are difficult to forge. **Face Recognition** is one kind of a Biometric Verification.

Though there are some weaknesses of facial recognition system, there is a tremendous use in the every country as it is fast and non-invasive identity verification as people don't have to remember their login credentials(username, id, password) and time taken is also very less compared to manual login This system can be effectively used in ATM's ,identifying duplicate voters, passport and visa verification, driving license verification, in defense, competitive and other exams, in governments and private sectors.

Objective:

- ✔ The **objective of face recognition** is, from the incoming image, to find a series of data of the same **face** in a set of training images in a database.
- ✔ Ensuring the Recognition process is carried out in real-time, something that is not available to all biometric **facial recognition software** providers.
- ✔ To ensures **no duplicity or misuse/mishandling of the records**, and can be used to detect frauds, burglars etc.
- ✔ Efficient and accurate retrieval of person's data based on his face, thereby making it unique from other verification process

CHAPTER 3

Methodology

- **For Face Recognition:**

1) Steps: -

We will be making use of **Haar Cascade Classifiers** (as discussed in pages 15,16) i.e.: **haarcascade_frontalface_default.xml** to detect and recognize face through the video capturing frame.

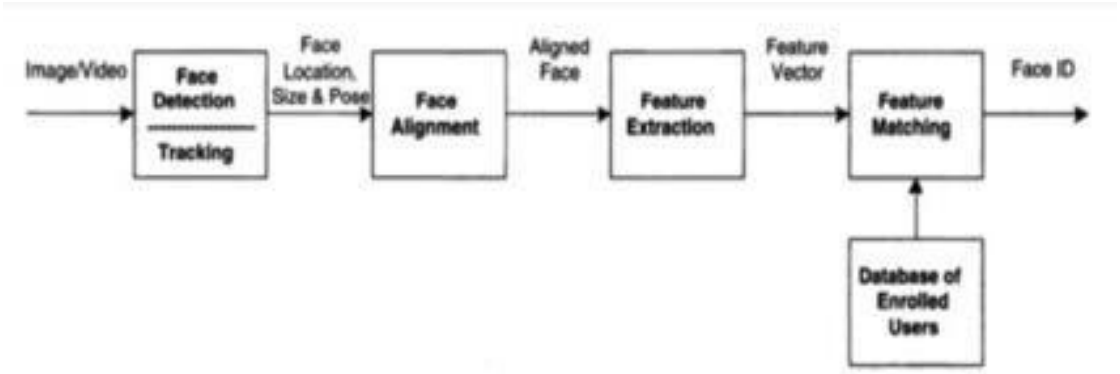


Fig 3.1: Flow chart for Face Recognition

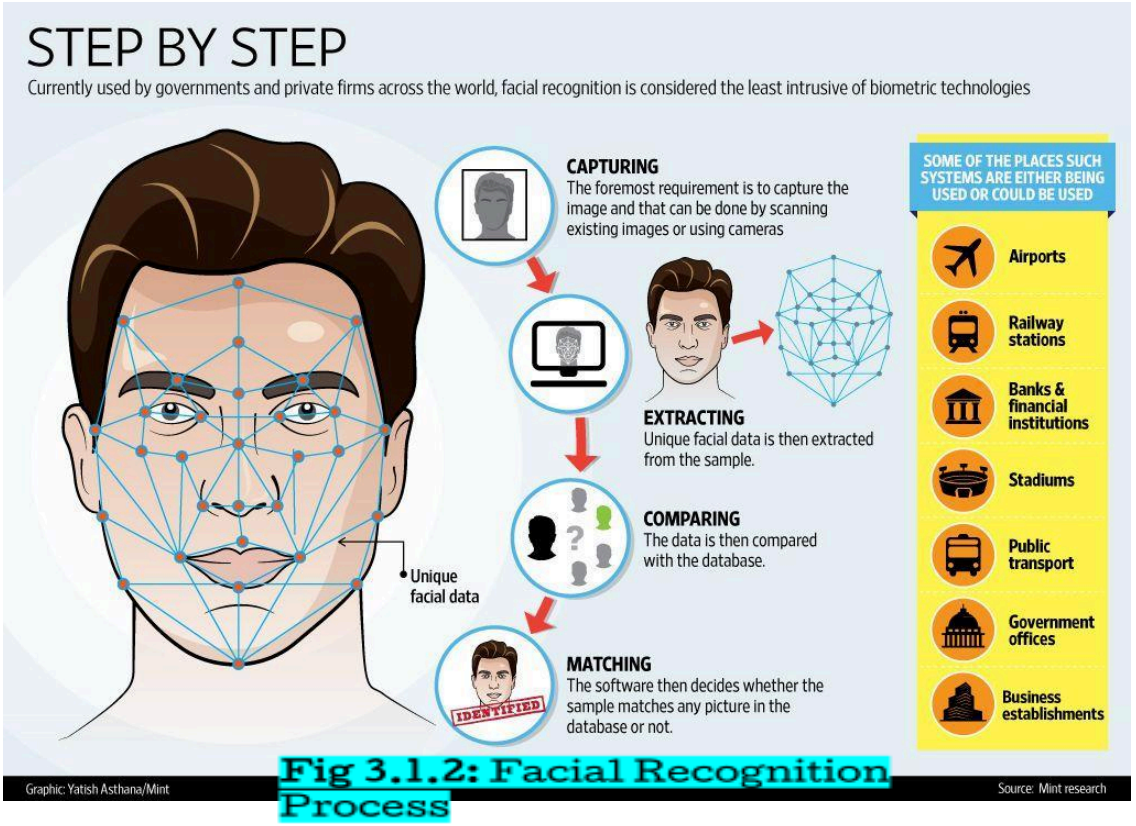


Fig 3.1.2: Facial Recognition Process

Face Detection: -

To begin, the Camera/WebCam will capture the frame, detect and recognize the face in the frame. The face is best detected when the person is looking directly at the camera. The technological advancements have enabled slight variations from this to work as well. This step ensures locating one or more faces in the image and mark with a **bounding box**.

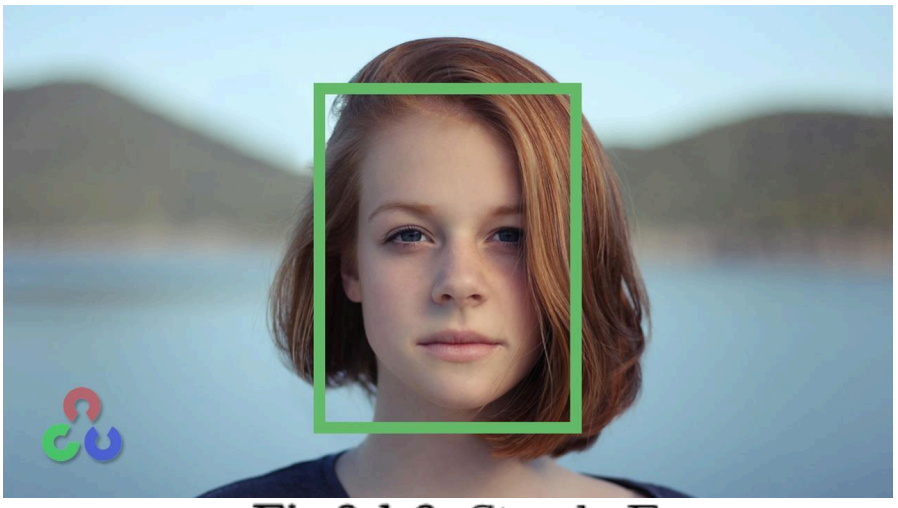


Fig 3.1.3: Step 1- Face Detection

Face Analysis & Alignment: -

Next, a photo of the face is captured and analyzed. Most facial recognition relies on 2D images rather than 3D because it can more conveniently match a 2D photo

with public photos or those in a database. Distinguishable landmarks or nodal points make up each face. **Haar Cascade**, with its pre-trained algorithm will analyze the nodal points and other features of the face such as the distance between your eyes or the shape of your cheekbones.

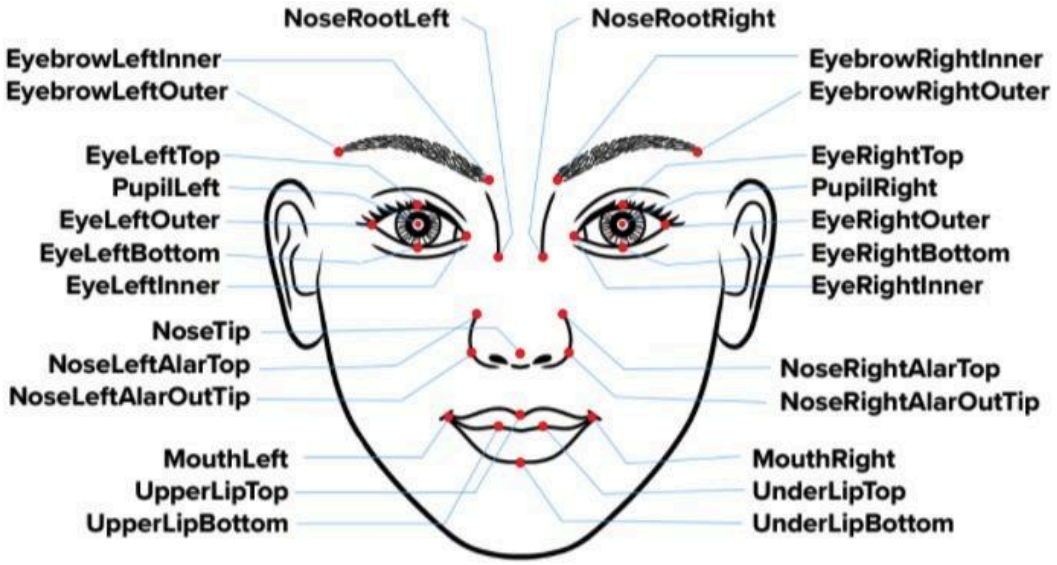


Fig 3.1.4: Step 2 – Face Analysis & Alignment

Feature Extraction: -

The analysis of your face is then turned into a mathematical formula, modeling the image of a face as a two dimensional array of numbers, i.e., its **pixel values**.

These facial features become numbers in a code. This unique numerical code is called a face print, similar to the unique structure of a thumbprint. The system has to extract meaningful data from the facial images, identifying the most relevant bits of data and ignoring all of the "noise." It's also referred to as encoding.

Finally **Feature Extraction** includes normalizing the face to be consistent with the database, such as geometry and photometric. Extracted features from the face can be used for the recognition task.

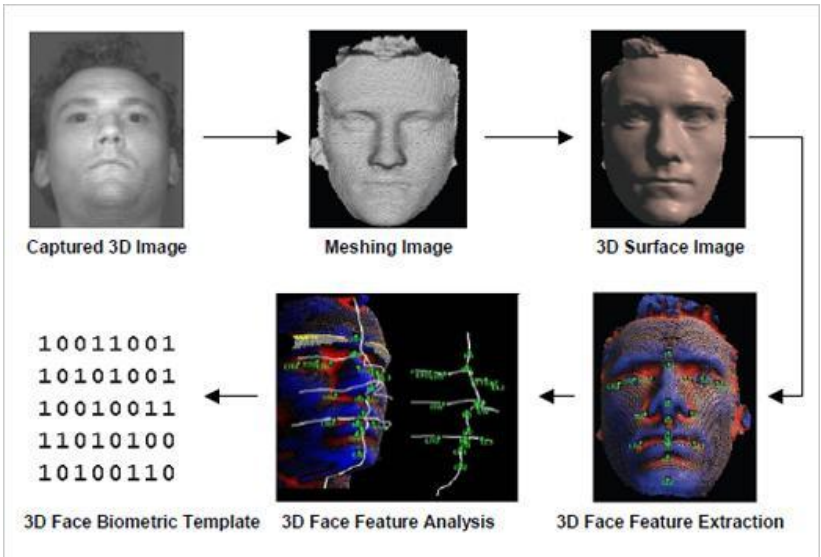


Fig 3.1.5: Step 3 - Feature Extraction

Feature Matching and Recognition: -

Face Recognition: - Perform matching of the unique data features of face obtained, against one or more known faces in a prepared **database**.

Fig 3.1.6: Step 4 - Successful Face Recognition

3) Training my model:-

- I have used around 2000 images to train a face for successful face recognition.
- By mentioning the path of the directory containing my training images, I used the os.walk (directory) function from the **os** package to walk/read through each image from that sub file of that directory.
- Then, by using **os.path.join ()** function, I connected/joined all those images machine-learned onto one single file with the extension "**training_whole.yml**", (which will then be fed into Haar Cascade Classifiers, to extract facial features) that contains all the data of images fed into it while it was put to learn/train.

- .yml is the human readable data serialization language, mostly used to configure files and applications ,where raw data is being stored or transmitted for training and testing purposes.

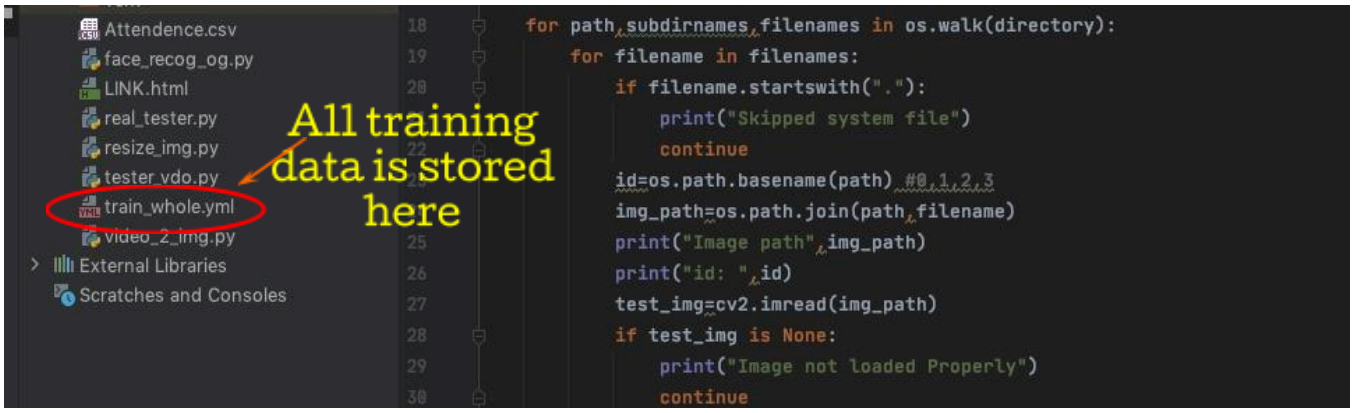


Fig 3.2.1: Code snippet for accessing the data (images), required for training purpose

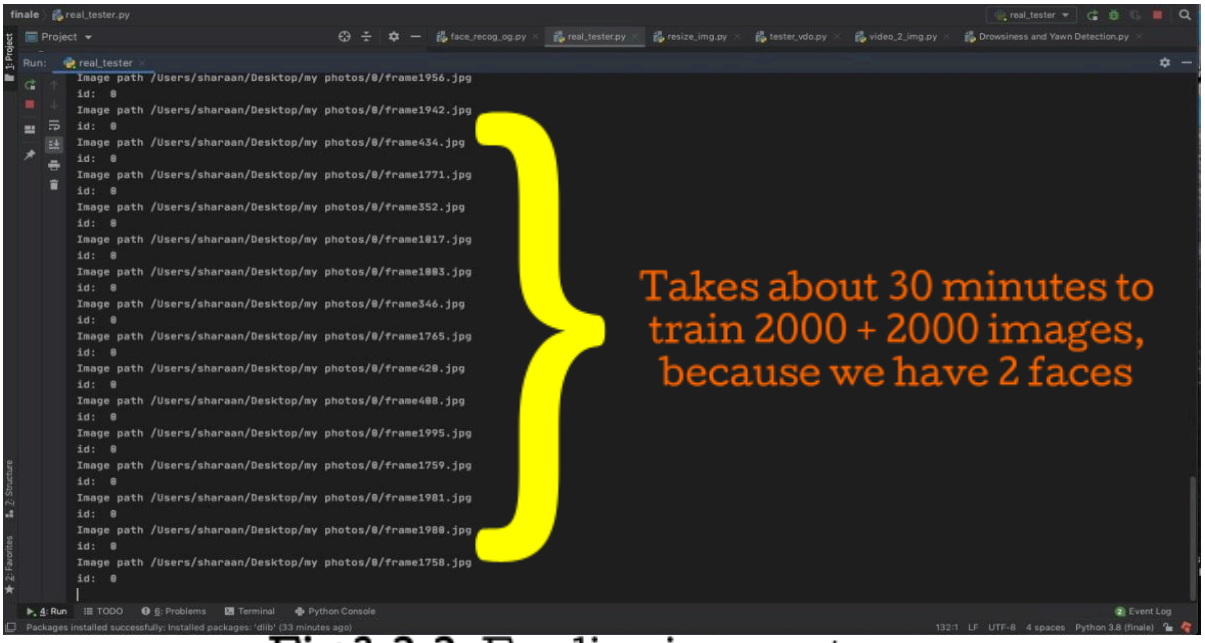


Fig 3.2.2: Feeding images to my model to train it to recognize faces

For Drowsiness and Yawn Detection:

- **For Drowsiness Detection**:

  We will be making use of **Shape Predictor Tool** (as discussed in pages 17) i.e.: **shape_predictor_68_face_landmarks.dat** to detect, localize and trace facial regions/parts through the video capturing frame.

  The **Shape Predictor Tool** estimates the location of 68 co-ordinates (x, y)that maps the facial points onto a human's face**.** The regions/parts of the face are mentioned as below :-

✔ Jaw = 1-17
✔ Left eyebrow = 18-22
✔ Right eyebrow = 23-27
✔ Nose = 28-36
✔
✔ **Left eye = 37-42**
✔ **Right eye = 43-48**

} - The Landmark region points of interest for **Drowsiness Activity Detection**

✔ Outer lip = 49-60
✔ Inner lip (Gap) = 61-68

} - The Landmark region points of interest for **Yawning Activity Detection**

For this project, we are only interested in Left eye, Right eye and Outer, Inner lip and we would omit the rest of facial region detections by passing only the points of the eyes and lips through a numpy array.

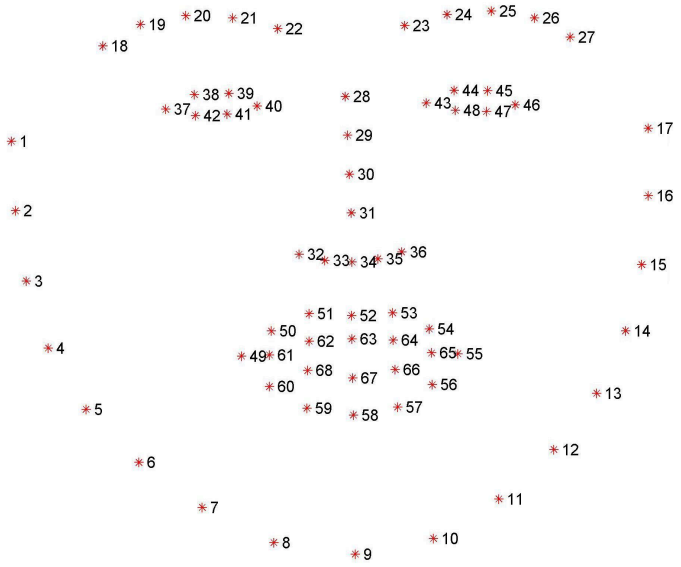


Fig 3.2.1: 68 –point Landmark Detection of a human face by the Shape Predictor tool of Dlib library

Detecting Drowsiness:

- After extracting only the user's both eyes from the facial landmark tool, we will have to do some processing.
- Here, we use the concept of **EYE ASPECT RATIO(EAR)** to detect the Drowsiness.
- The procedure is mentioned below.

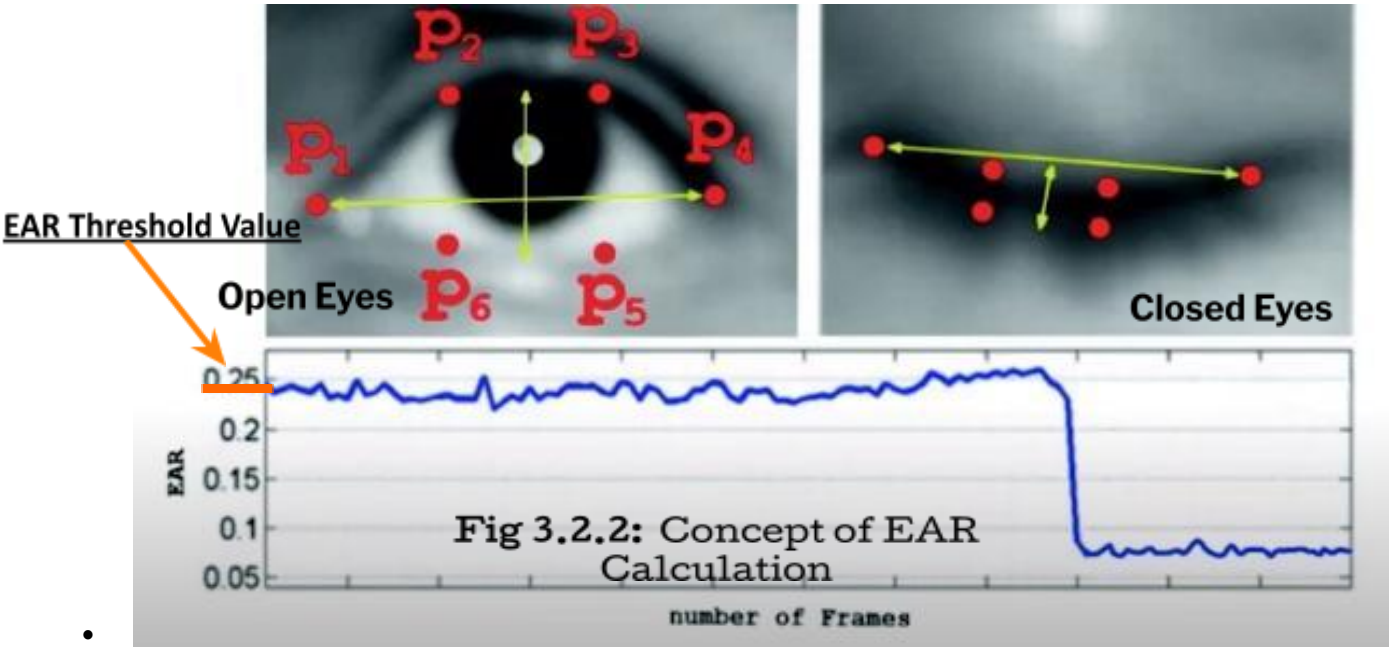


Fig 3.2.2: Concept of EAR Calculation

After analyzing and obtaining all 6 points of an eye, we will be drawing a **CONVEX HULL** i.e.: a boundary around all the adjacent 6 points on the face's both the eyes to completely detect it

When **eye is open**, value of EAR is high/more and if eyes are closed, EAR value is dropped and becomes low.

Ideally (also, in this graph), 0.25 is the ideal threshold value of EAR after calculation, below which, if EAR value falls, the Drowsiness will be detected by a ringing alarm sound, to alert wake up the person.

Formula For Calculating EYE ASPECT RATIO (EAR)

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Fig 3.2.3: EAR Ratio Formula

The 6 points i.e. **p1, p2, p3, p4, p5, p6,** are the 6 points obtained by Shape Predictor by detected & tracing the each individual human eye

Modulus in the EAR formula is basically the **Euclidean Distance** b/w those 2 respective points.

Fuzzy Based Model for Drowsiness Detection:

Fuzzy model prepared for the drowsiness detection system has been prepared to enhance performance and accuracy. It has two input variables and an output. This LF/HF (Low frequency band power (LF) (Represent parasympathetic nerve system of the human body) and High Frequency band power (HF) was analyzed and found out the range of the ratio of LF/HF which relevant to the drowsiness states.

Fuzzy Inputs:

Eye shut duration was measured using the frames of the video. For every 50 frames of the video, number of frames, which have not detected Iris, has been monitored and the count has been used as an input. According to the range this inputs has been categorize into three fuzzy sets.

1. Normal
2. Average
3. Danger

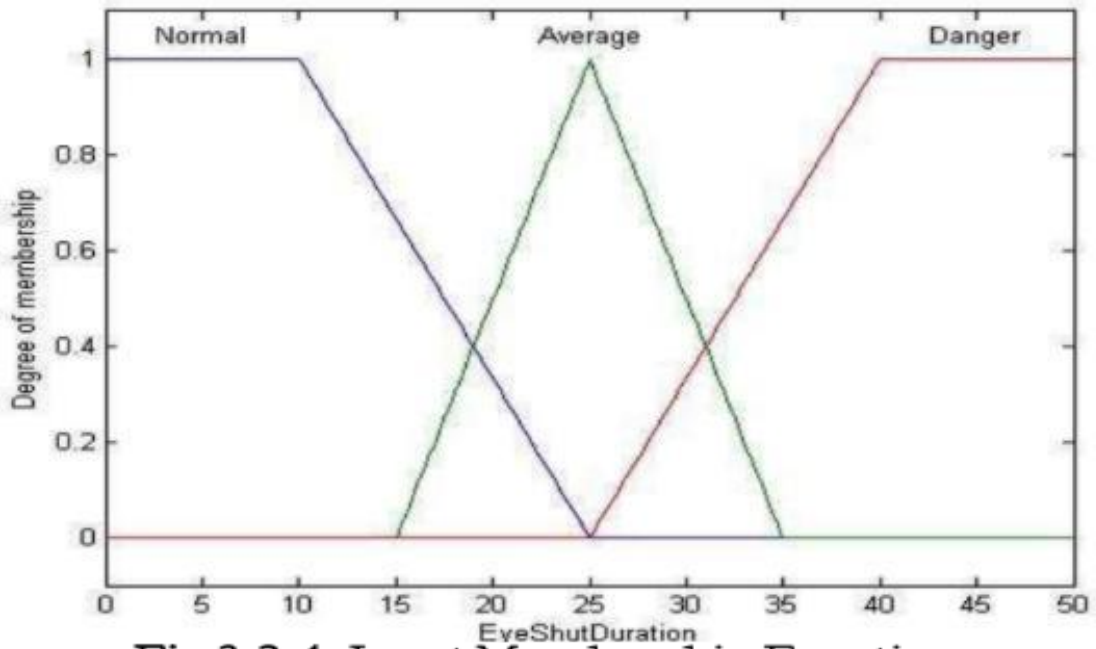


Fig 3.2.4: Input Membership Function– Eye Shut Duration

Fuzzy Output:

The output of the system is drowsiness level. We proposed three main stages of drowsiness levels to observe. The outputs are given from range of indexes from 0 to 1. They are as follows,

1. Good

2. Observe

3. Alarm

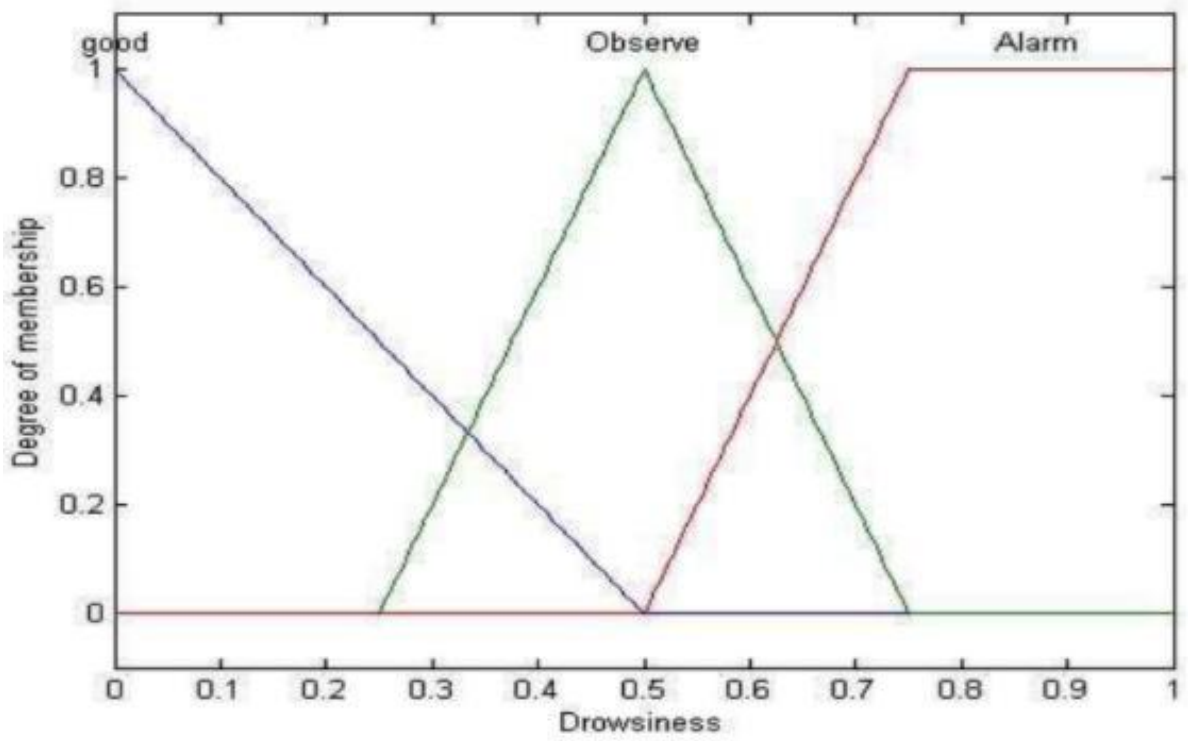


Fig 3.2.5: Output Membership Function- Drowsiness Level

Defining Fuzzy Rules:

After testing the fuzzy system for couple of subjects we came up with seven fuzzy rules to the fuzzy system. When defining the rules most concern was given to the eye shut duration since when eye shut duration is high there is a high probability of drowsiness. The seven rules are as follows,

1. If Eye Shut Duration is danger, then Drowsiness is Alarm.
2. If Eye Shut Duration is average and LF/HF is low, then Drowsiness is Alarm.
3. If Eye Shut Duration is normal and LF/HF is mid, then Drowsiness is good.
4. If Eye Shut Duration is normal and LF/HF is high, then Drowsiness is good.
5. If Eye Shut Duration is average and LF/HF is high, then Drowsiness is good.
6. If Eye Shut Duration is average and LF/HF is mid, then Drowsiness is observe.
7. If Eye Shut Duration is normal and LF/HF is low, then Drowsiness is observe.

		Eye Shut Duration		
		Normal	**Average**	**Danger**
LF/HF Ratio	**Low**	Observe	Alarm	Alarm
	Mid	Good	Observe	Alarm
	High	Good	Good	Alarm

Fig 3.2.6: Fuzzy Rules Tables based on **output membership-function** for Drowsiness Detection system

✔ For Yawn Detection:-

With the **Shape Predictor Tool** which we made use of previously we will be detecting the Gaping width or distance between the upper lip and the lower lip.
The regions/parts of the face are mentioned as below: -

✔ Jaw = 1-17
✔ Left eyebrow = 18-22
✔ Right eyebrow = 23-27
✔ Nose = 28-36
✔
✔ Left eye = 37-42
✔ Right eye = 43-48

} - The Landmark region points of interest for **Drowsiness Activity Detection**

✔ **Outer lip = 49-60**
✔ **Inner lip (Gap) = 61-68**

} The Landmark region points of interest for **Yawning Activity Detection**

Detecting Yawning Activity:

✔ After extracting only the user's top and bottom lip feature from the facial landmark tool, we will have to do some processing.

✔ We will have to calculate the distance between the upper and lower lip, called the gaping width, says here, X.

✔ Then, we will have to set a threshold value (minimum value of gaping width), for which would detect the activity of yawning of the person, and compare it with X.

✔ If the distance value X, is greater than threshold value, then yawning would be detected, ringing an automated speech/alarm, to alert and freshen up the driver (person) and make him active while driving.

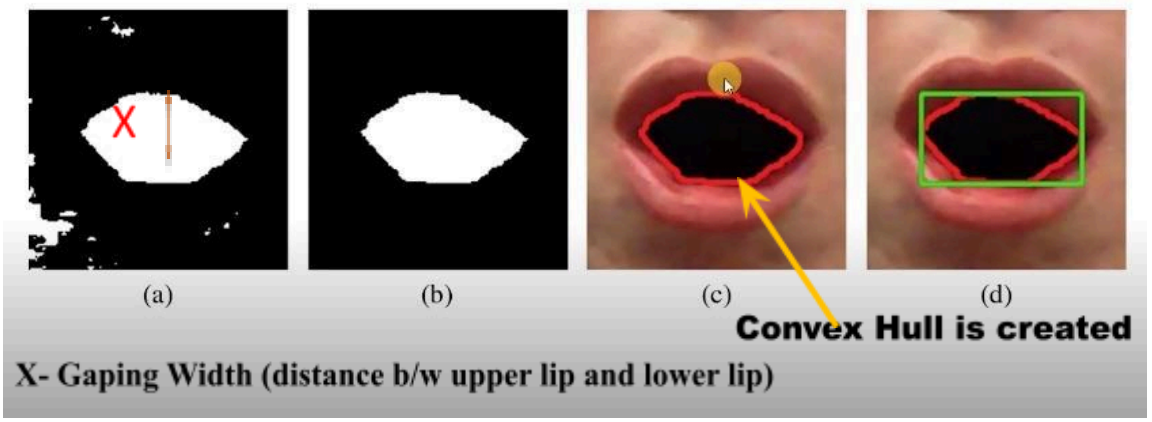


Fig 3.3.1: Gaping Width Calculation using Convex Hull

Final Steps for Cumulatively Detecting Drowsy and Yawn state simultaneously

- ✔ Importing necessary Libraries.
- ✔ Using EAR-Eye Aspect Ratio Concept to calculate the Minimum Eye and Mouth Width Threshold and detect DROWSINESS and YAWN respectively.
- ✔ Defining function to calculate EYE ASPECT RATIO.
- ✔ Extracting only Left and Right Eyes from total 68-point facial landmark Template and taking its average value.
- ✔ Defining function to calculate lip distance.
- ✔ Extracting Upper Lip and Lower Lip from total 68-point facial landmark Template.
- ✔ Taking input from command line
- ✔ Defining all thresholds for eye distance and mouth's lip width distance.
- ✔ Loading Predictor and Detector to detect the Drowsiness and Yawn Predictor.
- ✔ Loading input from WebCam.
- ✔ Resizing the Video Input frame and then checking the condition for Drowsiness and Yawning of Person, then ringing alert alarm to warn the person performing the same, if conditions are violated.

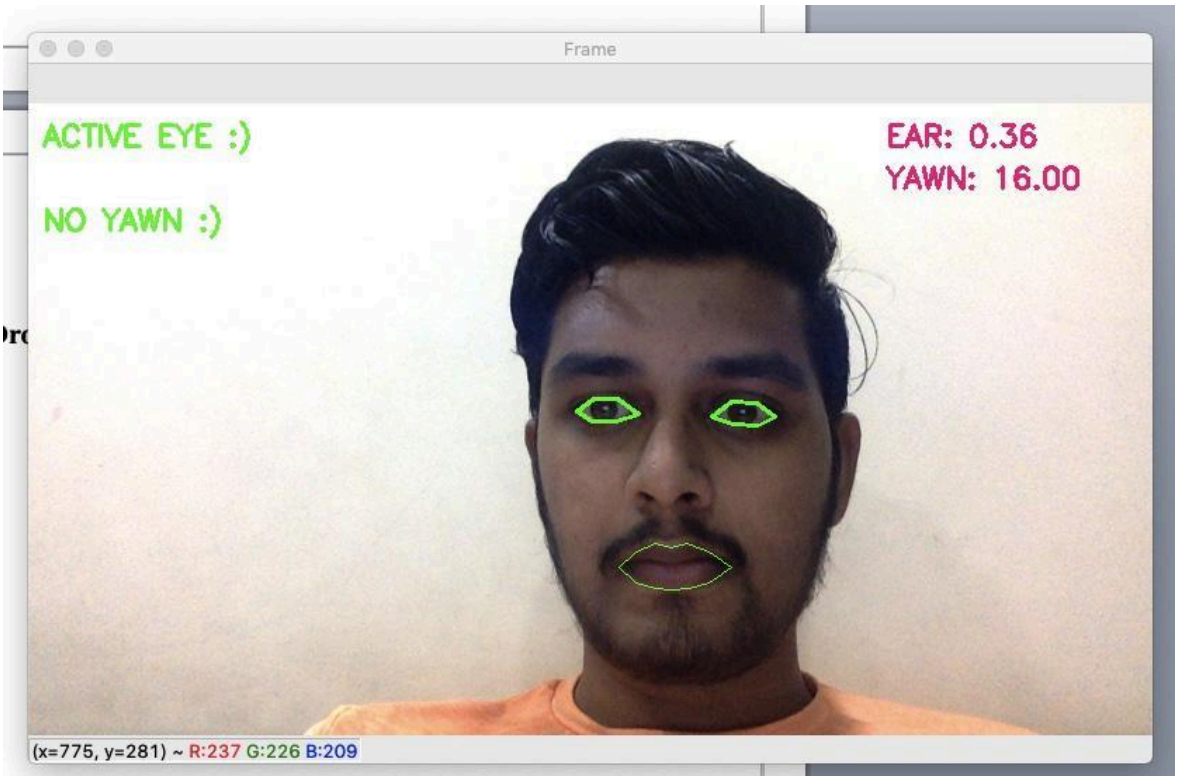
Output and Result Analysis

1) Detecting and analyzing Drowsiness Activity:

We have **0.31** as the ideal threshold value of EAR after calculation, below which, if EAR value falls, the Drowsiness activity will be detected, by an alert sound.

1) Without Spectacles:

Case 1: Non-Drowsy Person (with an active eyes) and **Straight Face**:



Result:

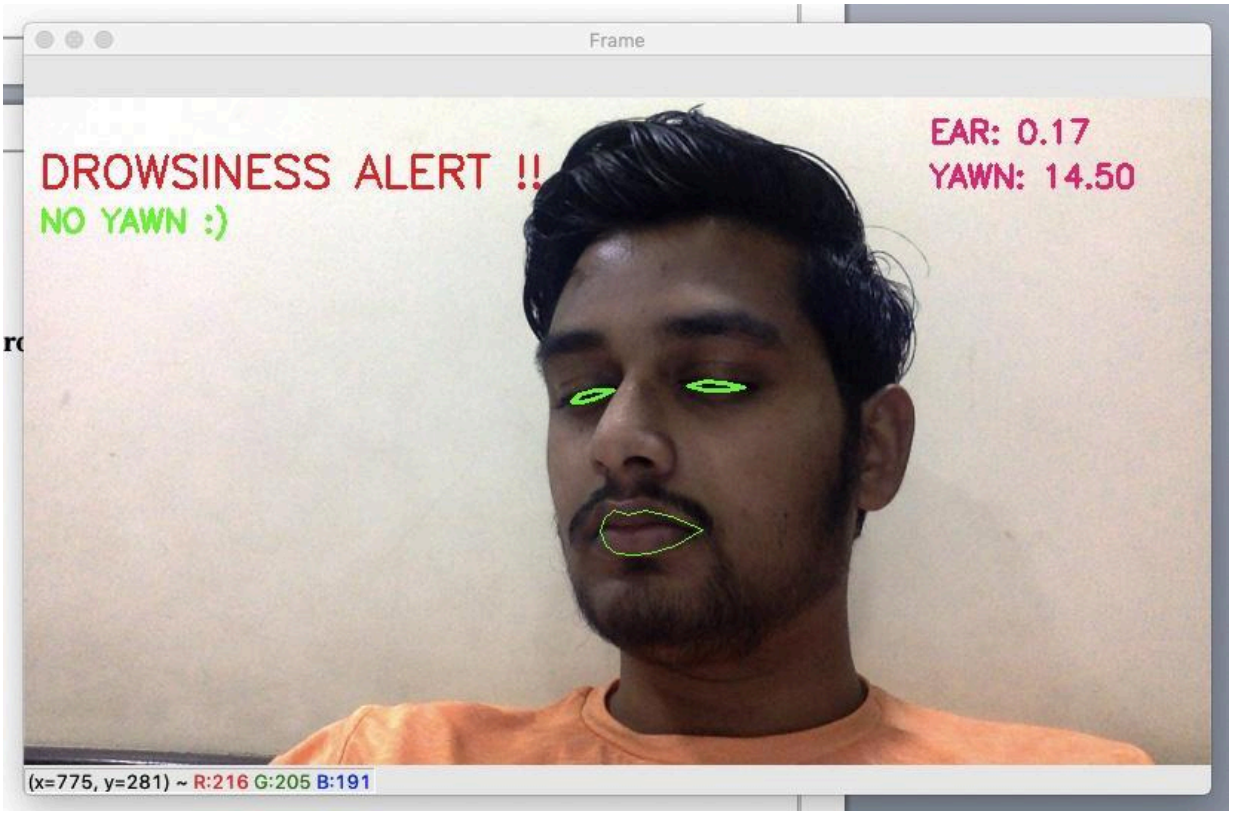
Yes, our application detected the scenario correctly as an **Active Eye**, i.e.: person is **not drowsy**, as the real-time calculated EAR value is **0.36** (greater than our EAR Threshold value- **0.31**)

Case 2: Drowsy Person (with inactive eye) and **Straight Face**:

Result:

Yes, our application detected the scenario correctly as an **Inactive Eye**, i.e.: person is **drowsy,** as the real-time calculated EAR value is **0.21** (less than our EAR Threshold value- **0.31**), and so, thereby alerted him.

Case 3: Drowsy Person (with inactive eye) and **Tilted Face**:

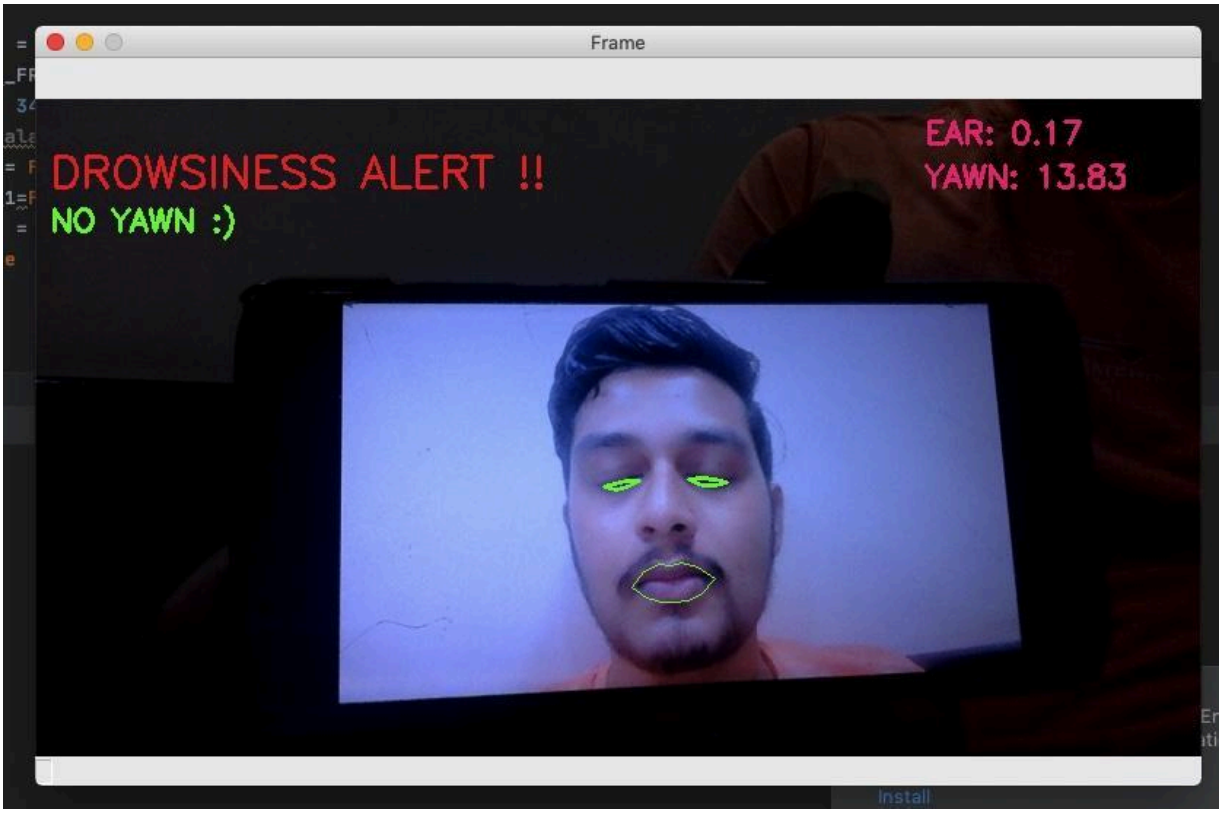


Result:

Yes, our application detected the scenario correctly, as **a Inactive Eye**, i.e.: person is **drowsy**, as the real-time calculated EAR value is **0.17** (less than our EAR Threshold value- **0.31**), and so, thereby alerted him.

Case 4:

<u>Detecting through mobile screen, the Drowsy Person (with inactive eye) and **Straight Face**:</u>



<u>Result:</u>

Yes, our application detected the scenario correctly, even through a mobile screen, as **an Inactive Eye**, i.e.: person is **drowsy**, as the real-time calculated EAR value is **0.17** (less than our EAR Threshold value- **0.31**), and so, thereby alerted him.

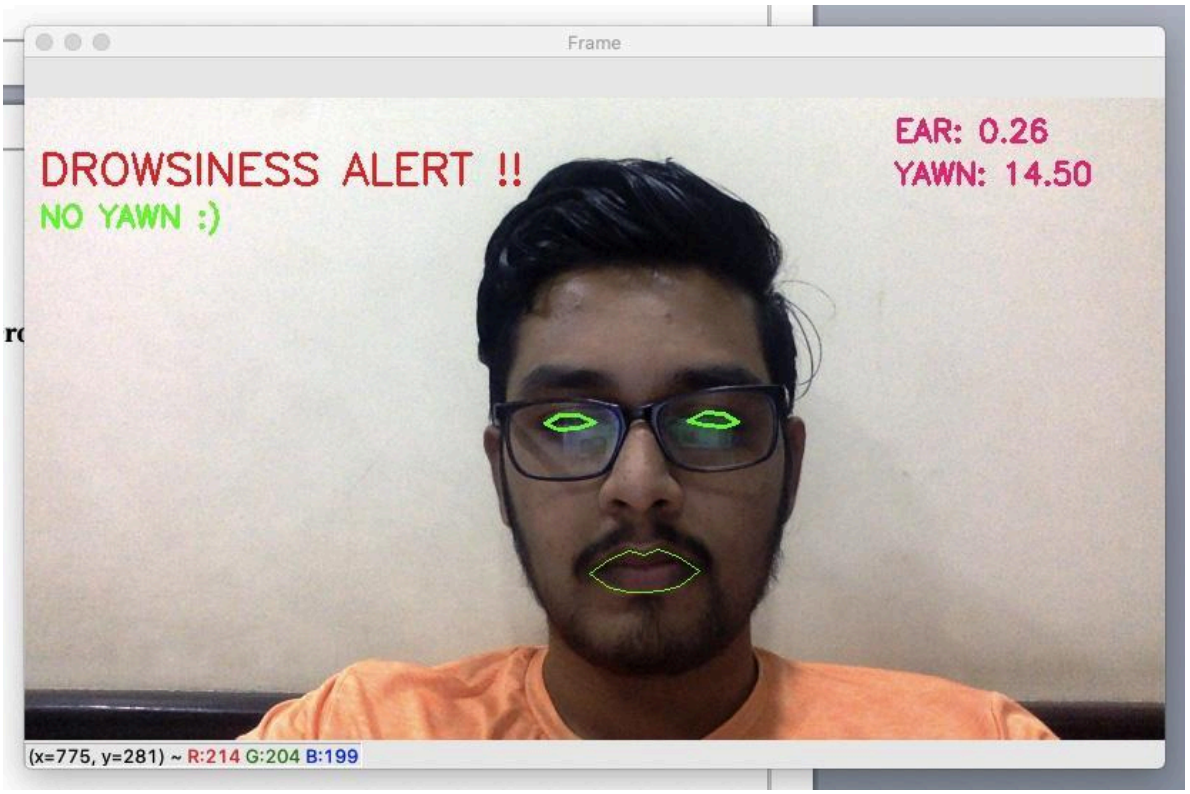
2) With Spectacles:

Case 1: Non-Drowsy Person (with an active eyes) and **Straight Face**:

Result:

Yes, our application detected the scenario correctly as an **Active Eye**, i.e.: person is **not drowsy**, as the real-time calculated EAR value is **0.32** (less than our EAR Threshold value- **0.31**)

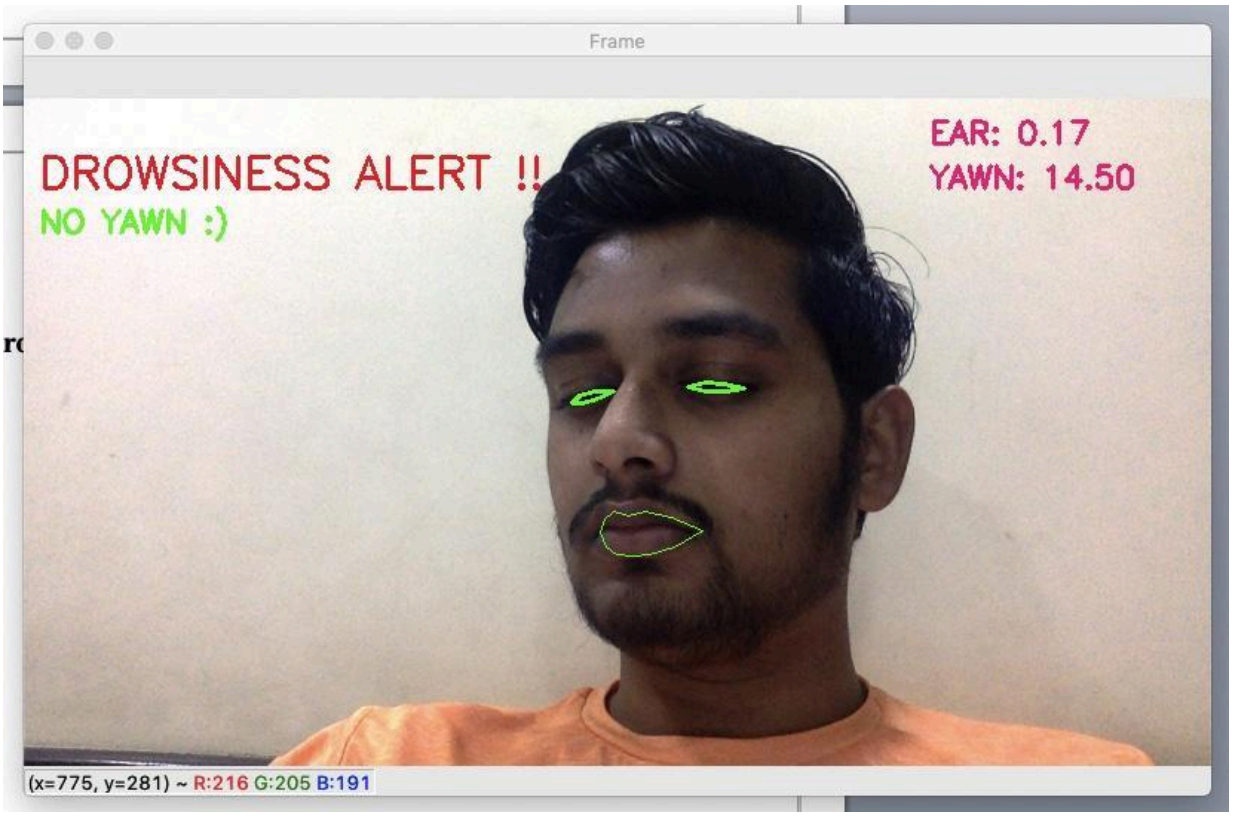
Case 2: Drowsy Person (with inactive eye) and **Straight Face**:



Result:

Yes, our application detected the scenario correctly as an **Inactive Eye**, i.e.: person is **drowsy**, as the real-time calculated EAR value is **0.26** (less than our EAR Threshold value- **0.31**), and so, thereby alerted him.

Case 3: Drowsy Person (with inactive eye) and **Tilted Face**:



Result:

Yes, our application detected the scenario correctly as an **Inactive Eye**, i.e.: person is **drowsy**, as the real-time calculated EAR value is **0.17** (less than our EAR Threshold value- **0.31**), thereby alerting him

Case 4:

Detecting through mobile screen, the Drowsy Person (with inactive eye) and **Straight Face**:

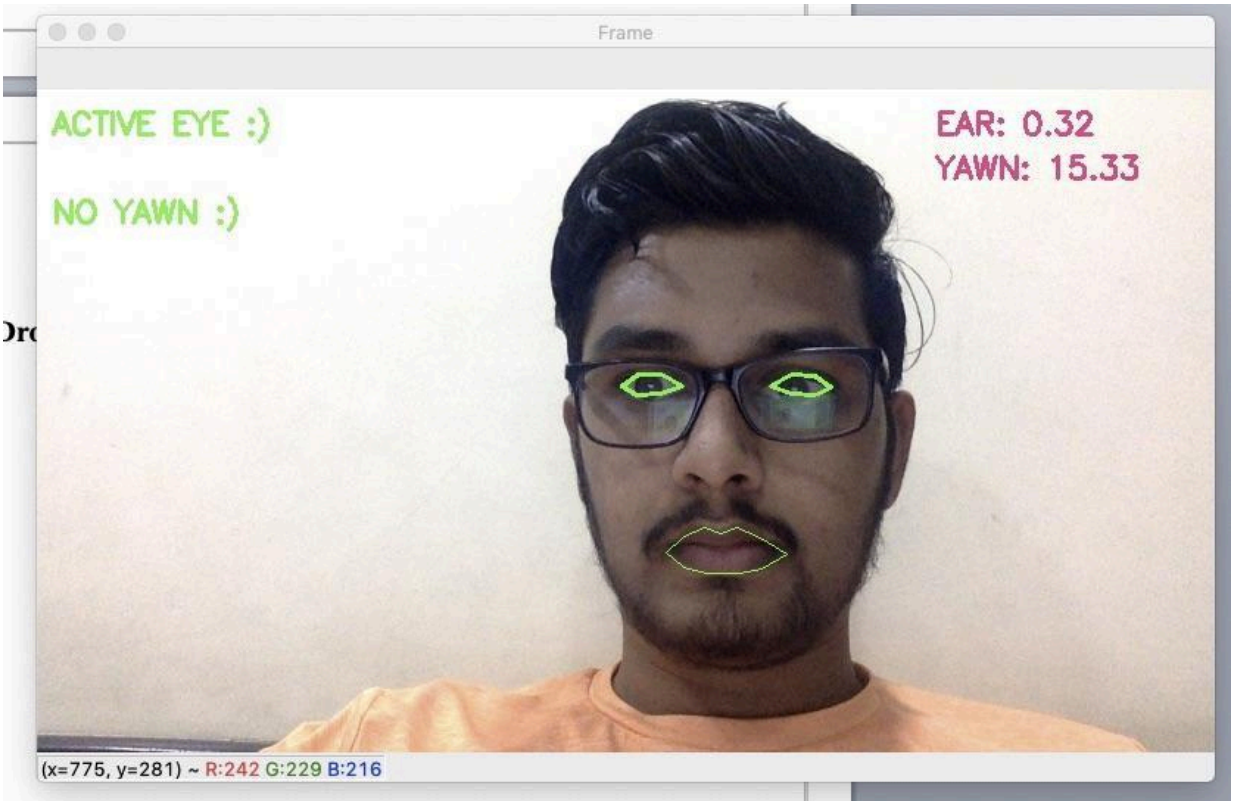
Result:

Yes, our application detected the scenario correctly, even through a mobile screen, as **an Inactive Eye**, i.e.: person is **drowsy**, as the real-time calculated EAR value is **0.22** (greater than our EAR Threshold value- **0.31**), and so, thereby alerted him.

2) Detecting and analyzing Yawn Activity:

We have **34 units** as the ideal threshold value of Gaping width (the width between upper lip and lower lip at that instant) after calculation, above which, if the width rises, the Yawn activity will be detected, by an alert sound.

Case 1: No Yawning Activity and **Straight face**:



Result:

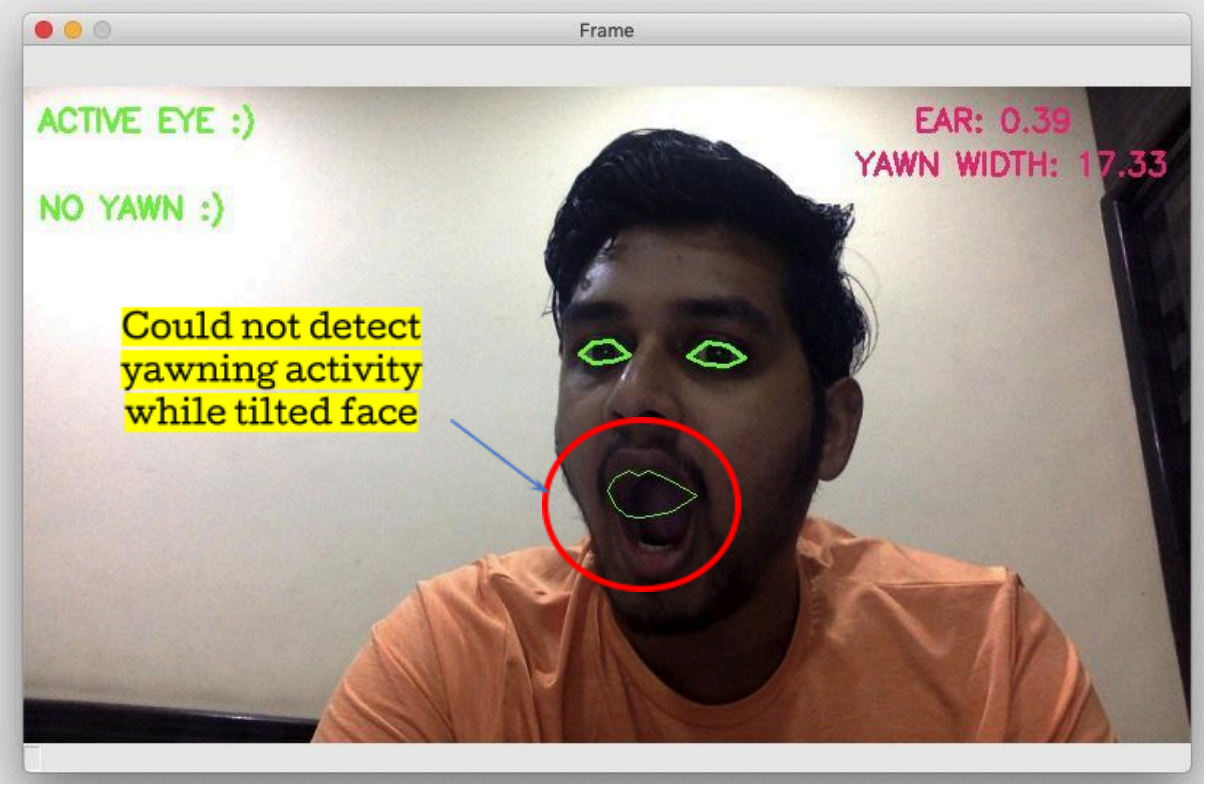
Yes, our application detected the scenario correctly, i.e.: the person is **expresses no yawning activity**, as the real-time calculated Gaping width value is **15.33 units** (lesser than our Yawning Threshold value- **34 units**)

Case 2: Yawning Activity performed and **Straight face**:

Result:

Yes, our application detected the scenario correctly, i.e.: the person is performing a **Yawning activity**, as the real-time calculated Gaping width value is **36.83 units** (greater than our Yawning Threshold value - **34 units**), and so, thereby alerted him.

Case 3: Yawning Activity performed and **Tilted face**:

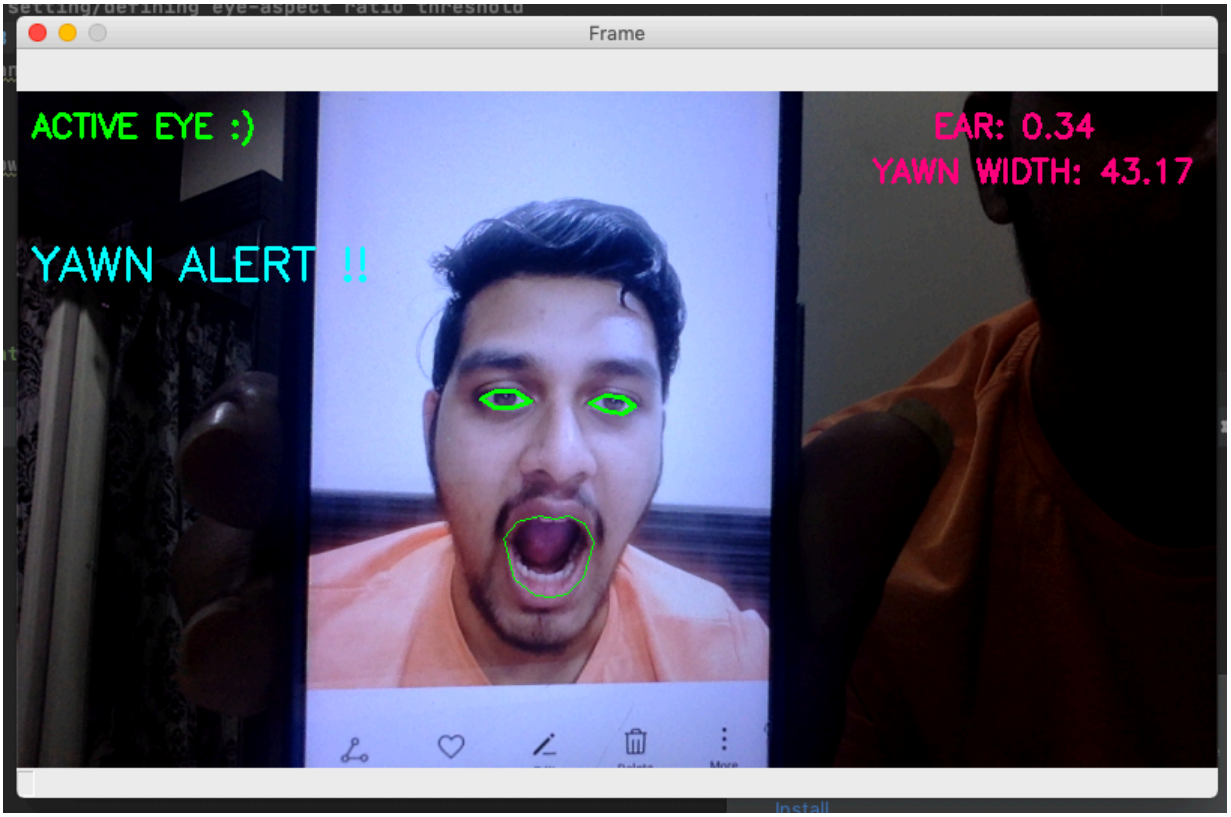


Result:

NO, our application detected is not detecting scenario correctly, i.e.: even though the person is performing a **Yawning activity**, the system is not able to detect it.

Case 4:

Detecting through mobile screen, the Yawning Activity and **Straight Face:**



Result:

Yes, our application detected the scenario correctly, i.e.: the person is performing a **Yawning activity**, even through a mobile screen, as the real-time calculated Gaping width value is **46.17 units** (greater than our Yawning Threshold value - **34 units**), and so, thereby alerted him.

3) Simultaneously detecting and analyzing Drowsiness & Yawn Activity:

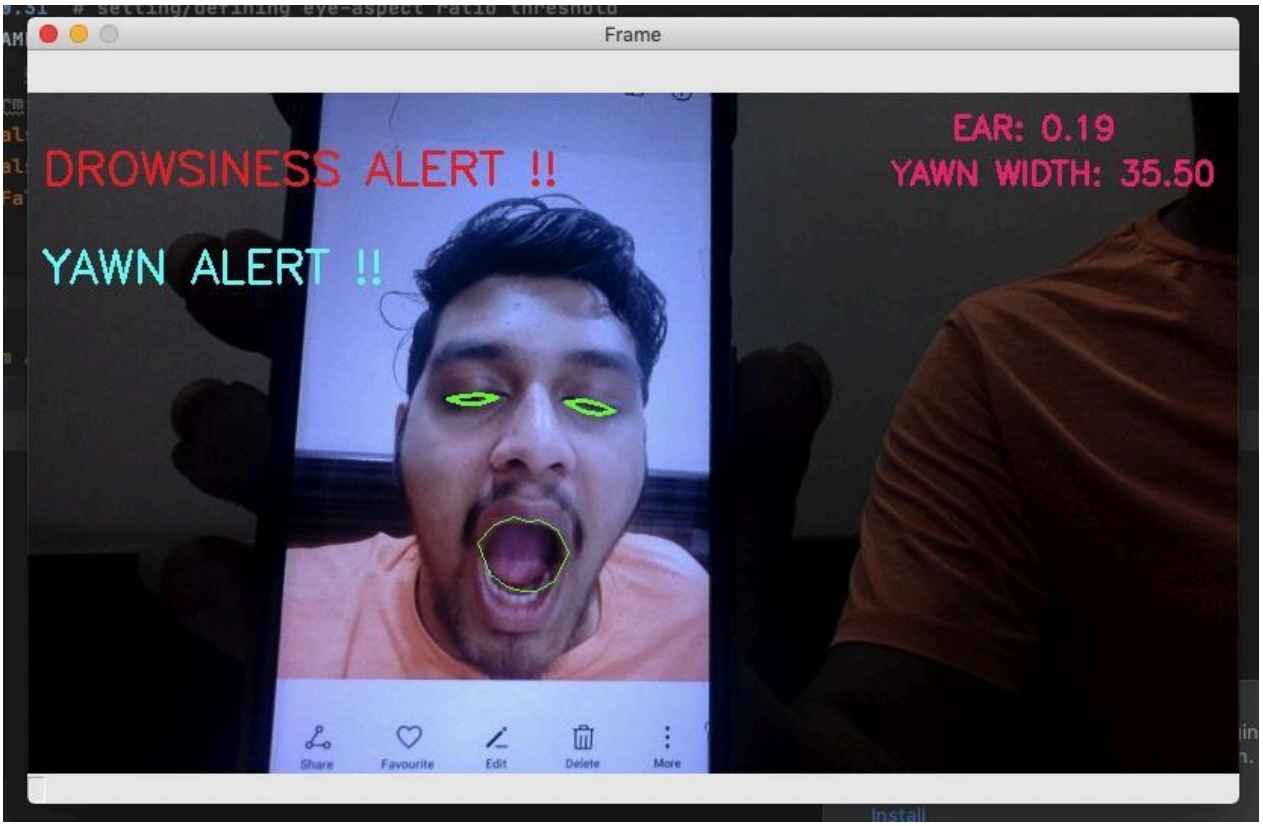
1) Without Spectacles:

Case 1: Detecting our final application through Laptop's Webcam:

Result:

Yes, our application detected the scenario correctly, i.e.: **simultaneously detected Drowsiness and Yawn activity, without spectacles,** at the same time without one affecting the execution of other, by checking the thresholds assigned for respective detectors.

Case 2: Detecting our final application through mobile screen:



Result:

Yes, our application detected the scenario correctly, i.e.: **simultaneously detected Drowsiness and Yawn activity, without spectacles through a mobile screen** at the same time without one affecting the execution of other, by checking the thresholds assigned for respective detectors.

2) With Spectacles:

Case 1: Detecting our final application through Laptop's Webcam:

Result:

Yes, our application detected the scenario correctly, i.e.: **simultaneously detected Drowsiness and Yawn activity, even with spectacles,** at the same time without one affecting the execution of other, by checking the thresholds assigned for respective detectors.

Case 2: Detecting our final application through mobile screen:

Result:

Yes, our application detected the scenario correctly, i.e.: **simultaneously detected Drowsiness and Yawn activity, even with spectacles, through the mobile screen** at the same time without one affecting the execution of other, by checking the thresholds assigned for respective detectors.

4) Accuracy of our model: -

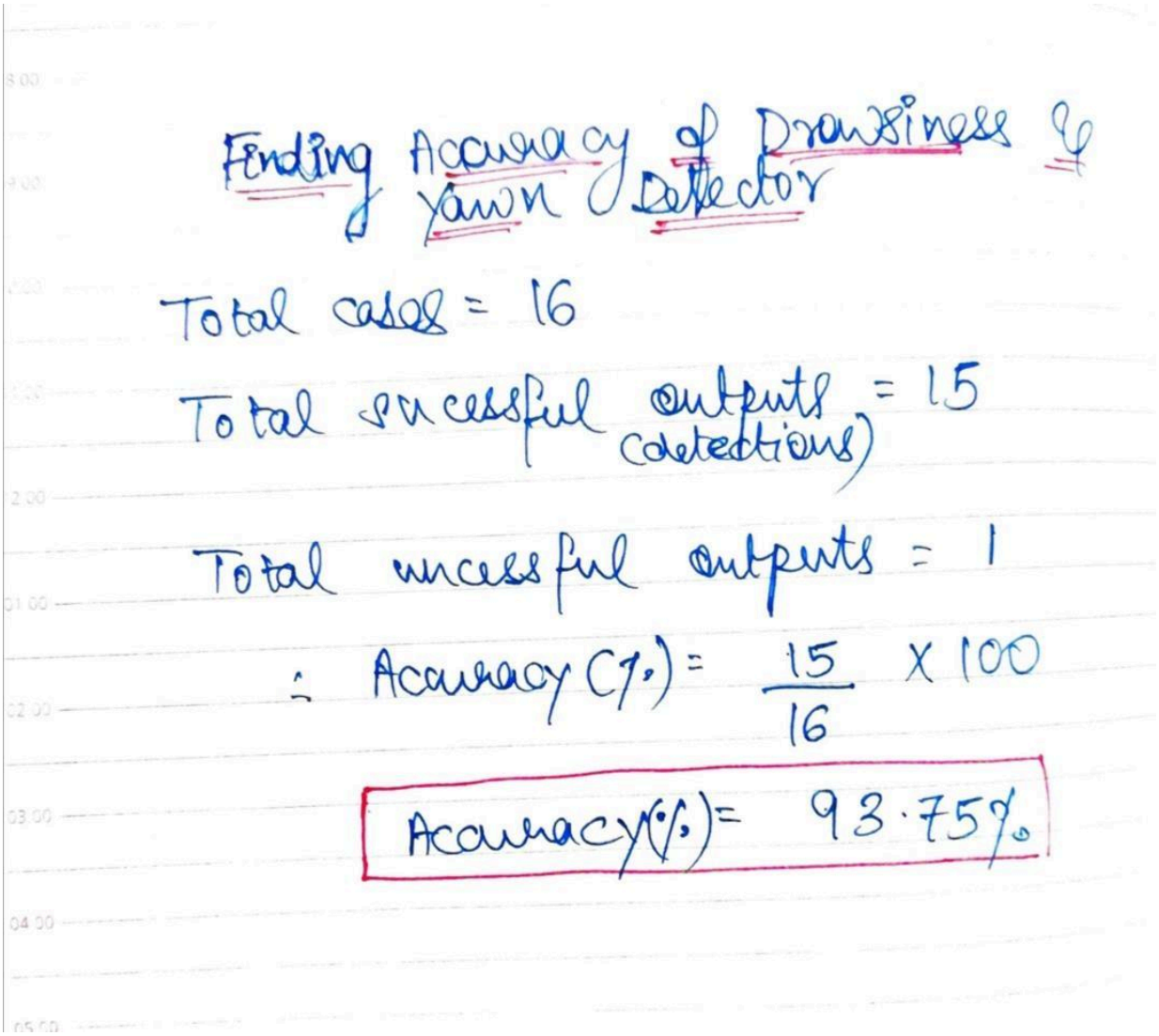


Fig 3.3.2: Determining Accuracy of our Drowsiness and Yawn Detector

4) Output For Face Recognition: -

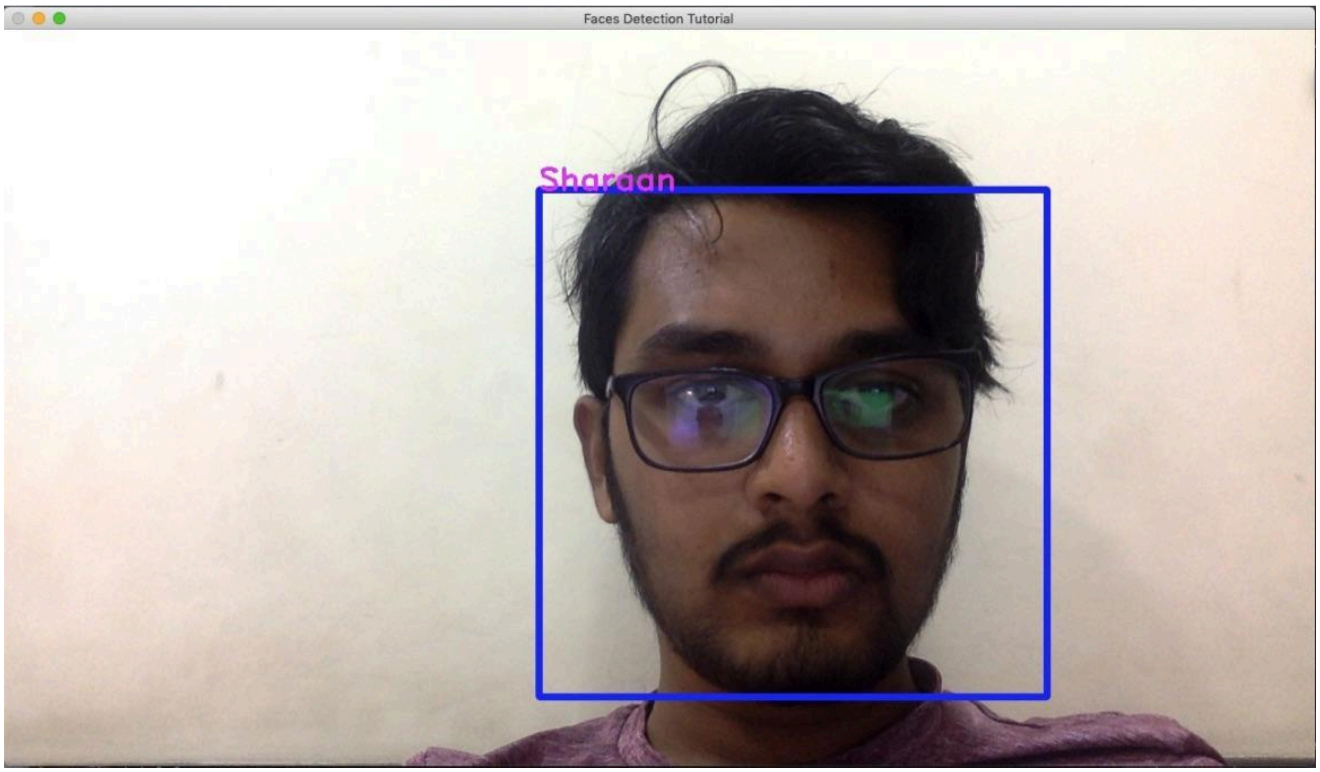


Fig 3.3.3: Output of only our Face Recognition model.

5) Combined final output of our application:-

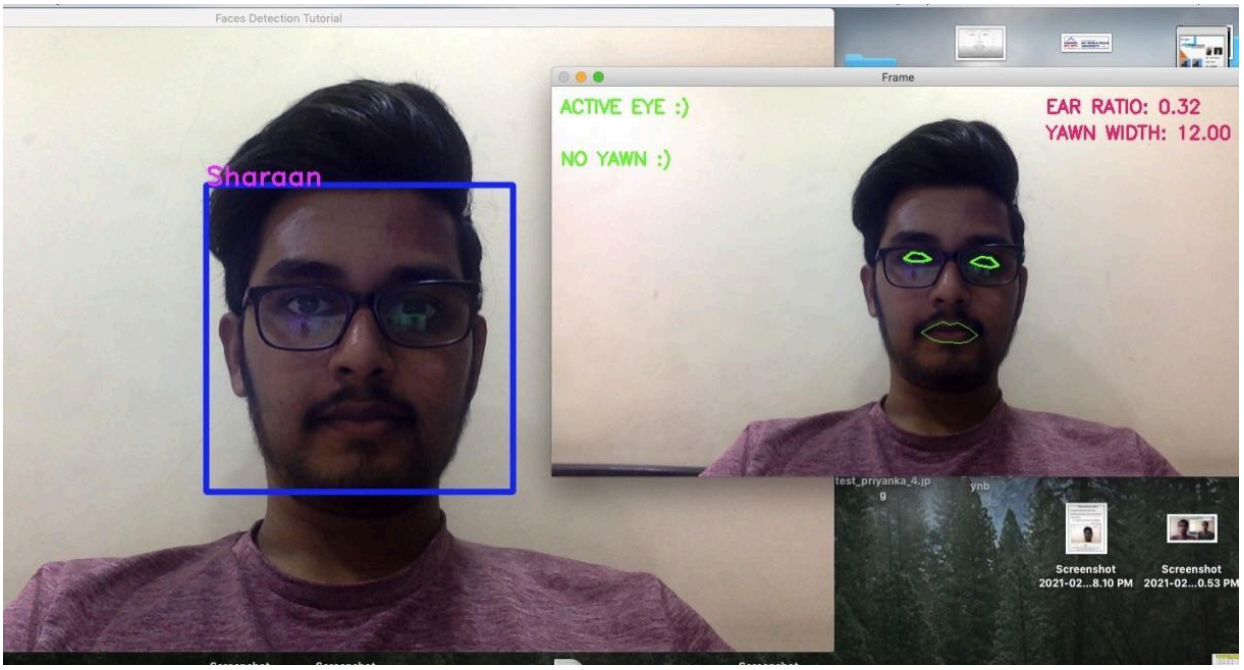


Fig 3.3.4: Output of our **Drowsiness and Yawn detector system with Face Recognition**

Real-Time Demonstration of my **Drowsiness and Yawn detector system with Face Recognition**

[Click here to see demonstration]

Please wait for 15-20 seconds after clicking the link, it would take some time in some PC's but will **surely open the Demonstration Video.**

https://www.youtube.com/watch?v=zQ3HieqW7BM&feature=youtu.be

If the above doesn't work, copy paste this link on Youtube to see Demonstration Video

Libraries used in this Project

- **OpenCV (Computer Vision) –**

  OpenCV is a library of programming functions mainly aimed at real-time computer vision. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. In our project, we have used this library to load the Haar Cascade Classifier and perform basic preprocessing operations on the input (video) frame obtained through the WebCam.

- **Dlib –**

  Dlib is a toolkit for making real world machine learning and data analysis applications in C++. Here, I have used it to load our Shape Predictor Tool to detect facial landmark regions.

- **Scipy –**

  Open-source Python library, which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.

- **Argparse –**

  Argparse is the “recommended command-line parsing module in the Python standard library.” It's what you use to get command line arguments into your program

- **ImUtils –**

  Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, and displaying Matplotlib images easier with OpenCV and Python interpreters.

- **Numpy –**

  Adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays, here performing operations on pixel values of image obtained as an input through camera.

- **Time –**

  Used to get/obtain the current time and date from the system.

- **Pyttsx3 -**

  Python Package to convert Text To Speech. This library is used in our project to fire an alert sound/warning ring of speech saying “wake up sir” if the driver drowsiness is detected, or “please stop yawning sir”, if the driver yawning activity is detected, thereby to awake the person/driver to be active while driving.

- **Threading –**

  To perform Threading functions: wait, sleep etc. in our program and also to perform drowsiness and yawning detection simultaneously (Multithreading).

Software requirements

✔ **Language**: Python Programming
✔ **Interpreter Version**: 3.7, 3.9 Interpreter
✔ **IDE Used**: Pycharm

Hardware requirements

✔ Laptop/PC for program execution (MAC)

CHAPTER 4

Conclusion

The Drowsiness and Yawn detection and correction system developed is capable of detecting drowsiness and yawning activity in a rapid manner. The system, which can differentiate normal eye blink and drowsiness which can prevent the driver from entering the state of sleepiness while driving. To be on a safer note, if the system also detects yawning activity by an abnormal increase in the size of driver's mouth width for a certain amount of time, and with the combination of machine learning algorithms, it can predict whether the driver is sleepy or not. The system works well even in case of drivers wearing spectacles and under low light conditions also. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for about two seconds, the alarm beeps to alert the driver and the speed of the vehicle is reduced. The same is for yawning activity detection. Using drowsiness and yawn

detection system, driver safety can be implemented in normal cars also. it takes

care of the issue of stressing out for individuals having fatigue-related issues to inform them about the drowsiness level and yawn activity while driving. By doing this many accidents will be reduced which will in turn, provide safe life to the driver and vehicle safety. A system for driver safety and car security is presented only in the luxurious costly cars.

Face recognition is an emerging technology that can provide many benefits. In conclusion, in our research, after preprocessing the input face images using some advanced image processing techniques such Contrast Adjustment, Bilateral Filter, Histogram Equalization, so as to have better image features and the same advanced image processing techniques will be applied to the training/template face images plus an image blending method to ensure high quality training/template face images.

Cumulatively, after combining both these applications/software's, we obtain a prototype which can be embedded onto a mobile application, called as **Real-time drowsiness and yawn detector system with Face Recognition,** where it will first recognize the driver/person who is driving (if face present/stored in database) as soon as he enters his vehicle, keep his track of route through the GPS, that he would be connected to, from his the social media account(Google) and then go on detecting for his drowsy or yawning state(also during alcoholism), and if no action taken by driver, notify the nearest road safety officials or traffic police officers to take necessary actions against him before he takes a toll of lives of number of other road user's, thereby preventing occurrence of any mishap.

Future Scope

The future works may focus on the utilization of outer factors such as **vehicle states, sleeping hours, weather conditions, mechanical data**, etc., for **fatigue measurement.** Driver drowsiness and yawning activity (to detect sleep beforehand), pose a major threat to highway safety, and the problem is particularly severe for commercial motor vehicle operators. Twenty-four hour operations, high annual mileage, exposure to challenging environmental conditions, and demanding work schedules all contribute to this serious safety issue. Monitoring the driver's state of drowsiness, his yawning activity and vigilance and providing feedback on their condition so that they can take appropriate action is one crucial step in a series of preventive measures necessary to address this problem. Currently there is not adjustment in zoom or direction of the camera during operation. Future work may be to **automatically zoom in on the eyes once they are localized**.
The model can also be incrementally improved by adding parameters like **blink rate & state of the car extra** etc. Further, we can plan to work on the project by adding a sensor to track the heart rate in order to prevent accidents caused due to sudden heart attacks to drivers.

Face recognition is a both challenging and important recognition technique. Among all the biometric techniques, face recognition approach possesses one

great advantage, which is its **user-friendliness (or non-intrusiveness).** In this paper, we have given an introductory survey for the face recognition technology. We have covered issues such as the generic framework for face recognition, factors that may affect the performance of the recognizer, and several state-of-the-art face recognition algorithms.

References

[1] Belhumeur, P., hespanha, J., and Kriegman, D., "Eigenfaces vs. Fisherfaces: Recognition

[2] COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHOLKOPF-A. BLAKE'S. ROMDHANI, AND P. TORR.

[3] OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI

[4] https://ieeexplore.ieee.org/document/6602353

[5] https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[6] https://www.pandasecurity.com/en/mediacenter/panda-security/facial-recognition-technology/

‘

School of Electronics and Communication Engineering

Academic Year 2020 - 2021

T. Y. Mini Project - I Trimester VIII

SYNOPSIS

Title of Project : Real Time Face Recognition using Python in OpenCv.

Domain of my Project : AI, Machine and Deep learning

Guide Name : Prof. Shweta Pawar

Batch Co-ordinator Name : Prof. Shweta Pawar

Name of Student : Sharaan Thayanithi

Roll No:	PRN No:	Name of Student	Basket	Contact No:	Email
PA 53	S1032181645	Sharaan Thayanithi	AI	**9137225214**	sharaan2001@gmail.com

Introduction:

Face detection is a computer vision technology that helps to locate/visualize human faces in digital images. This technique is a specific use case of object detection technology that deals with detecting instances of semantic objects of a certain class (such as humans, buildings or cars) in digital images and videos. With the advent of technology, face detection has gained a lot of importance especially in fields like photography, security, and marketing.

Face recognition systems **capture an incoming image from a camera device** in a two-dimensional or three-dimensional way depending on the characteristics of the device.

This biometric facial recognition procedure requires an internet connection since the database cannot be located on the capture device as it is hosted on servers.

In this comparison of faces, it **analyses mathematically** the incoming image without any margin of error and it **verifies that the biometric data matches** the person who must use the service or is requesting access to an application, system or even building.

OpenCV uses a type of face detector called a **Haar Cascade classifier**. Given an image, which can come from a file or from live video, the face detector examines each image location and classifies it as "Face" or "Not Face." Classification assumes a fixed scale for the face, say 50x50 pixels. Since faces in an image might be smaller or larger than this, the classifier runs over the image several times, to search for faces across a range of scales.

The classifier uses data stored in an XML file to decide how to classify each image location. The OpenCV download includes four flavors of XML data for frontal face detection, and one for profile faces. It also includes three non-face XML files - one for full body (pedestrian) detection, one for upper body, and one for lower body

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones .

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

Objectives / Aim of the project

- **IMPROVED PUBLIC SECURITY** : ensures no duplicity or misuse/mishandling of the records, and can be used to detect frauds ,burglars etc

  - Efficient and accurate retrieval of person's data based on his face, thereby making it unique from other verification process

- **IMPROVED PUBLIC SECURITY** : ensures no duplicity or misuse/mishandling of the records, and can be used to detect frauds ,burglars etc

- **Fast and Non-Invasive Identity Verification-RETAIL**:
  People don't have to remember their login credentials and time time taken is also very less compared to manual login,

Block Schematic:

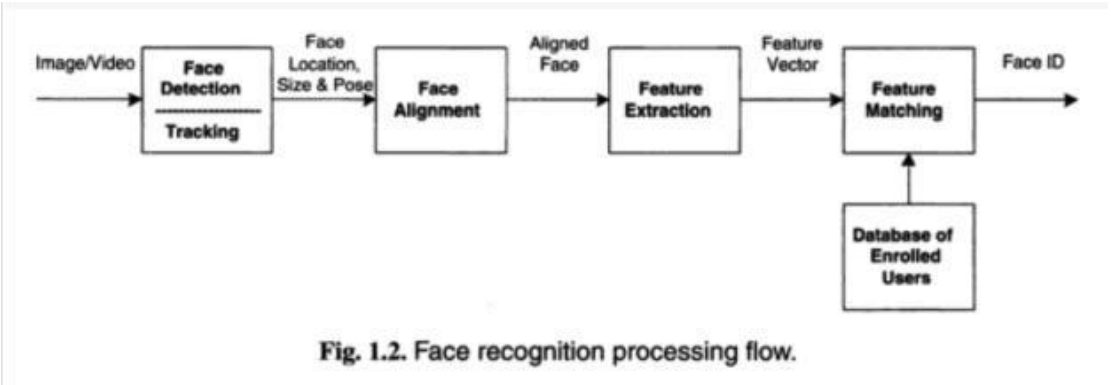


Fig. 1.2. Face recognition processing flow.

ALGORITHM

1. **Face Detection**. Locate one or more faces in the image and mark with a bounding box.
2. **Face Alignment**. Normalize the face to be consistent with the database, such as geometry and photometrics.
3. **Feature Extraction**. Extract features from the face that can be used for the recognition task.
4. **Face Recognition**. Perform matching of the face against one or more known faces in a prepared database.

A given system may have a separate module or program for each step, which was traditionally the case, or may combine some or all of the steps into a single process.

METHODOLOGY

- First, look at a picture and find all the faces in it
- Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.
- Third, be able to pick out unique features of the face that you can use to tell it apart from other people— like how big the eyes are, how long the face is, etc.
- Finally, compare the unique features of that face to all the people you already know to determine the person's name.

When image quality is taken into consideration, there is a number of factors that influence the system's accuracy. It is extremely important to apply various image pre-processing techniques to standardize the images that you supply to a face recognition system, like it should be "lumination dependent", and there are also many other issues, such as the face should also be in a very consistent position within the images (such as the eyes being in the same pixel coordinates), consistent size, rotation angle, hair and makeup, emotion (smiling, angry, etc), position of lights (to the left or above, etc).

SOFTWARE REQUIREMENTS

Software used: Python (3.7 Interpreter) – Pycharm , MS Excel

Libraries used: OpenCV, Matplotlib, Numpy

Modules used: face_recognition, dlib

HARDWARE REQUIREMENTS

Nil

References:

https://machinelearningmastery.com/introduction-to-deep-learning-for-face-recognition/#:~:text=Face%20recognition%20is%20often%20described,extraction%2C%20and%20finally%20face%20recognition.

https://www.pyimagesearch.com/done-code-conf/

https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

youtube.com/cleverprogrammer

https://www.youtube.com/watch?v=R7B8sSByZGQ&t=7134t

https://www.geeksforgeeks.org/opencv-python-tutorial/

https://www.sas.com/en_in/insights/analytics/machine-learning.html

https://www.expert.ai/blog/machine-learning-definition/

https://ieeexplore.ieee.org/abstract/document/5344047

Author:

Zhiming Qian

Chuxiong Normal University, Chuxiong, Yunnan, China

Dan Xu

Department of Computer Engineering, Yunnan University, Kunming, China

Publisher: IEEE

Published in: Chinese Conference on Pattern Recognition

Publication Year: 2010

Signature of Student's Signature of Guide Signature of Batch Coordinator

Details of Participation in Project Competition

Sr. No.	Name & Place of project competition / Exhibition	Date	Certificate / Prizes won (if any)

(Attach Xerox of certificate/s)

Details Paper publication / Presentation

Sr. No.	Name of the organizing society	Date	Certificate / Prizes won (if any)
