ONLINE EXAM MONITORING SYSTEM

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF **BACHELOR OF ENGINEERING** IN **COMPUTER SCIENCE AND ENGINEERING** OF THE ANNA UNIVERSITY

MINI PROJECT WORK

Submitted by

DIVYA A

1817116

HAFEEZA BEGUM M

1817118

KEERTHANA K

1817123

PAVITHRA T

1817137

2022

Under the Guidance of

Dr.K.KUMAR M.E., Ph.D

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF TECHNOLOGY

(An Autonomous Institute affiliated to Anna University)

COIMBATORE - 641 013

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University)

COIMBATORE - 641 013

MINI PROJECT WORK

JANUARY 2022

This is to certify that this project work entitled

ONLINE EXAM MONITORING SYSTEM

is the bonafide record of project work done by

DIVYA A
1817116

HAFEEZA BEGUM M
1817118

KEERTHANA K
1817123

PAVITHRA T
1817137

of B.E.(COMPUTER SCIENCE AND ENGINEERING) during the year 2021 - 2022

Project Guide
Dr.K.KUMAR M.E.,Ph.D

Head of the Department
Dr.J.C.MIRACLIN JOYCE PAMILA M.E., Ph.D

Submitted for the Project Viva-Voce examination held on ________________

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We express our sincere gratitude to **Dr.P.THAMARAI, Ph.D.,** Principal, Government College of Technology, Coimbatore for providing us all facilities that we needed for the completion of this project.

We whole-heartedly express our thankfulness and gratitude to **Dr.J.C.MIRACLIN JOYCE PAMILA, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering, Government College of Technology, for helping us to successfully carry out this project.

Our thankfulness and gratitude to our respectable project guide **Dr.K.KUMAR, M.E., Ph.D.,** Associate Professor (CAS) , Department of Computer Science and Engineering who has been an immense help through the various phases of the project. With his potent ideas and excellent guidance, we were able to comprehend the essential aspects involved.

We extend our sincere thanks to **Dr.S.RATHI, M.E., Ph.D.,** Professor (CAS), **Prof.T.RAJA SENBAGAM, M.E.,** Assistant Professor, **Dr.A.MEENA KOWSHALYA, M.E., Ph.D.,** Assistant Professor , **Dr.R.BHAVANI, M.E., Ph.D.,** Assistant Professor , **Dr.R.MUTHURAM, M.E., Ph.D.,** Assistant Professor , **Prof.L.SUMATHI, M.E.,** Assistant Professor and **Prof.C.GAYATHRI, M.E,** Assistant Professor for all their valuable suggestions to the completion of the project.

We thank our fellow classmates M.Hafeeza Begum , S.Mirnalini , A.Divya for helping us in the testing phase of the project.

We thank all the non-teaching staff and our friends for their cooperation towards the successful completion of the project.

We would like to dedicate the work to our parents for their constant encouragement throughout the project.

SYNOPSIS

Owing to pandemic, most of the institutions are in a situation to use virtual mode of education. Institutions are unable to monitor student's activity precisely through online, especially when it comes to exam. The quality of education is spoiled because of unfair practices by the students during examinations. The invigilator should have complete coverage of the students to conduct exams in a sincere manner. The ability to efficiently proctor remote online examinations is an important limiting factor to the scalability of this next stage in education. Presently, human proctoring is the most common approach of evaluation, by either requiring the test taker to visit an examination center. However, such method is labor-intensive and costly. By combining continuous identity verification and automatic detection of malpractice or suspicious activities by a student, this system provides a scalable, online, automated, human interaction free proctoring system that can be accessed by test takers and administrators to a truly efficient solution to conventional problem of online exam proctoring.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Traditional assessment refers to standardized testing that uses questions with a limited number of answer choices. It includes multiple choice, true or false and some short answer responses. Traditional assessment evaluates the learning and retaining capacity of a child. It analyses how much of the provided material or syllabus has been acquired by the student. It also helps educators or teachers to compare the performances of different students. The teacher gets a preview of a student's knowledge conveniently. It assesses a student's learning through a set of questions curated as per the specified syllabus. Exams and tests conducted are pen and paper-based. The answers to every question pertain to a particular subject and do not fluctuate as per the opinions of the people, hence the evaluation by the teachers is also elementary and straight forward. Students develop their retaining and understanding capabilities. They also learn to recognize and reconstruct their intellect and build their cognitive abilities. This approach is overall more simple, straightforward and time-saving. The teachers are efficiently able to manage more children in a lesser amount of time. It is quiet, reliable and fixed. Traditional assessments do not require extra tools and hence is very economical. A simple pen and paper can be used in the procedure.

The pandemic caused further shocks to the system with schools forced to shut down during the lockdown period and the transition of students and teachers to online teaching-learning. In India, around 250 million students were affected due to school closures at the onset of lockdown induced by COVID-19. COVID-19 also acted as a catalyst for digital adoption in school education. Education institution faced struggles in conducting exams during this pandemic and they conducted online exams for students. Proctoring during online exams is a challenging factor for the institutions. As in conventional exams, a proctoring process during the exam is needed to verify whether the student who took the exam is valid.

The Proposed method is to create an automated proctoring system where the user can be monitored automatically through the webcam. Before Starting the Examination the student face is detected and Verified. Monitoring the student during online examination includes Eyeballs tracking, Mobile Phone Detection, Count the number of person in the frame.

1. Face Detection - Detecting one face in an image and extracting the most important features from the face.
2. Face Verification - A facial recognition system uses biometrics to map facial features from a photograph or video. It compares the information with a database of known faces to find a match. Facial recognition can help verify a person's identity.
3. Eye Tracking - Track eyeballs and report if candidate is looking left, right or up.
4. Mobile Phone Detection - Find and report any instances of mobile phones.
5. Person Counting - Instance segmentation to count number of people and report if no one or more than one person detected.

CHAPTER 2
LITERATURE REVIEW

In literature various techniques has been proposed by researchers to proctor the online examination.

[1] MTCNN or Multi-Task Cascaded Convolutional Neural Networks is a neural network which detects faces and facial landmarks on images. MultiTask Cascaded Convolutional Neural Network is a modern tool for face detection, leveraging a 3-stage neural network detector.

[2] VGG is an innovative object-recognition model that supports up to 19 layers. Built as a deep CNN, VGG also outperforms baselines on many tasks and datasets outside of ImageNet. VGG is now still one of the most used image-recognition architectures. VGG incorporates 1x1 convolutional layers to make the decision function more non-linear without changing the receptive fields. The small-size convolution filters allows VGG to have a large number of weight layers.

[4] Cosine Similarity Metric Learning (CSML) for learning a distance metric for facial verification. The use of cosine similarity leads to an effective learning algorithm which can improve the generalization ability of any given metric.

[5][6] OpenCV is a python library which is used to create a real time I detector with the help of Dlib.it reads the frame through webcam.it uses numpy array and also thresholding for Masking.

[7] YOLO is a Convolutional Neural Network (CNN) for performing object detection in real-time.It allows the model to look at the whole image at test time, so its predictions are informed by the global context in the image. YOLO and other convolutional neural network algorithms “score” regions based on their similarities to predefined classes.High-scoring regions are noted as positive detections of whatever class they most closely identify with.

[8] Haar Cascade is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features. The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them.

[9] A HOG is a feature descriptor generally used for object detection. HOGs are widely known for their use in pedestrian detection. A HOG relies on the property of objects within an image to possess the distribution of intensity gradients or edge directions. Gradients are calculated within an image per block.

[10] The face detection using advanced method deep neural network which uses deep learning frame work. Deep neural network model which is embedded with latest open cv and by using the deep learning model frame work which is weighted with some other files.

Table 2.1 Summary of the Face Detection Methods

Ref	**Technique**	**Advantages**	**Limitations**
[8]	Haar Cascade Face Detector	Simple Architecture	Doesn’t work on non-frontal images
[9]	Hog Face Detector in Dlib	Fastest method on CPU	Does not work for side face and extreme non-frontal faces,like looking down or up.
[10][3]	DNN Face Detector	Works for different face orientations-up,down,left,right,side-face.Works even under substantial occlusion.	It is extremely expensive to train due to complex data models.
[11]	Spatial Pyramid Pooling	It is robust to object deformations	Not well working in the high detail requirement identification task.
[12]	CNN Face Detector	Works for different face orientations.	Does not detect small faces as it is trained for minimum face size of 80×80.

CHAPTER 3
PROJECT DESIGN

3.1 PROBLEM STATEMENT

Student cheating during exams is a widespread phenomenon around the world. With the expansion of e-learning, exam monitoring becomes more difficult. Therefore, real-time monitoring is necessary to ensure the student's identity continuously throughout the evaluation period. An online exam Monitoring system that provides automatic and continuous monitoring.

3.2 OBJECTIVE

Objective of this project is to monitor the students during the online examination using the webcam. Most common way of malpractice done using Moblie phone , copying from the written note or getting help from other . All this malpractices done during examination can be detected using this monitoring process.

3.3 PROPOSED SYSTEM

Proposed System is to create an automated proctoring system where the student can be monitored automatically through the webcam. Student verification include face detection and face recoginition . The detected student face during the entry of examination is verified against the dataset . After verification the student is being continuously monitored during test hours . Eyeballs movement of the student is monitored and report whether the student looks up , right or left. Usage of mobile phones during the examination is detected and report the any instances of mobile phones . Getting the help from the other person is detected by Instance segmentation count the number of person and report if no one or more than one person detected.

3.4 PROJECT WORKFLOW

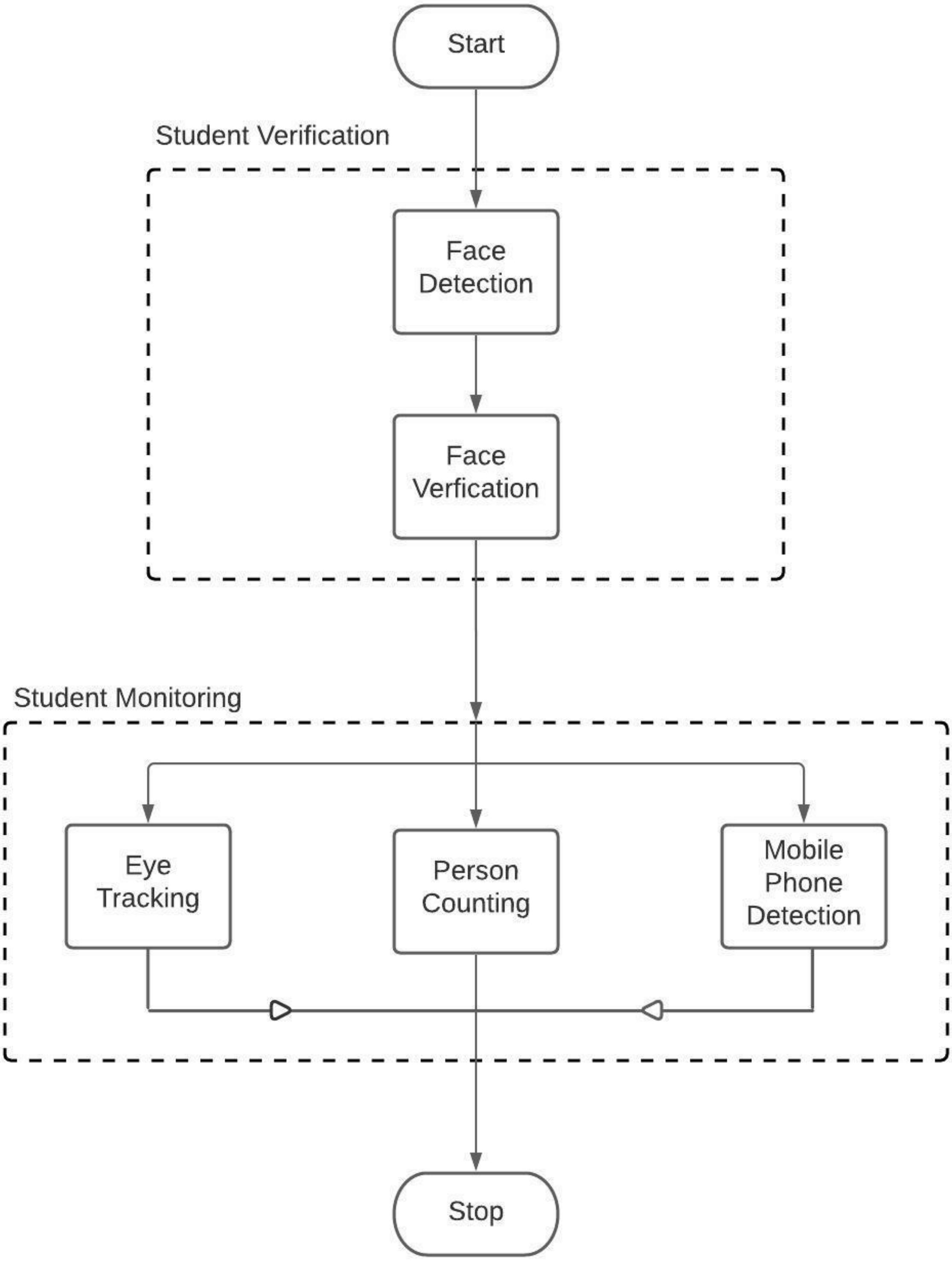

Fig.3.1 Workflow of student verification and monitoring system

3.5 MODULES

The modules identified and implemented in the proposed system are listed below:

1. Face Detection
2. Face Verification
3. Eye Tracking
4. Mobile Phone Detection
5. Person counting

Face Detection

Face Detection done using MTCNN Algorithm. It not only detects the face but also detects five key points as well. It uses a cascade structure with three stages of CNN. It use a fully convolutional network to obtain candidate windows and their bounding box regression vectors, and the highly overlapped candidates are overlapped using Non-maximum suppression (NMS). Next, these candidates are passed to another CNN which rejects a large number of false positives and performs calibration of bounding boxes. In the final stage, the facial landmark detection is performed.

Face Verification

Face verification is the task of deciding by analyzing face images, whether a student is who he/she claims to be. This is very challenging due to image variations in lighting, pose, facial expression, and age. The Face verification is done by computing the distance between two face vectors. Distance metrics are essential for face verification accuracy. Cosine Similarity Metric Learning (CSML) for learning a distance metric for facial verification.

Eye Tracking

Eye tracking is done by finding the face and then finding the eyes. The facial keypoint detector takes a rectangular object of the dlib module as input which is simply the coordinates of a face. To find faces we can use the inbuilt frontal face detector of dlib. After detecting the face, eye region is detected with the help of facial landmark features. Using the face landmarks dataset, we can point out 68 landmarks on the face. each landmark is assigned with an index. Using these indices, the desired region of the face is detected. After extracting eye region, it is processed for detecting eye movements. Eye tracking is done using openCV .

Mobile Phone Detection

Object detection is one of the major techniques that is used for locating the objects present in the given image or the video. The task of object detection uses mainly machine learning and deep learning methods for detecting accurate objects. YOLOv3 (You Only Look Once, Version 3) is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images. YOLO uses features learned by a deep convolutional neural network to detect an object.

Person Counting

Counting the humans present in the videos has various applications in intelligent systems. Vision-based object counting has multiple tasks involved in it like detection of a particular object, recognizing the object, and also tracking that object. Three categories can help in performing such tasks that are regression methods, clustering methods, and detection-based methods. The regression method uses a regression function by making the use of the regions that are used for detection, and this is used for the counting. The clustering method does the counting by tracking certain features for the discrete objects, and their trajectories are clustered and used for counting. The detection based method that uses tracking, trajectory, extraction for counting.

3.6 REQUIREMENTS

3.6.1 Required Software Environment

This section explains the software environment in which the algorithm executes. The details of the environment, libraries used are explained below.

Anaconda Prompt

Anaconda is popular because it brings many of the tools used in data science and machine learning. Like Virtual Environment, Anaconda also uses the concept of creating environments so as to isolate different libraries and versions. It provides more than 1500 Python/R data science packages which are suitable for developing machine learning and deep learning models.

GOOGLE COLABORATORY

Google Colaboratory is a free Jupyter notebook environment running wholly in the cloud to write and execute code in Python. Google Colab gives us about 13 GB of RAM and 25 GB of storage for free. Colab does not require a setup, plus the notebooks that we create can be simultaneously edited. The greatest advantage is that in Colab, the most popular machine learning libraries are already installed.

3.6.2 Libraries Required

The following libraries were used in building the model in python .

NumPy

NumPy (Numerical Python) is a linear algebra library in python. NumPy library is an important foundational tool for studying Machine Learning.

Pandas

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. Pandas works well with many other data science modules inside the Python ecosystem.

Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. It supports fast prototyping. It seamlessly runs on CPU as well as GPU.

TensorFlow

TensorFlow is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers.

OpenCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV, the CV is an abbreviation form of a computer vision, which is defined as a field of study that helps computers to understand the content of the digital images such as photographs and videos.

Scipy

SciPy is an 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.

CHAPTER 4

IMPLEMENTATION

4.1 Face Detection

MTCNN Algorithm is used for Face Detection. It uses a fully convolutional network to obtain candidate windows and their bounding box regression vectors, and the highly overlapped candidates are overlapped using Non-maximum suppression

4.1.1 MTCNN Algorithm

Multi-task Cascaded Convolutional Network is a Cascaded Network of three CNNs . The first stage has a Fully Connected Proposal Network that is used to obtain candidate windows and reduce the overlapping and number of boxes. The first stage takes as input an image pyramid made up of differently scaled copies of the input image. This provides the model with a wide range of window sizes to choose from, and helps the model in being scale invariant. The second stage is a CNN Refine Network(R-Net). It further reduces the number of boxes and merges overlapping candidates using non-maximum suppression (NMS). The Output Network in the third stage does more of the same things that R-Net does, and it adds the 5-point landmark of eyes, nose and mouth in the final bounding box containing the detected face.

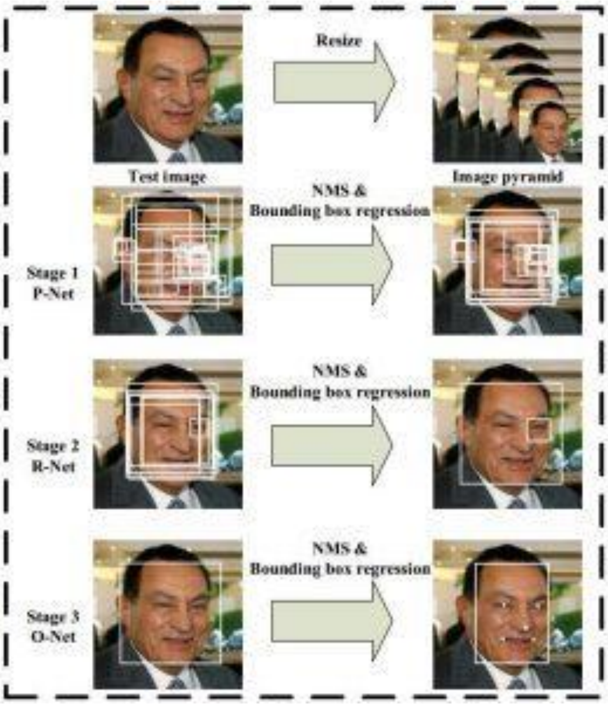

Fig 4.1 MTCNN Architecture

MTCNN breaks down the task into three stages and builds a pipeline as shown in fig.4.1.

- **Stage-1**: P-Net: In this stage, it produces candidate windows by a shallow convolutional network.
- **Stage-2**: R-Net: The objective in this stage is to reject as many non-face windows as possible. The network used here is deeper.
- **Stage-3**: O-Net: This uses even complex network to further refine the output of R-net.

The face is detected with bounding box and marking the five facial keypoints which are left eye, right eye, nose, left edge of mouth, right edge of mouth as shown in fig.4.2.

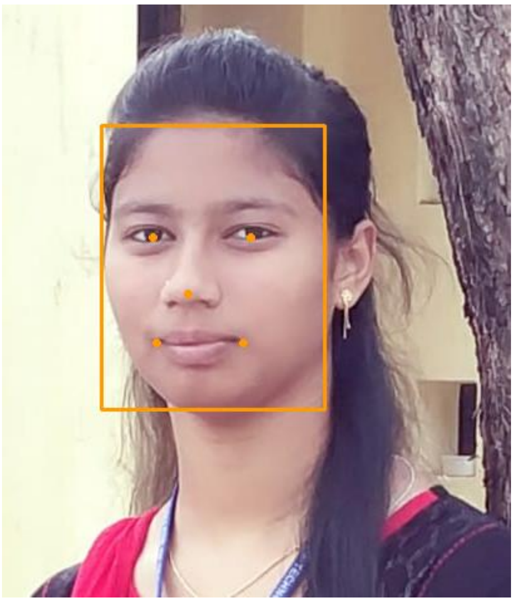

Fig 4.2 Detected face with Facial Keypoints

4.2 Face Verification

MTCNN used for face detection to find and extract faces from image. mtcnn library used to create a face detector and extract face. The result from the face extraction is a list of bounding boxes, where each bounding box defines a lower-left-corner of the bounding box, as well as the width and height. After the face detection and extraction, [2]VGGFace model used to perform face Verification.

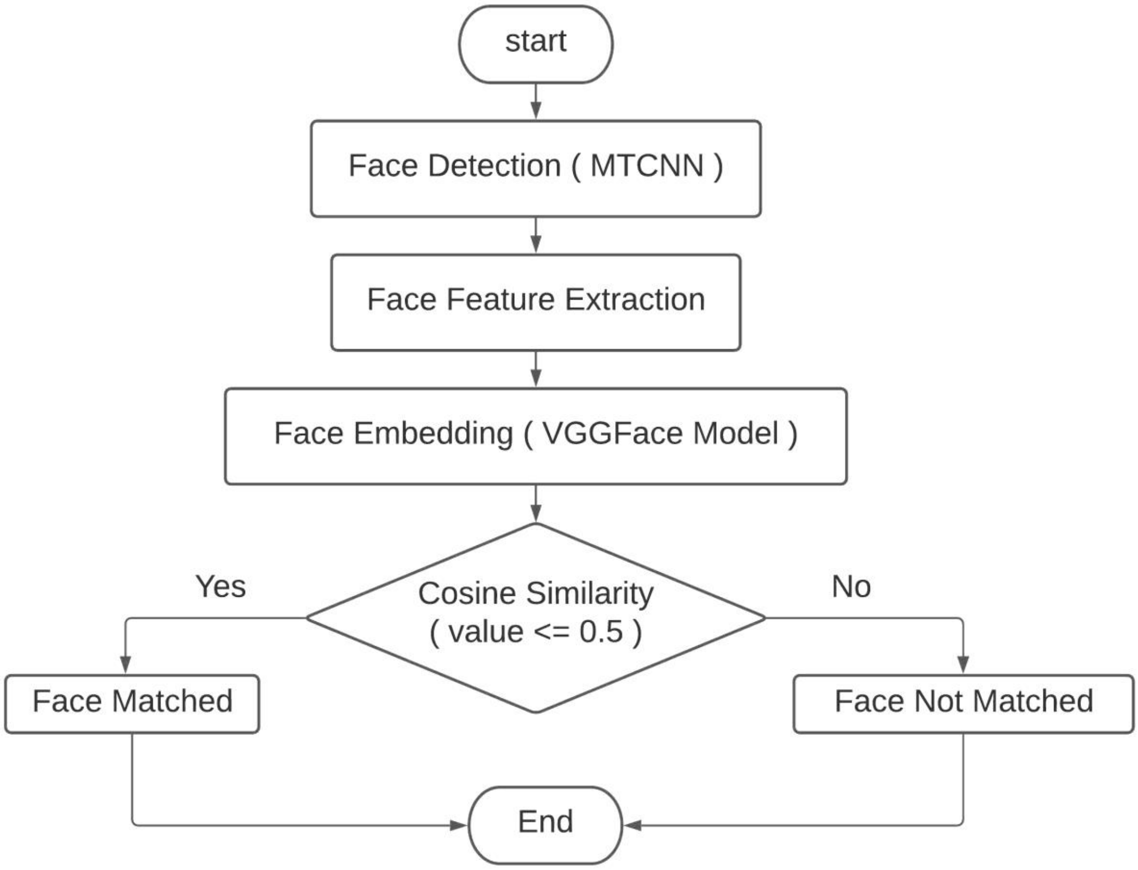

Fig 4.3 Flow Chart for Face Verification

4.2.1 Face Feature Extraction

Facial feature extraction is very much important for the initialization of processing the face recognition. In input image, face feature like nose, left eye, right eye, lips and mouth should be fully or partially visible. To extract the those features we use MTCNN.

4.2.2 Face Embedding

A face embedding is a vector that represents the features extracted from the face. This can then be compared with the vectors generated for other faces. For example, if a vector that is close to the compared vector then that may be the same person, if the vector that is deviated from the compared vector then that may be a different person. Cosine distance are calculated between two embeddings and faces are said to match if the distance is below a predefined threshold value <= 0.5.

4.2.3 Face Similarity

Face similarity is checked by using [4] Cosine Similarity Metric Learning. Cosine similarity measures the similarity between two vectors. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. The output can be a non-negative similarity score between 0 and 1, 0 or value nearer to 0 if the two images are completely similar to each other, otherwise 1.

$$\text{similarity}\,(x,y) = \cos(\theta) = \frac{x \cdot y}{|x||y|}$$

4.3 Eye Tracking

To find faces, The inbuilt frontal face detector of dlib facial key point detector is used which takes a rectangular object of a dlib model as input which is simply coordinates of face. It creates a new black mask using NumPy of the same dimensions as the webcam frame. Store the (x,y) coordinates of the points of the left and right eye from key points array shape and draw them on the mask. It takes an image, points as a NumPy array and color as arguments and returns an image with the area between those points filled with that color. The eye area is drawn in white, eye ball and remaining area are colored with black. Convert the result to grayscale to make the image ready for thresholding. Thresholding is used to create a binary mask. So, the task is to find an optimal threshold value against which can segment out the eyeballs from the rest of the eye and then we need to find its center. Threshold value will be different for different lighting conditions so we can make an adjustable trackbar for controlling the threshold value. The steps involved in eye tracking are ,

- Face and eye detecting
- Tracking both eyes individually
- Extracting features of eyes
- Tracking eye movements

4.3.1 Eye Tracking Process

Face tracker is not just tracking face visually. It serves as an important information source for the eye detection. It provides two important information.

1) Location of the face area

2) Center of the face image according to the whole image, thus we can calculate the width information of the face

A 3D rigid object has only two kind of motions with respect to a camera.

Translation :

Moving the camera from its current 3D location (X, Y, Z) to a new 3D location (X', Y', Z') is called translation. Translation has 3 degrees of freedom — we can move in the X, Y or Z direction. Translation is represented by a vector $\mathbf{t}$ which is equal to $(X' - X, Y' - Y, Z' - Z)$

Rotation :

Rotate the camera about the X, Y and Z axises. A rotation, therefore, also has three degrees of freedom. There are many ways of representing rotation. we can represent it using Euler angles (roll, pitch and yaw), a 3×3 rotation_matrix, or a direction of rotation (i.e. axis) and angle.

To calculate the 3D pose of an object in an image the following informations are needed.

2D coordinates of a few points

We need the 2D (x,y) locations of a few points in the image. In the case of a face, we could choose the corners of the eyes, the tip of the nose, corners of the mouth etc. Dlib's facial landmark detector provides us with many points to choose from as shown in the fig.4.4. will use the tip of the nose, the chin, the left corner of the left eye, the right corner of the right eye, the left corner of the mouth, and the right corner of the mouth.

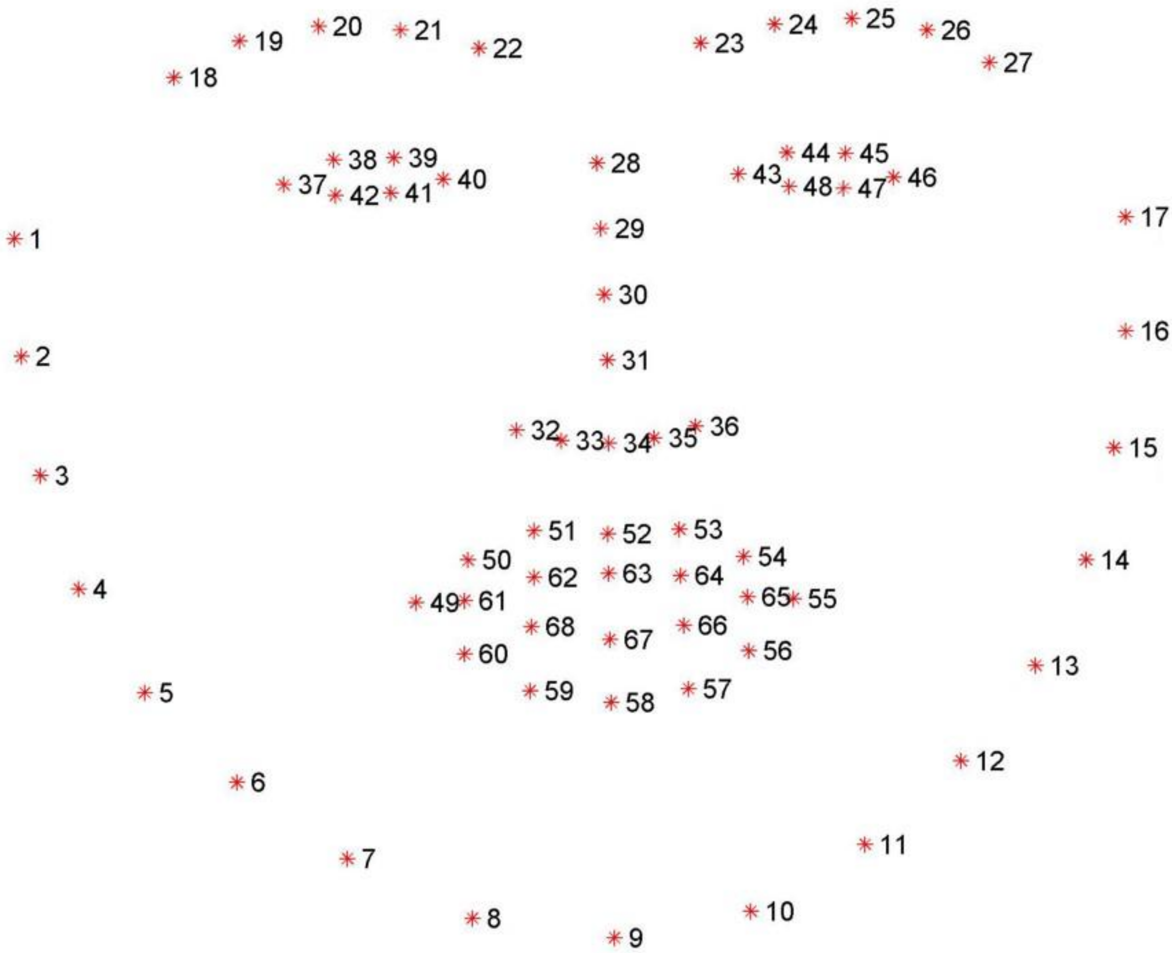

Fig 4.4 Dlib facial keypoints

3D locations of the same points :

We need the 3D location of the 2D feature points. 3D locations of a few points in some arbitrary reference frame.

1. Tip of the nose : (0.0, 0.0, 0.0)
2. Chin : (0.0, -330.0, -65.0)
3. Left corner of the left eye : (-225.0, 170.0, -135.0)
4. Right corner of the right eye : (225.0, 170.0, -135.0)
5. Left corner of the mouth : (-150.0, -150.0, -125.0)
6. Right corner of the mouth : (150.0, -150.0, -125.0)

The above points are in some arbitrary reference frame / coordinate system. This is called the **World Coordinates** .

Intrinsic parameters of the camera.

The camera is assumed to be calibrated. the focal length of the camera, the optical center in the image and the radial distortion parameters are needed to calibrate the camera. Approximate the optical center by the center of the image, approximate the focal length by the width of the image in pixels and assume that radial distortion does not exist.

The 3D coordinates of the various facial features are in **world coordinates**. Transform the 3D points in world coordinates to 3D points in **camera coordinates**. The 3D points in camera coordinates can be projected onto the image plane using the intrinsic parameters of the camera.

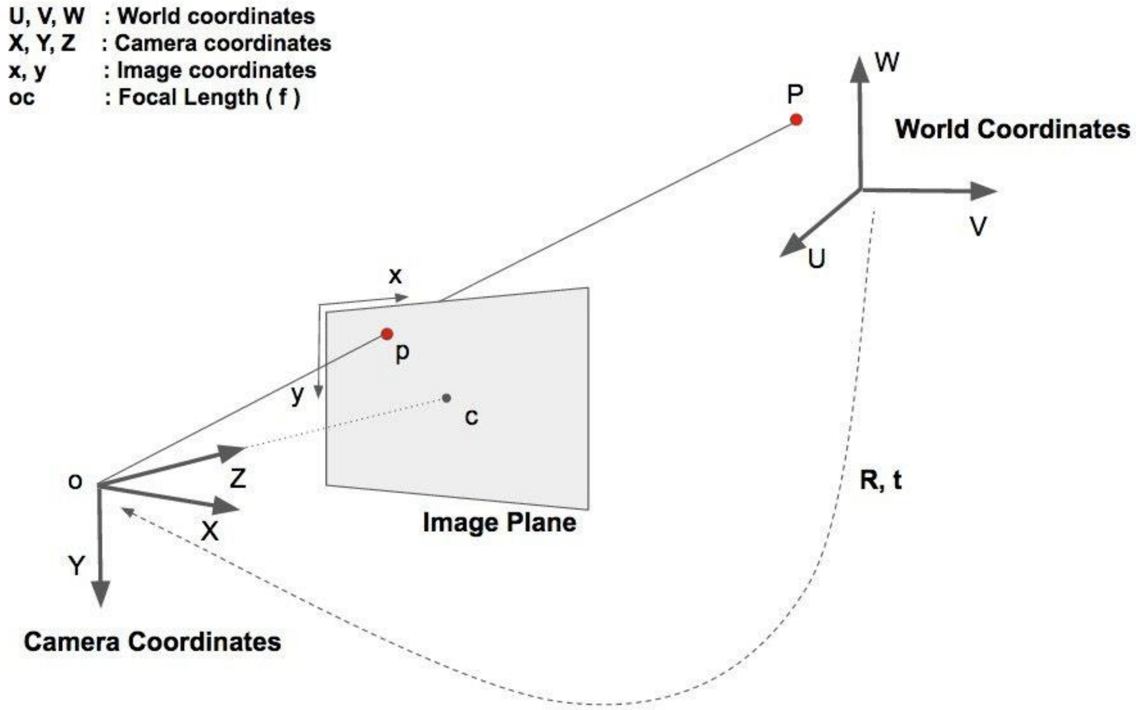

Fig 4.5 Transformation of 3d points

4.3.2 Eye Feature Extraction

A region in an image is a group of connected pixels with similar properties. Regions are important for the interpretation of an image because they correspond to areas of interest in a scene . So considering an eye image, the eye area can be partitioned into two simple overlapping regions which may be accepted to be the features of a human eye.

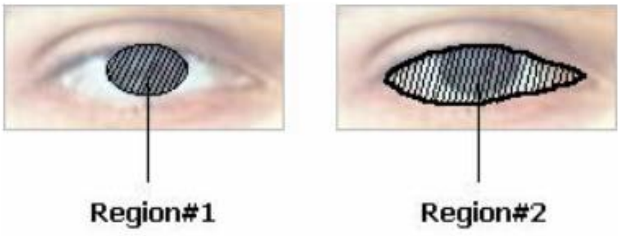

Fig 4.6 Human eye regions

Region#1 : The area occupied by iris and pupil parts of the eye.
Region#2 : The visible eyeball area. Region#2 contains region#1.
There are two common approaches for partitioning an image into regions;

1) Region-based segmentation.
2) Boundary estimation using edge detection.

The second approach which is simply finding the pixels that lie on a region boundary is used for indicating region#1 and region#2 in images.

Region#1 Detection

The region formed by eye pupil and iris can be obtained by an approach based on edge detection. This area has a significant change in the image intensity when compared to the surrounding area formed by the whites of the eye and skin. Due to the noise and other factors such as ambient lightening, some preprocessing steps shall be applied to the eye area before edge detection.

1) Color Space Conversion

In order to facilitate the detection of edges, it is essential to determine changes in intensity in the neighborhood of a point. This is achieved by converting color eye images into gray-scale images. Using different components of different color spaces, such as Z component of XYZ color space, instead of RGB to gray-scale conversion results in better edge strengthened images.

2) Thresholding

There is a strong edge content in the area between region#1 and its outside area. However there are still many points with lower intensity values than region#1 and which will be marked as edges during edge detection. Therefore the color space converted eye area image is binary thresholded to obtain a binary image eliminating edges formed by lower intensity areas.

3) Edge Detection

It is expected that the steps up to now will produce a segmented image having only two regions which are object of interest and remaining area. However, due to several factors there are some regions marked as region#1.

4) Circle Detection

Finding the edge points the edge image that best fits to form a circle. So the whole edge image is searched using circle detection algorithm and the most voted circle is accepted as the boundary of region#1.

Circle Detection Algorithm

Implemented circle detection algorithm is based on a voting method in the edge image. The most voted pixel is accepted as the center of the circle .

1) Circle detection algorithm is based on a voting mechanism where edge pixels vote for the center pixel. Main idea of circle detection algorithm is Edges, white pixels in the edge image are the boundaries of the circle and the pixel which is surrounded by most of the edge pixels with equal distances is the center of the circle.

2) Using the whole image as the search area for center coordinates and edges is a time consuming process, so limiting the search areas by some rules which are found experimentally reduces the time to calculate the center significantly.

a) Center is chosen from the eye area given below;

$$Minimum\ X\ for\ Center = \frac{Width\ of\ eye\ area}{4}$$

$$Maximum\ X\ for\ Center = \frac{Width\ of\ eye\ area \times 3}{4}$$

$$Minimum\ Y\ for\ Center = \frac{Height\ of\ eye\ area}{4}$$

$$Maximum\ Y\ for\ Center = \frac{Height\ of\ eye\ area \times 3}{4}$$

b) Edges voting for the center are chosen from the eye area given below;

$$Minimum\ X\ for\ Edges = \frac{Width\ of\ eye\ area}{6}$$

$$Maximum\ X\ for\ Edges = \frac{Width\ of\ eye\ area \times 5}{6}$$

$$Minimum\ Y\ for\ Edges = \frac{Height\ of\ eye\ area}{6}$$

$$Maximum\ Y\ for\ Edges = \frac{Height\ of\ eye\ area \times 5}{6}$$

3)During the implementation phase we sometimes have observed that edge points which do not belong to the boundary of region#1 may form a perfect circle. So in order to avoid this, checking the radius of the best fitted circle with the rule is necessary.

$$Radius < \frac{\min(image\_width, image\_height)}{3}$$

As a result of circle detection algorithm, radius and coordinates center of the perfect circle is also provided. The location of the center compared to the width of the eye area may be accepted as an indication of the direction of gaze. The width and coordinates of the perfect ellipse fitted onto the visible eyeball area i.e. region#2 with the information of circle fitted onto the region#1 gives more accurate results but we have used the width information of the eye area which is supplied by adaptive eigen eye method and coordinates of the circle fitted onto the region#1 instead.

Region#2 Detection

Region#2 contains region#1. Whole eye area may be accepted as visible eyeball area surrounded by skin so somehow eliminating the skin region will output the eyeball area seen by the imaging system.

Steps for detecting boundary of region#2;

1) Color Space Conversion

Color space is mandatory when dealing with images for thresholding, external energy minimization of a snake, region partitioning etc. but also the type of color space conversion and component used play a very important role. Eye area during detection of region#2 must be seen as eyeball area surrounded by skin. So using the color information of human skin, skin region may be eliminated as a first preprocessing step. To obtain the color information of the whole area we have used the HUE value and obtained quite satifying results before thresholding even if the images were noisy and taken under bad lightening conditions

2) Thresholding

Thresholding is used for converting gray-scale image obtained as a result of the HUE component of the HSV color space into a binary image. By thresholding, objects of interest, here reigon#2, are separated from the background. Binary thresholding with a fixed value may lead to good results in most cases, Adaptive thresholding is a better type of thresholding as strong illumination gradient may occur in the eye area during the movement of the head.

4.4 Mobile Phone Detection and Person counting

[7] You Only Look Once (YOLOv3) based on the Custom dataset for detecting and recognizing the objects in the indoor environment such as offices or rooms. OpenCV library is used for capturing and processing the images. YOLOv3 detects and recognizes the objects in each image.

The following steps are employed for detecting and recognizing the objects :

Step 1: OpenCV library was used to operate the webcam for the purpose of capturing the frames (images).

Step 2: OpenCV was used to resize each frame (image) to 416x416.

Step 3: If there are objects in the frame, YOLOv3 will detect and recognize these objects depending on the weight file. However, if the frame has no objects, then the next frame is selected. The weight file is generated from the training process.

Step 4: When YOLOv3 detect and recognize the objects, their locations are calculated depending on the width of the frame and the center point of these objects.

Object Detection With YOLOv3

The keras-yolov3 provides a lot of capability for using YOLOv3 models, including object detection, transfer learning, and training new models from scratch. These were trained using the DarkNet code base on the MSCOCO dataset. The model architecture is called a "*DarkNet*" and was originally loosely based on the VGG-16 model. we need to load the model weights. The model weights are stored in whatever format that was used by DarkNet.

Working of YOLOv3

YOLO is a Convolutional Neural Network (CNN) for performing object detection in real-time. CNNs are classifier-based systems that can process input images as structured arrays of data and identify patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still maintains accuracy. It allows the model to look at the whole image at test time, so its predictions are informed by the global context in the image. YOLO and other convolutional neural network algorithms “score” regions based on their similarities to predefined classes. High-scoring regions are noted as positive detections of whatever class they most closely identify with.

Precision for Small Objects

Higher the average precision, the more accurate it is for that variable.

Specificity of Classes

The new YOLOv3 uses independent logistic classifiers and binary cross-entropy loss for the class predictions during training. YOLO v3 uses a multilabel approach which allows classes to be more specific and be multiple for individual bounding boxes. YOLOv3 performs real-time detections.

Model Weights

YOLO’s COCO pretrained weights by initializing the model with **model = YOLOv3().** Using COCO’s pre-trained weights means that you can only use YOLO for object detection with any of the 80 pretrained classes that come with the COCO dataset.

Making a Prediction

The convolutional layers included in the YOLOv3 architecture produce a detection prediction after passing the features learned onto a classifier or regressor. These features include the class label, coordinates of the bounding boxes, sizes of the bounding boxes, and more. Since the prediction with YOLO uses 1 x 1 convolutions (hence the name, “you only look once”), the size of the prediction map is exactly the size of the feature map before it. In YOLOv3 , the way this prediction map is interpreted is that each cell predicts a fixed number of bounding boxes. Then, whichever cell contains the center of the ground truth box of an object of interest is designated as the cell that will be finally responsible for predicting the object. There is a ton of mathematics behind the inner workings of the prediction architecture.

1) Anchor Boxes

Object detectors using YOLOv3 usually predict log-space transforms, which are offsets to predefined “default” bounding boxes. Those specific bounding boxes are called anchors. The transforms are later applied to the anchor boxes to receive a prediction. YOLOv3 in particular has three anchors. This results in the prediction of three bounding boxes per cell (the cell is also called a neuron in more technical terms).

2) Non-Maximum Suppression

Objects can sometimes be detected multiple times when more than one bounding box detects the object as a positive class detection. Non-maximum suppression helps avoid this situation and only passes detections if they haven’t already been detected. Using the NMS threshold value and confidence threshold value, NMS is implemented to prevent double detections.

Interpreting Results

Interpreting the results of a YOLO model. Multiple factors go into a successful interpretation and accuracy rating, such as the box confidence score and class confidence score used when creating a YOLOv3 computer vision model.

Class Confidence and Box Confidence Scores

Each bounding box has an x, y, w, h, and box confidence score value. The confidence score is the value of how probable a class is contained by that box, as well as accuracy of bounding box . The bounding box width and height (w and h) is first set to the width and height of the image given. Then, x and y are offsets of the cell and all 4 bounding box values are between 0 and 1. Then, each cell has 20 conditional class probabilities implemented by the YOLOv3 algorithm. The class confidence score for each final boundary box used as a positive prediction is equal to the box confidence score multiplied by the conditional class probability. The conditional class probability is the probability that the detected object is part of a certain. YOLOv3’s prediction has 3 values of h, w, and depth.The boundary boxes with high confidence scores (more than 0.25) are kept as final predictions.

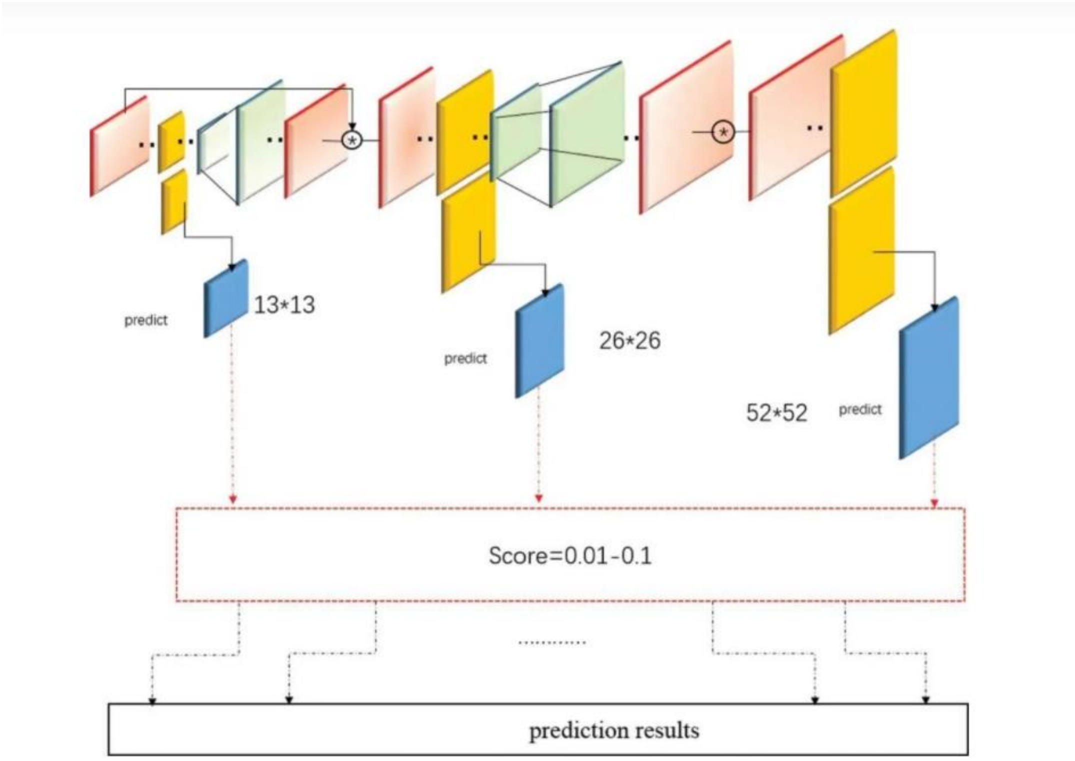

Fig 4.7 YOLOv3 prediction results

If there is no person detected in the frame then it shows that No person detected in the command prompt. If the person count exist more than one in the frame it will shows that More than one person detected .

During Exam if the student getting any help from external devices like mobile phone, the appearance of the mobile phone will be detected with bounding box and it will show the information of mobile phone detected in command prompt .

CHAPTER 5

OUTPUT

5.1 Screenshots

The face is detected using MTCNN. The output image with bounding box on face and 5 facial keypoints as shown in the fig.5.1

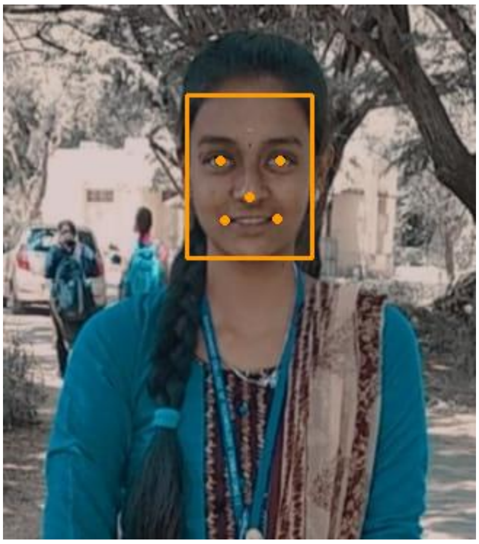

Fig 5.1 Face detection using MTCNN

The face verification is done by providing two images which are extracted and checking the similarity, if the faces are similar ,output as face matched as shown in fig.5.2

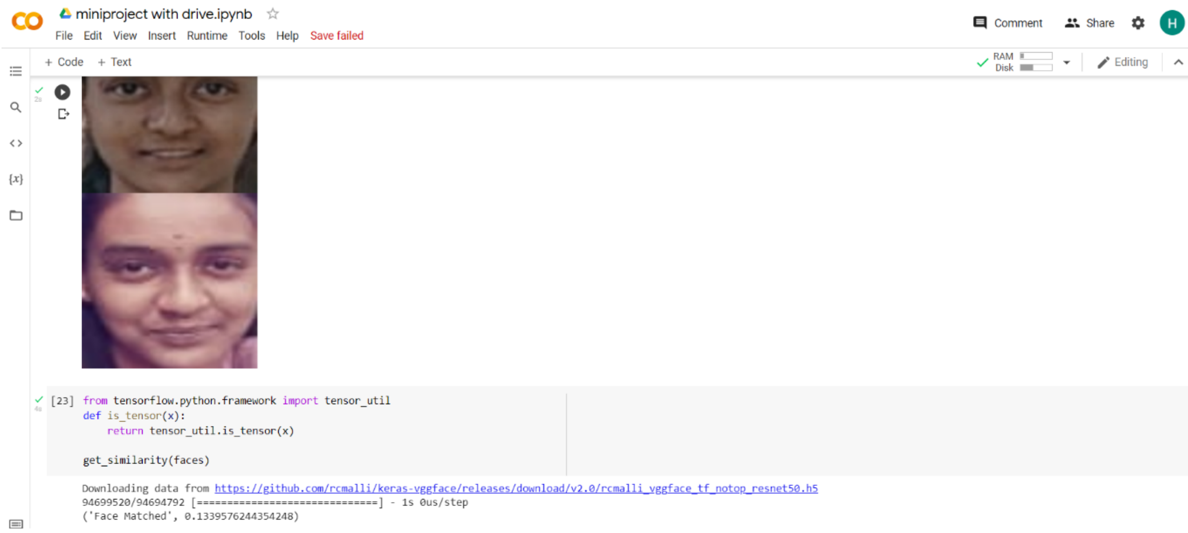

Fig 5.2 Similar faces Matched

Providing two different images which are extracted and checking the similarity whether the faces are not similar ,output as face not matched as shown in fig.5.3

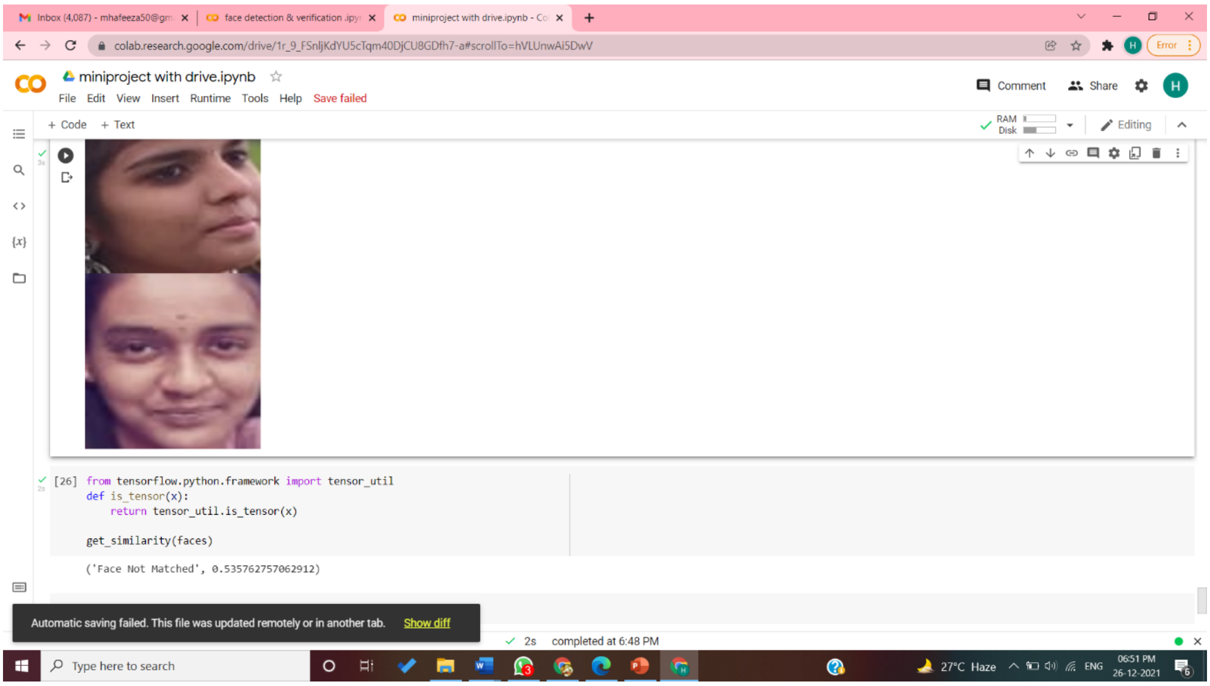

Fig 5.3 Dissimilar faces Not Matched

During the monitoring process, the webcam is enabled. If more than one person is detected in frame, then it will notify in command prompt as shown in fig.5.4

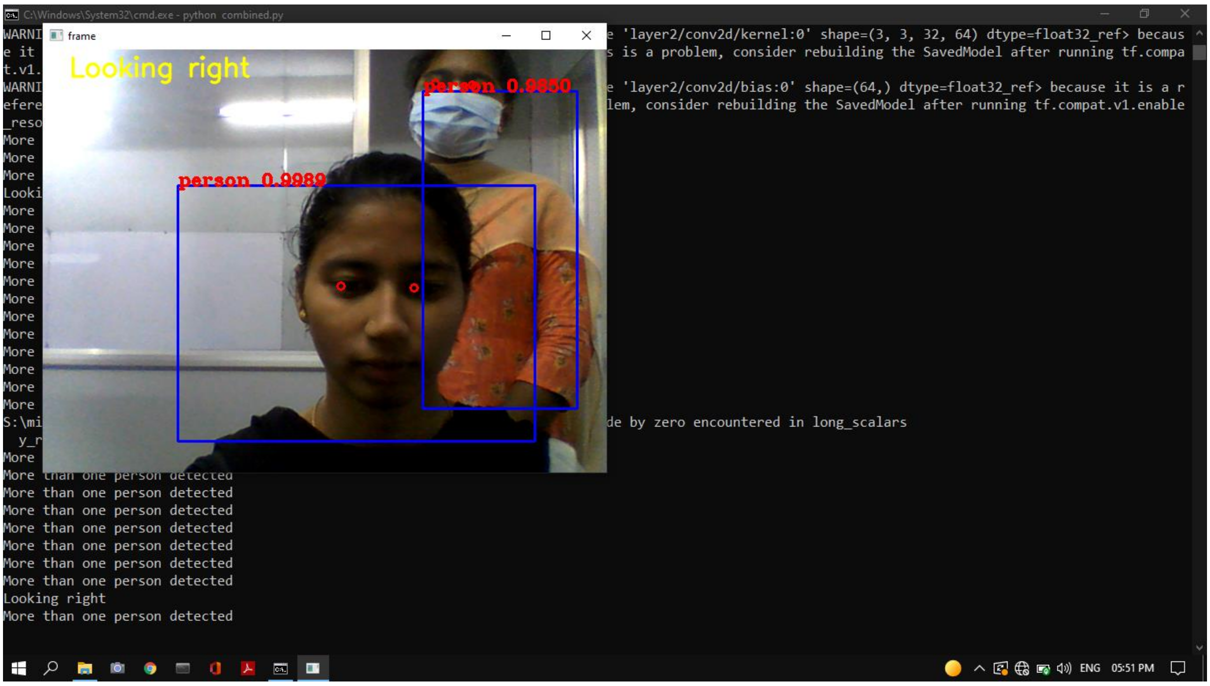

Fig 5.4 More than one person detected

During the monitoring process, if no person is detected in frame, then it will notify that no person is detected in command prompt as shown in fig.5.5

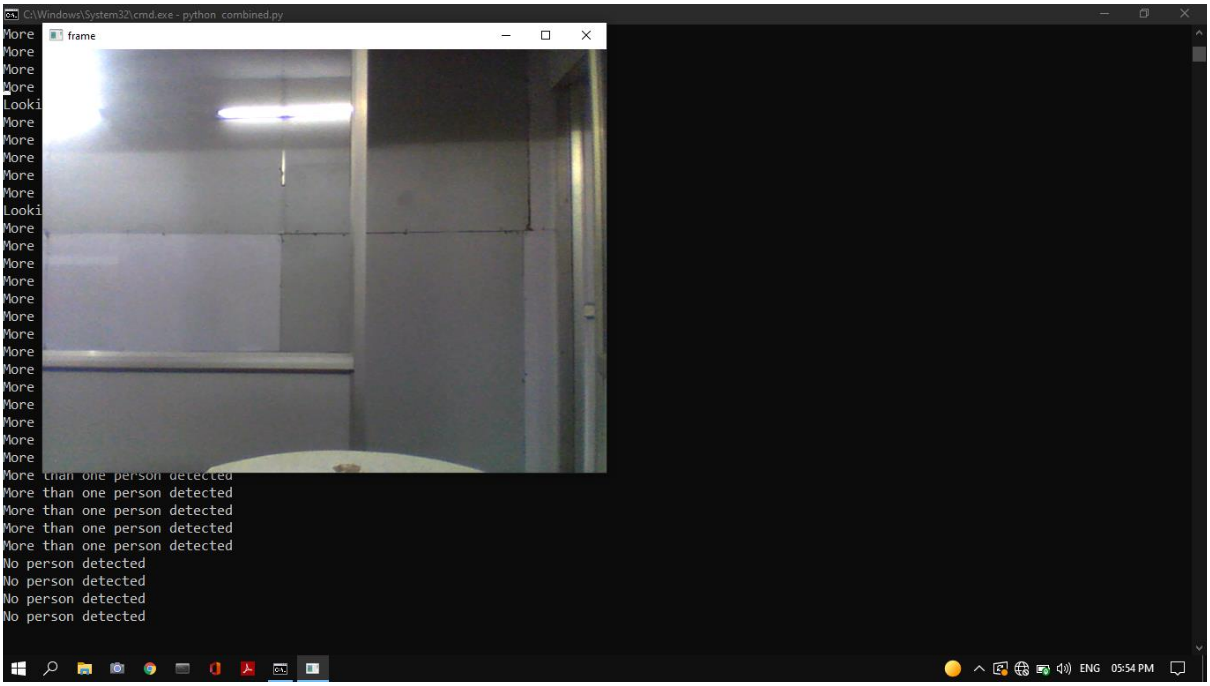

Fig 5.5 No person detected

If mobile phone is detected in the frame, then it will notify that mobile phone detected in command prompt as shown in fig.5.6

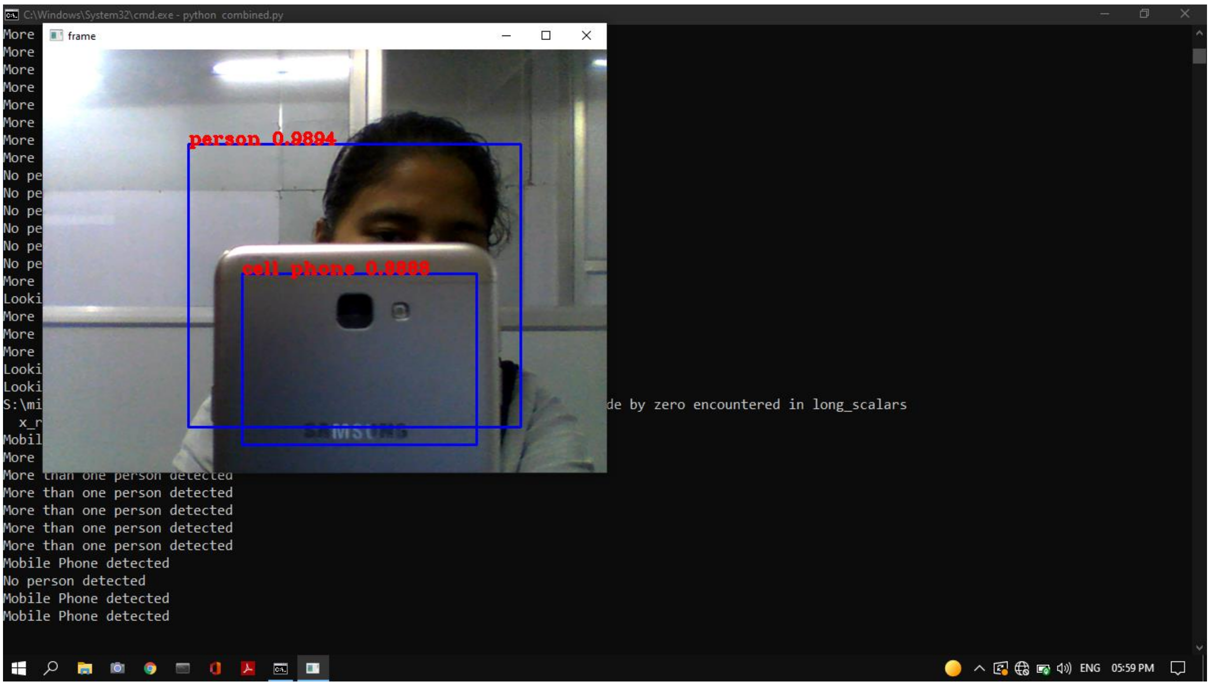

Fig 5.6 Mobile Phone detected

In fig.5.7 shows the binary masking of eye tracking used to monitor eye movement with adjustable thresholding value useful for clear visualization of eyeball movement.

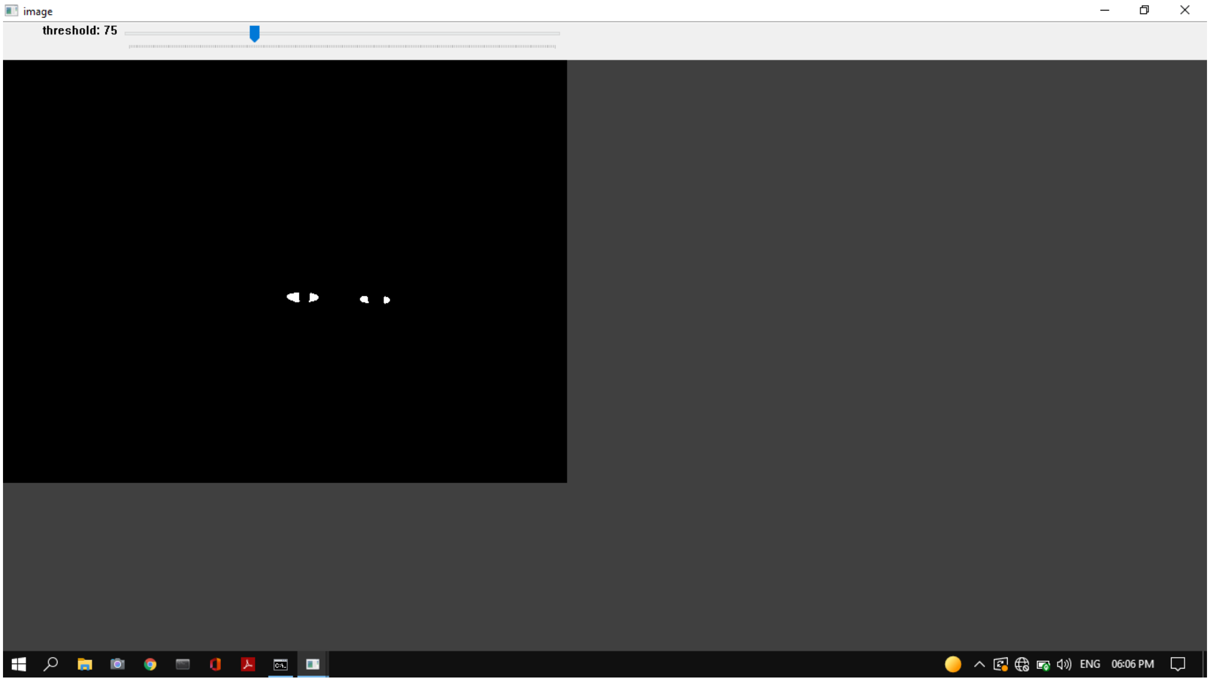

Fig 5.7 Binary masking for Eye Tracking

During eye tracking process, student is more deviated to the right side of the frame, then it will indicate that looking right in screen as well as in command prompt as shows in fig.5.8

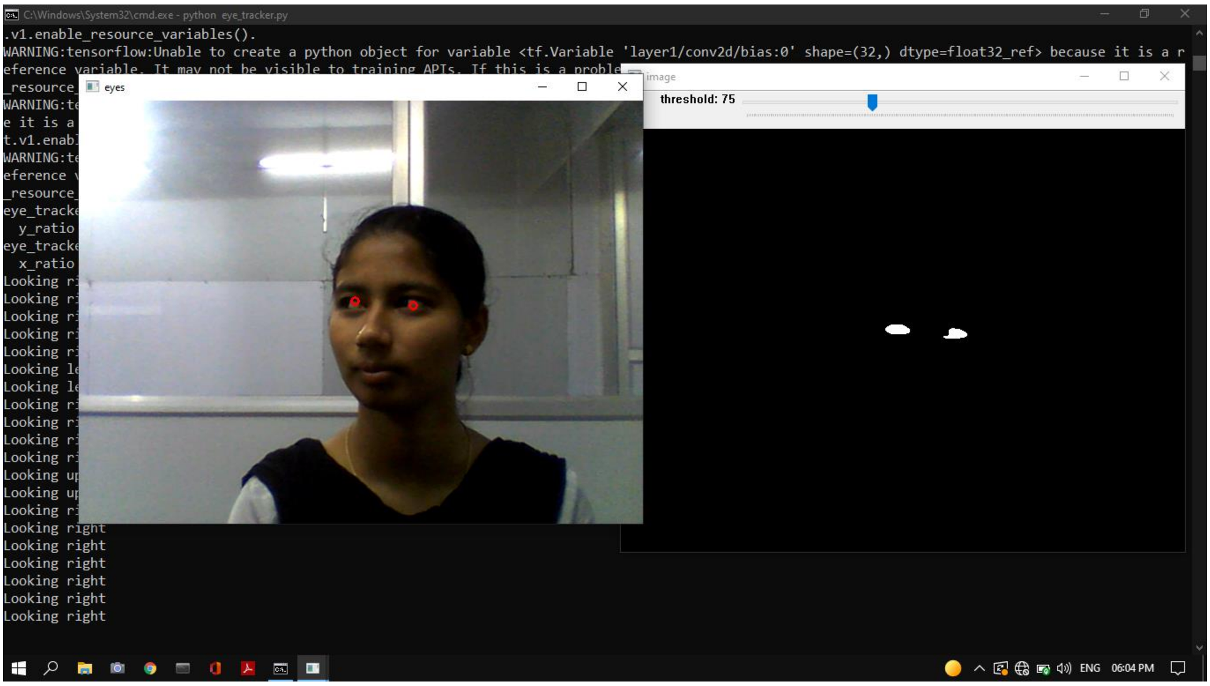

Fig 5.8 Looking right Noted

During eye tracking process, student is more deviated to the left side of the frame, then it will indicate that looking left in screen as well as in command prompt as shows in fig.5.9

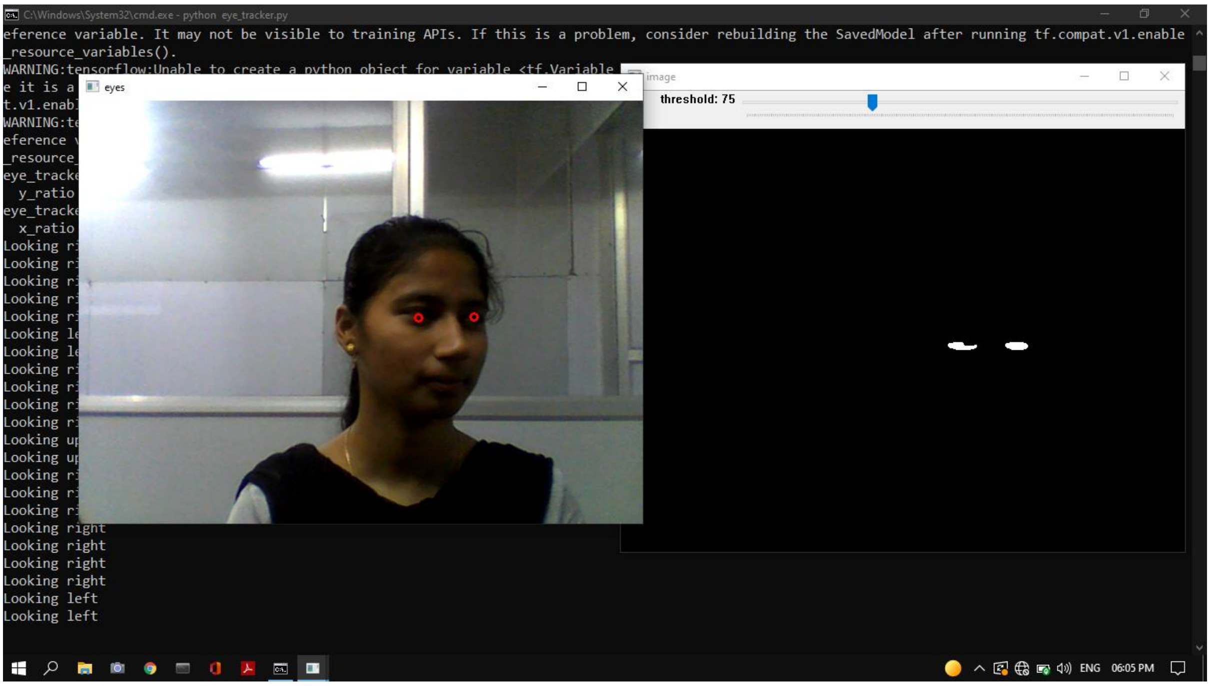

Fig 5.9 Looking Left Noted

5.2 Conclusion

Online proctoring can transform the education sector and has made everything possible virtually. The system can ensure the authenticity of the test by preventing the candidate from cheating and indulging in unfair means during the assessment. With Online Proctoring, educational institutes don’t need to delay or postpone examinations amid the COVID-19 outbreak.

The automated proctoring system where the user can be monitored automatically through the webcam. Before Starting the Examination the student face is detected and Verified. Monitoring the student during online examination includes Eyeballs tracking, Mobile Phone Detection, Count the number of person in the frame.

5.3 Future Work

As a part of future work, we are planning on integrating this model into a real time dynamic website, which would further improve its functionality and application. Along with the features eye tracking, mobile phone detection and person counting, the system is more improved with face spoofing and audio tracking.

CHAPTER 6

REFERENCE

[1] Zhang, Kaipeng & Zhang, Zhanpeng & Qiao, Yu. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters. 23. 10.1109/LSP.2016.2603342.

[2] Cao, Qiong & Shen, Li & Xie, Weidi & Parkhi, Omkar & Zisserman, Andrew. (2018). VGGFace2: A Dataset for Recognising Faces across Pose and Age. 67-74. 10.1109/FG.2018.00020.

[3] A. H. S. Ganidisastra and Y. Bandung, "An Incremental Training on Deep Learning Face Recognition for M-Learning Online Exam Proctoring," 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), 2021, pp. 213-219, doi: 10.1109/APWiMob51111.2021.9435232.

[4] Yadav, Jyotsna & Rajpal, Navin & Vishwakarma, Pradeep. (2016). Face recognition using Symlet, PCA and cosine angle distance measure. 1-7. 10.1109/IC3.2016.7880231

[5] Mehrubeoglu, Mehrube & Pham, Linh & Le, Hung & Muddu, Ramchander & Ryu, Dongseok. (2011). Real-time eye tracking using a smart camera. Proceedings - Applied Imagery Pattern Recognition Workshop. 1-7. 10.1109/AIPR.2011.6176373.

[6] Hassan, Mohamed. (2017). Research design in eye-tracking experiments. 10.13140/RG.2.2.16417.81765.

[7] P. Adarsh, P. Rathi and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 687-694, doi: 10.1109/ICACCS48705.2020.9074315.

[8] L. Cuimei, Q. Zhiliang, J. Nan and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), 2017, pp. 483-487, doi: 10.1109/ICEMI.2017.8265863.

[9] S. Kadambari, G. Prabhu, D. Mistry and M. Khanore, "Automation of Attendance System Using Facial Recognition," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 2019, pp. 1-5, doi: 10.1109/ICAC347590.2019.9036819.

[10] V. Ghenescu, R. E. Mihaescu, S. Carata, M. T. Ghenescu, E. Barnoviciu and M. Chindea, "Face Detection and Recognition Based on General Purpose DNN Object Detector," 2018 International Symposium on Electronics and Telecommunications (ISETC), 2018, pp. 1-4, doi: 10.1109/ISETC.2018.8583861.

[11] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015, doi: 10.1109/TPAMI.2015.2389824.

[12] K. Yan, S. Huang, Y. Song, W. Liu and N. Fan, "Face recognition based on convolution neural network," 2017 36th Chinese Control Conference (CCC), 2017, pp. 4077-4081, doi: 10.23919/ChiCC.2017.8027997.