WEB-BASED TRACKING SYSTEM ENHANCED BY ARTIFICIAL INTELLIGENCE

[1] PRANEETHREDDY CHITTY,[2] Md. SABDAR ASHMI,[3] RAMYA VADLURI,[4] AZRA FARHEEN,
[5] GARREPALLY VINIL KUMAR
[123]Assistant Professor,[45]Students
Department of CSM
Vaagdevi College of Engineering, Warangal, Telangana

ABSTRACT:

This paper presents the development of an intelligent web-based tracking system that leverages artificial intelligence (AI) to enhance monitoring and management across various domains, such as logistics, healthcare, and personal asset tracking. Traditional tracking systems often rely on basic GPS and manual input, leading to inefficiencies and data inaccuracies. Our proposed system integrates advanced AI algorithms, including machine learning and predictive analytics, to provide real-time tracking, automated data processing, and insightful analytics. Users can access the system via a web interface, enabling seamless interaction and visualization of tracking data. Preliminary results demonstrate the system's capability to improve tracking accuracy and operational efficiency, highlighting its potential applications in various sectors. This research contributes to the growing field of AI-driven solutions, emphasizing the importance of intelligent systems in modern tracking and management.

1. INTRODUCTION

In an increasingly interconnected world, effective tracking and monitoring have become vital across numerous industries, including logistics, transportation, healthcare, and personal security. Traditional tracking methods, often reliant on manual processes and basic GPS technology, face challenges such as data inaccuracy, inefficiency, and a lack of real-time insights. As a result, there is a pressing need for more intelligent and automated systems that can enhance tracking capabilities and optimize operations.

This study proposes the development of an intelligent web-based tracking system powered by artificial intelligence. By incorporating advanced AI techniques such as machine learning, natural language processing, and predictive analytics, the system aims to provide users with accurate, real-time tracking information and actionable insights. The web-based interface allows users to access tracking data from any device, facilitating ease of use and enhancing user experience.

The objective of this research is to design a robust tracking system that not only improves data accuracy but also automates data processing and analysis, ultimately leading to better decision-making and resource management.

2. LITERATURE SURVEY

The application of artificial intelligence in tracking systems has gained considerable attention in recent years. Research has shown that integrating AI techniques can significantly enhance the performance and functionality of tracking solutions. For instance, machine learning algorithms have been successfully used to analyze historical tracking data, enabling predictive modeling that improves operational efficiency (Zhao et al., 2021). Additionally, AI-driven systems can automatically identify anomalies and patterns, providing real-time alerts that assist in proactive decision-making (Patel et al., 2020).

Several studies have demonstrated the effectiveness of AI in specific domains, such as logistics and supply chain management. Systems that incorporate AI can optimize route planning, manage inventory more efficiently, and improve delivery times by analyzing vast amounts of data and adapting to changing conditions (Huang et al., 2022). Furthermore, AI has been applied in healthcare tracking systems to monitor patient data and optimize resource allocation, illustrating its versatility and potential impact (Nguyen et al., 2021).

Despite these advancements, challenges remain, including data privacy concerns, integration complexities, and the need for user-friendly interfaces. Ensuring that AI-driven tracking systems are transparent and secure is critical for user acceptance and compliance with regulatory standards (Lee et al., 2021).

In summary, the literature indicates a growing trend toward the integration of AI in tracking systems, with promising outcomes in various sectors. By leveraging the capabilities of AI, these systems can provide enhanced accuracy, efficiency, and decision-making support, paving the way for more intelligent tracking solutions.

III.SYSTEM ANALYSIS

SYSTEM ARCHITECTURE:

EXISTING SYSTEM

- Physical proctoring of the exam involves invigilator being physically present at the examination center.
- For each batch of 50 students one person is needed to monitor the activities of the exam. This is traditional approach followed by many entities while managing examination process.

DISADVANTAGES

- Offline exam required large administrative and operational setup.
- Arrangement of question papers and answer sheet takes heavy cost.

- Chances of cheating or use of unfair means.
- To take offline exam of more candidates more invigilators are required.

PROPOSED SYSTEM

Online exam activity can reduce cost of physical exam up to certain extent. Online Examination Process can eliminate logistical activities related to paper-based exam process significantly.

Throughout the exam, we employed a webcam to obtain video input that was used to identify several factors such as the examinee's emotions, head pose estimate, multiple person detection, cell phone and book detection. When all of these features are integrated, they are utilized to detect harmful behaviors that may occur during the exam. As a result, the system makes a decision on the users' actions. Fig.1 depicts the block diagram of the mentioned system.

The remainder of this section elaborates the following topics:

1) Face Detection

2) Person and Phone Detection

3) Facial Landmarks Detection

a) Eye-ball Tracking

b) Mouth Movement Tracking

c) Head Pose Estimation

FUNCTIONALITIES

It includes four vision-based features as well as speech-to-text conversion:

1. Keep a watch on the candidate's eyes and note whether they're gazing left, right, or up.
2. By measuring the space between the candidate's lips at the start, you can see if he expands his mouth.
3. Find and report any mobile phones or other objects.
4. To figure out where the person is looking, estimate their head posture.
5. Create a text file using the user's words.

1. Face Detection

Face detection is a difficult computer vision task that involves detecting and locating people in images. Faces were previously found using Dlib's frontal face HOG detector. However, it did not produce satisfactory outcomes. Face identification models like Haar, dlib, Multi-task Cascaded Convolutional Neural Network (MTCNN), and OpenCV's DNN module were compared. The DNN module in OpenCV gives the best result. Eye tracking, mouth open close detection, and head position estimation all rely on face detections.

For the face detector, an extra quantized model has been added, which may be used by setting the option quantized to True when calling the get face_detector method(). On our machine, the conventional version of the face detector delivered 17.5 frames per second, while the quantized version gave 19.5 frames per second. Because it is uint8 quantized, this would be extremely handy when distributing on edge devices.

2. Person and Phone Detection

To recognize people and mobile phones in the webcam feed, we used the pre-learned weights of YOLOv3 trained on the COCO dataset. Using pre-trained YOLOv3, We counted persons in a webcam. An alarm can be triggered if the count is not equal to one. The COCO dataset's mobile phone index is 67, thus we need to see whether any class indexes are equivalent to that, and then we can report a mobile phone. We can establish a single thread for eye-tracking and mouth detection because they are both based on dlib, and another thread can be used for the YOLOv3 tasks of people counting and mobility detection. YOLOv3 is applied to the webcam stream in the count _people_and_phone() function. The classifications of objects observed are then examined, and if more than one person or a mobile phone is discovered, suitable action is performed.

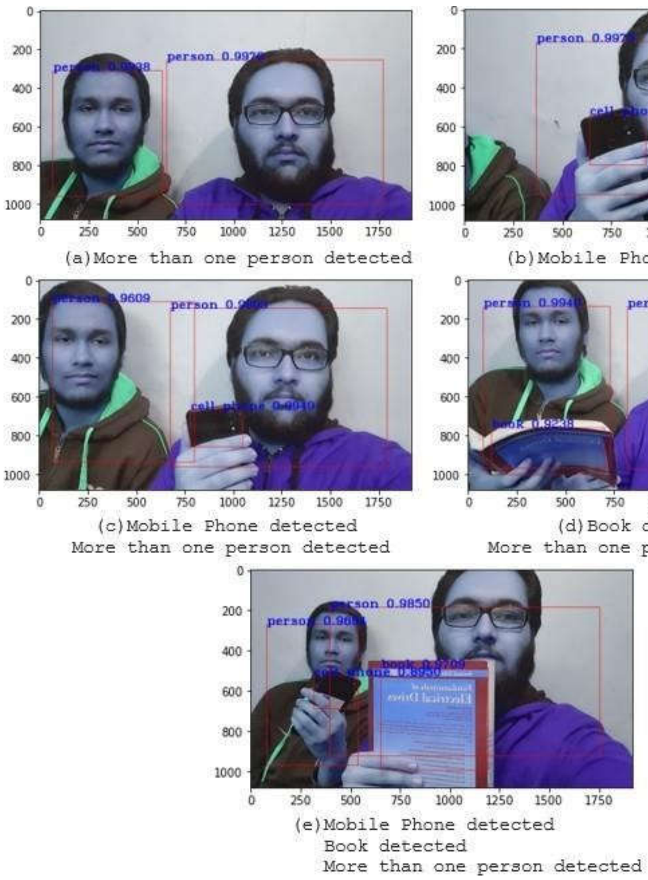

Fig 2.4.1 Output for Cell Phone, Book and Multiple person detection

3. Facial Landmarks Detection

The task of detecting and tracking significant facial landmarks is known as facial landmark detection. Previously, Dlib's model was used, however it does not work well when the face is at an angle. As a result, the proposed model for landmark detection is based on a convolutional neural network developed with tensorflow. It is used for tracking eyes, mouth opening detection, and head pose estimation. Caffe model of OpenCV's DNN module was used for the purpose of face detection. For facial landmark detection, we used a dlib's pre- trained model [7]. It gives 68 facial landmarks.

Six facial Landmark points were used which are: nose tip, chin, extreme left and right points of lips, and the left corner of the left eye and right corner of the right eye. The standard 3D coordinates of these facial landmarks were used and the rotational and translational vectors at the nose tip were estimated. After obtaining the required vectors, those 3D points were protected on a 2D surface that is our image.

a. Eye-ball Tracking

We'll need a face key points detector that can detect eyes in real time for eye tracking. For this, we'll use the dlib library's pre-trained network, which can recognize '68 key spots.' Dlib is utilized because, unlike a Convolutional Neural Network model, it can make predictions in real time.

Before we can continue on to image processing, we must first locate eyes. To find the eyes, we must first locate a face. The facial keypoint detector takes a rectangular object from the dlib module as input, which is just a set of face coordinates. The integrated frontal face detector of dlib is used to find faces.

b. Mouth Movement Tracking

This is used to see if the candidate speaks during the examination. This is analogous to the detection of eyes. This task makes use of Dlib's facial keypoints once more. For 100 frames, the distance between the lips keypoints (5 outer pairs and 3 inner pairs) is recorded. To achieve the final result, this is then averaged. The distance between the points increases as the user opens his or her mouth, and if the increase in distance is greater than a specific amount for at least three other pairs and two inner pairs, the user is flagged.

c. Head Pose Estimation

To determine where the head is facing, head pose estimation is performed. Because of the several steps required to solve it, head position estimation is a challenging problem in computer vision.

We must first locate the face in the frame, followed by the numerous facial markers. In today's world, recognizing a person's face appears to be a simple process, and this is especially true when the person is facing the camera. When the face is at an angle, the difficulty occurs. Add to that the fact that some facial landmarks are obscured by head movement. Then, in order to get the inclination, we must convert the points to 3D coordinates. Six points of the face namely nose tip, chin, extreme left and right points of lips, the left corner of the left eye, and the right corner of the right eye are required.

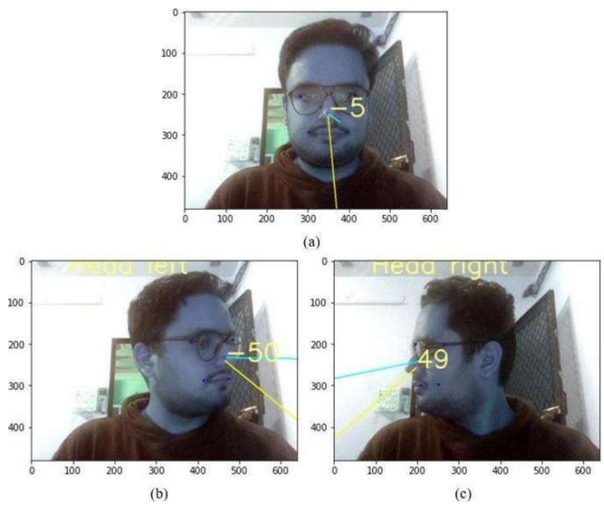

Fig 2.4.2 Output for Head Pose Estimation Model:
(a) Examinee in frontal stable state,
(b) Examinee viewing left,
(c) Examinee viewing right

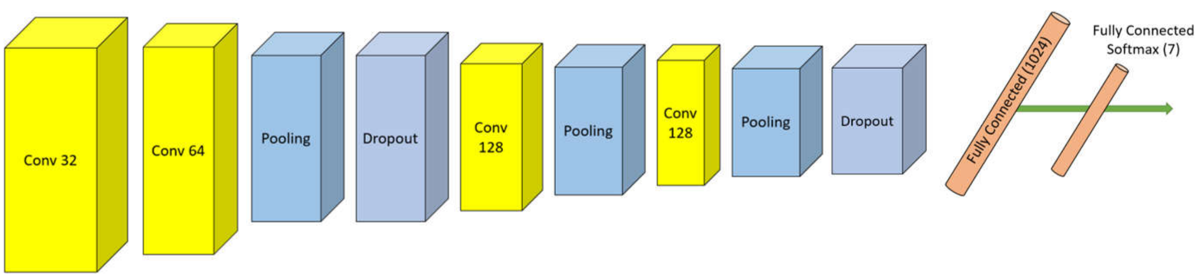

Fig 2.4.3 Architecture of the proposed system

ADVANTAGES

The use of AI-based remote proctoring and invigilation technologies mainly improves the authenticity of the exam conducted in a remote environment. Here are some of the significant advantages that AI-based remote proctoring offers.

- Better Accuracy
- Improved Security
- Better Scalability
- Cost-effective
- Smarter and Intelligent

FEASIBILITY STUDY

Preliminary investigation examines the project feasibility, the likelihood that the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economic feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any additional hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information systems that will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised to test the operational feasibility of a project include the following: -

- Is there sufficient support for the management from the users?

- ➢ Will the system be used and work properly if it is being developed and implemented?
- ➢ Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

TECHNICAL FEASIBILITY

The technical issues usually raised during the feasibility stage of the investigation include the following:

- ➢ Does the necessary technology exist to do what is suggested?
- ➢ Does the proposed equipment have the technical capacity to hold the data required to use the new system?
- ➢ Can the system be upgraded if developed?
- ➢ Are there technical guarantees of accuracy, reliability, ease of access and data security?

The model is generated using OpenCV and Dlib library of Python. Face detection is performed using the Histogram of Oriented Gradients based face detector, and the landmarks calculation are estimated using the 68-point landmark estimation algorithm. For any (most) given related input weights, configuration and names file, it predicts the name of the person. The software and hardware requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology.

IV.IMPLEMENTATION

4.1 MODULES DESCRIPTION

1. Real Time Video Input:

Real time video input is taken from the user with the use of Python's OpenCV library. VideoCapture() function is used take the input from either the integrated camera or web camera.

2. Face Detection:

The real time video input must be preprocessed before moving on to the next step in the algorithm pipeline. Face detection can then be performed on the preprocessed input.

<u>Image pre-processing:</u> The input needs to be pre-processed, i.e., flipped, resized and converted to

greyscale. As the frame received from the web camera is automatically flipped, we need to re-flip it using OpenCV flip() function. Resizing is done using imutils's resize() function. We convert the input to greyscale using OpenCV cvtColor(), as the complexity of dealing with greyscale pixels is lesser than compared to the coloured ones.

Face Detection: Dlib's Histogram of Oriented Gradients (HOG) algorithm based face detector is used for detecting faces in the pre-processed video input. In this algorithm, the gradient direction of an image in localized regions is used to build histograms. HOG is preferred as it is more accurate compared to Haar cascades, and the false positive ratio is smaller. It also allows motion of the user to be overlooked and hence is more practical. A call to the get_frontal_face_detector() function is enough to activate the face detector.

3. Facial Landmark Estimation:

Dlib library's 68 point facial landmark algorithm based predictor is used. This algorithm locates 68 distinct landmarks around the face. We make use of the shape_predictor () function in order to initialize it.

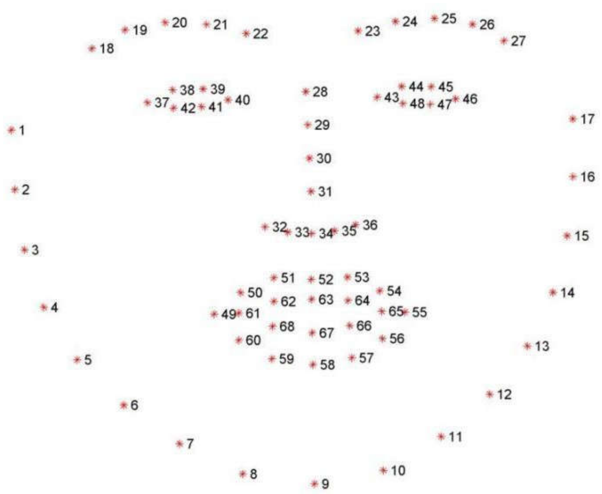

Fig 4.1.1 68 Facial landmarks mapped on the face

The 68 distinct landmarks estimated by dlib's 68 point landmark estimation algorithm are shown in Fig 4.1.1. Out of these, we are interested in the landmarks around the eyes mouth, and the nose.

We Extract the left and right eye coordinates, mouth coordinates and the nose pointer, then draw contours around eyes and mouth using OpenCV's drawContours() function. Using the extracted landmarks, the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) values are calculated.

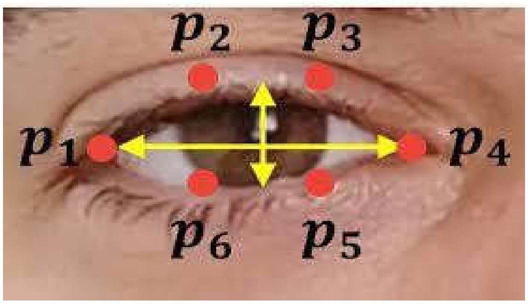

Fig 4.1.2 Landmarks around the eye

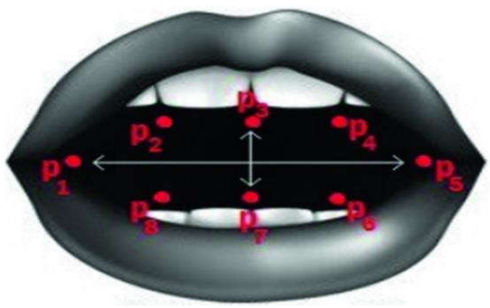

Fig 4.1.3 Landmarks around the mouth

4. Real-time eye tracking:

The first step is to download the required packages. Installation via pip Other than this we will need a facial keypoints detector that can detect eyes in real-time. For this will use a pre-trained network in the dlib library which can detect '68 key points'. Dlib is used because it can give predictions in real-time, unlike a CNN model which was very important for making an AI for online proctoring.

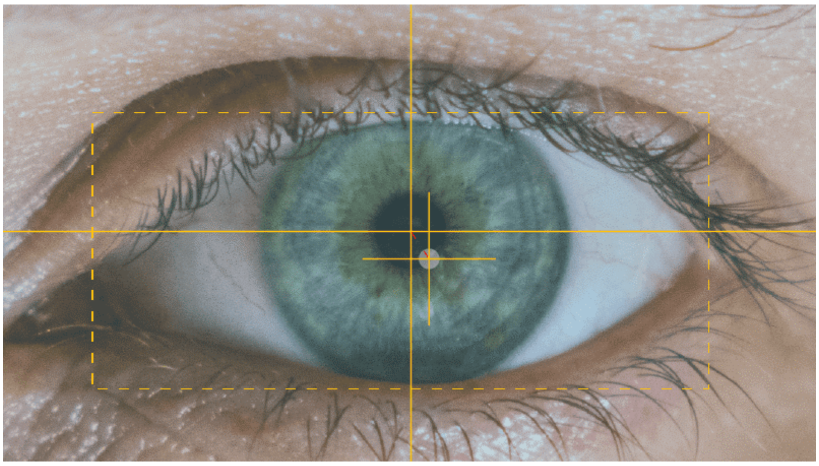

Fig 4.1.4 Eye-ball Tracking

5. Real-time Head Pose Detection:

Head pose estimation is a challenging problem in computer vision because of the various steps required to solve it. Firstly, we need to locate the face in the frame and then the various facial landmarks. Now, recognizing the face seems a trivial task in this day and that is true with faces facing the camera. The problem arises when the face is at an angle. Add to that some facial landmarks are not visible due to the movement of the head. After this, we need to convert the points to 3D coordinates to find the inclination. Sounds like a lot of work? Don't worry we will go step by step and refer two great resources that will make our work a lot easier.

We need six points of the face i.e. is nose tip, chin, extreme left and right points of lips, and the left corner of the left eye and right corner of the right eye. We take standard 3D coordinates of these facial landmarks and try to estimate the rational and translational vectors at the nose tip. Now, for an accurate estimate, we need to intrinsic parameters of the camera like focal length, optical center, and radial distortion parameters. We can estimate the former two and assume the last one is not present to make our work easier. After obtaining the required vectors we can project those 3D points on a 2D surface that is our image.

6. Person and Phone Detection:

YOLOv3 pre-trained model can be used to classify 80 objects and is super-fast and nearly as accurate as SSD. It has 53 convolutional layers with each of them followed by a batch normalization layer and a leaky RELU activation. To downsample, instead of using pooling they have used a stride of 2 in convolutional layers. The input format for it is that the images should be in RGB (so if using

OpenCV remember to convert), having an input type of float32, and can have dimensions 320x320 or 416x416 or 608x608.

Use yolo to get the predictions on the image. Load the class names files containing all the object names for which the model was trained.

Now to count persons or anything present in the classes.txt we need to know its index in it. The index of person is 0 so we need to check if the class predicted is zero then we increment a counter.

V. RESULTS

5.1 OUTPUT SCREENS

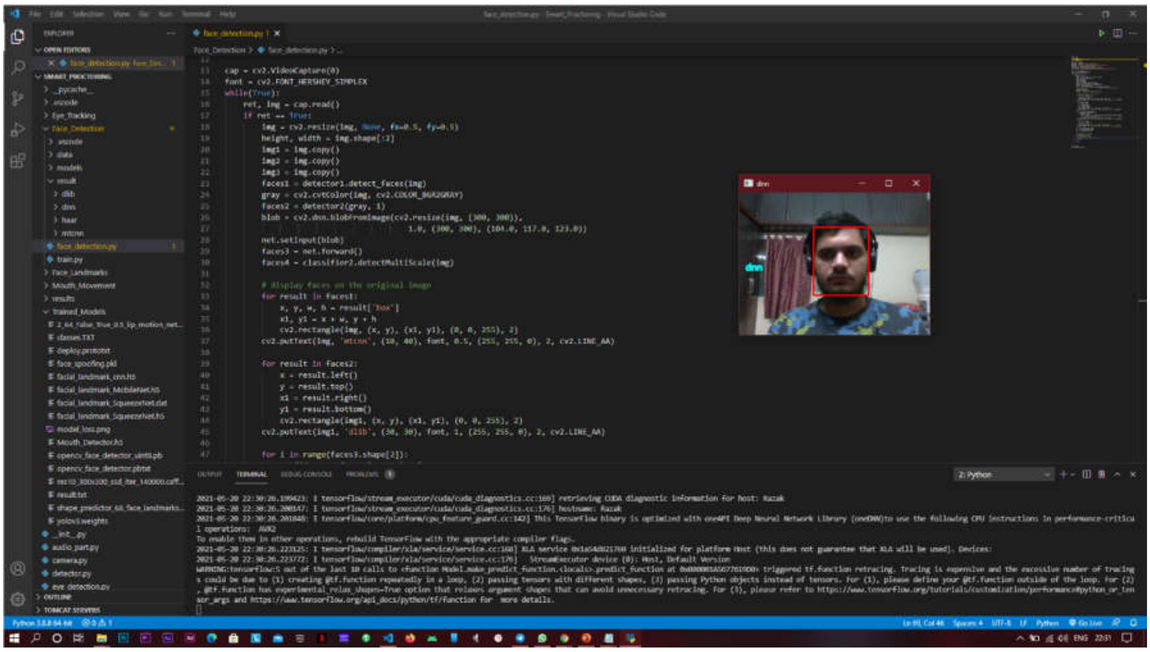

Fig 8.1.1 Detecting Face

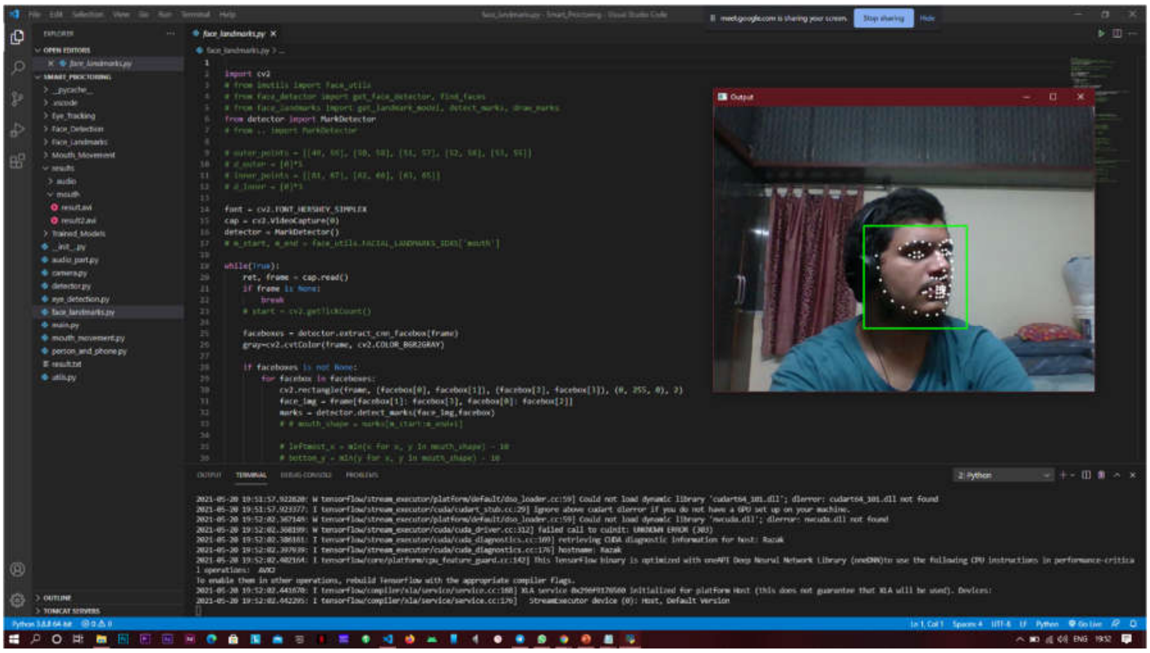

Fig 8.1.2 Detecting Face Landmarks

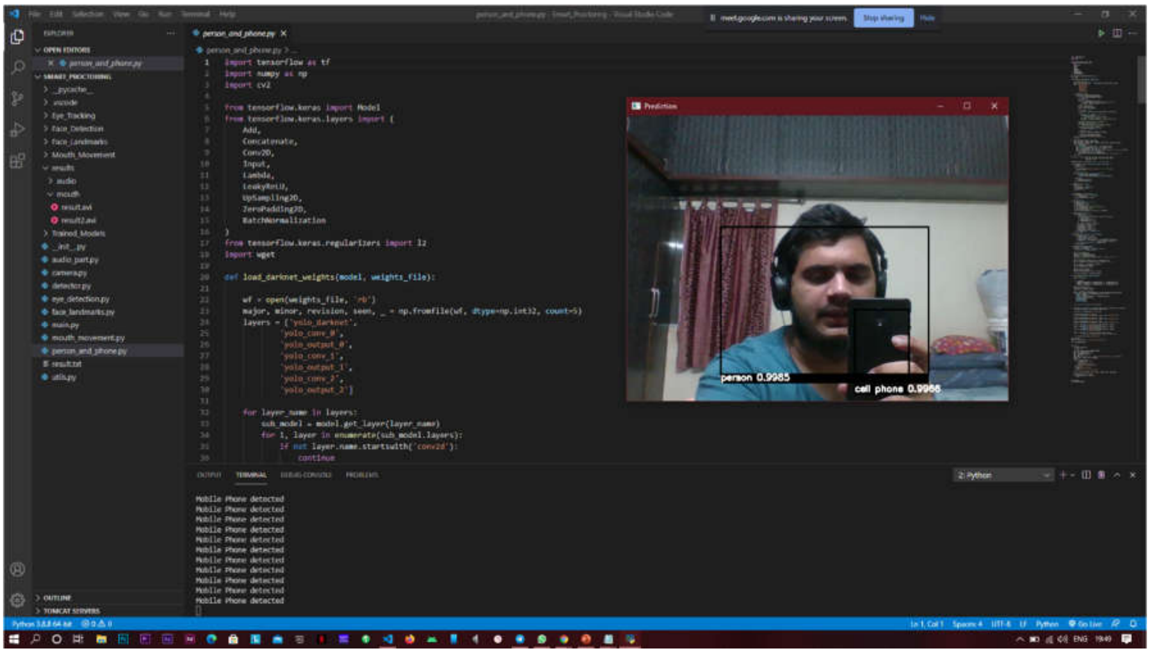

Fig 8.1.3 Detecting phone

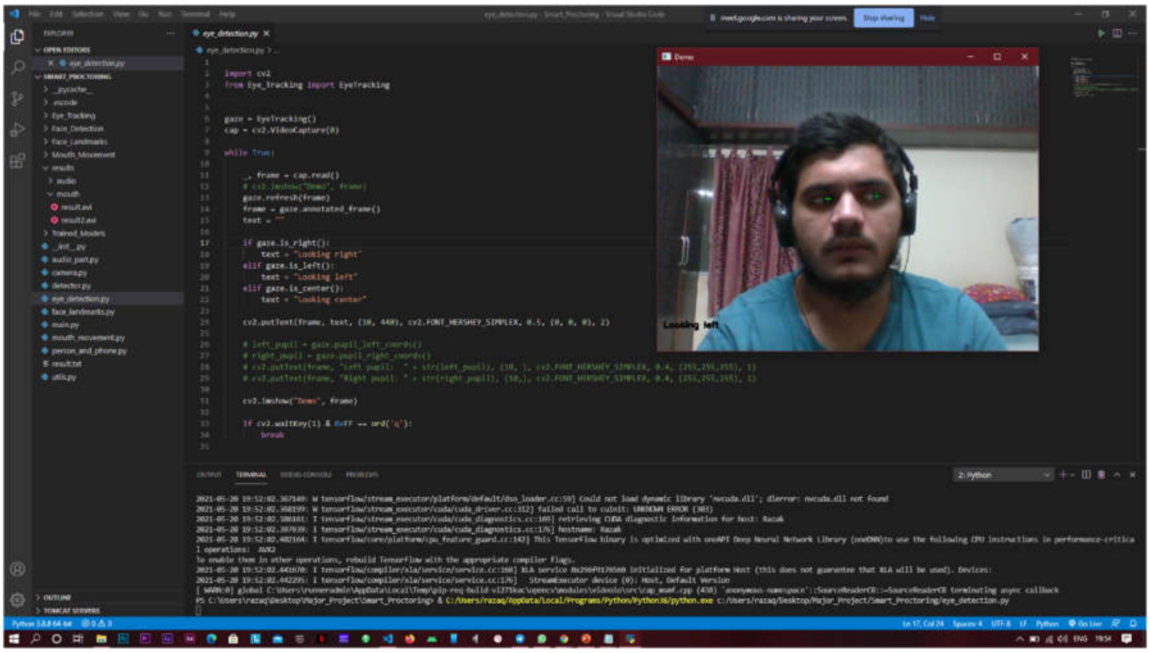

Fig 8.1.4 Tracking Eye-ball

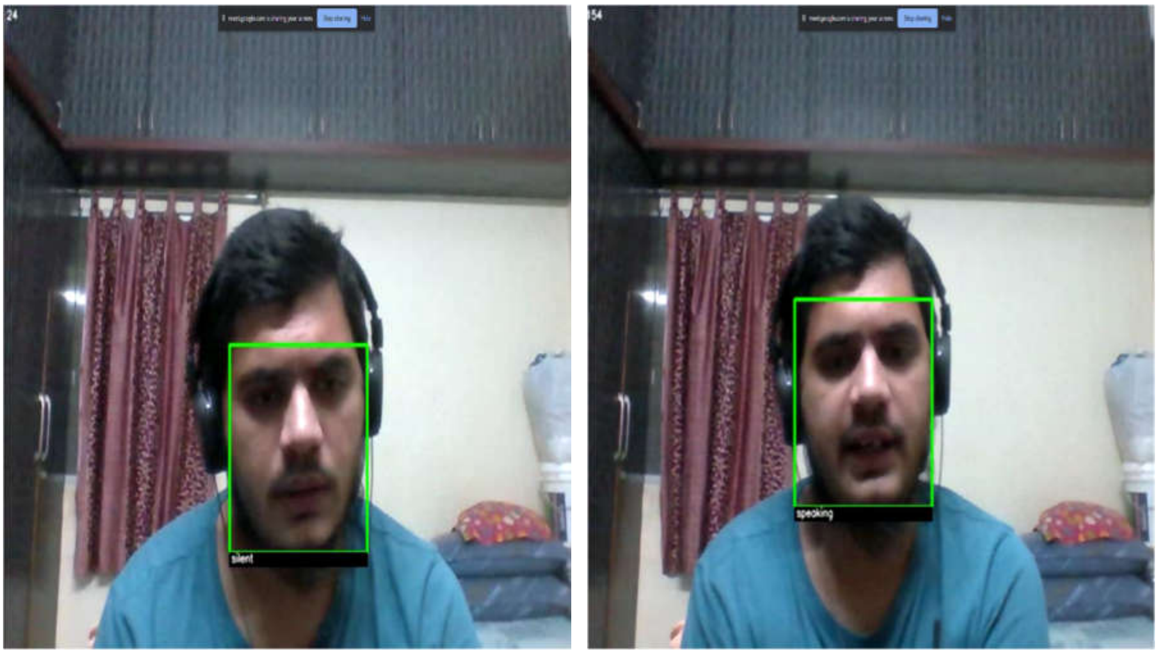

Fig 8.1.5 Detecting Mouth Movement

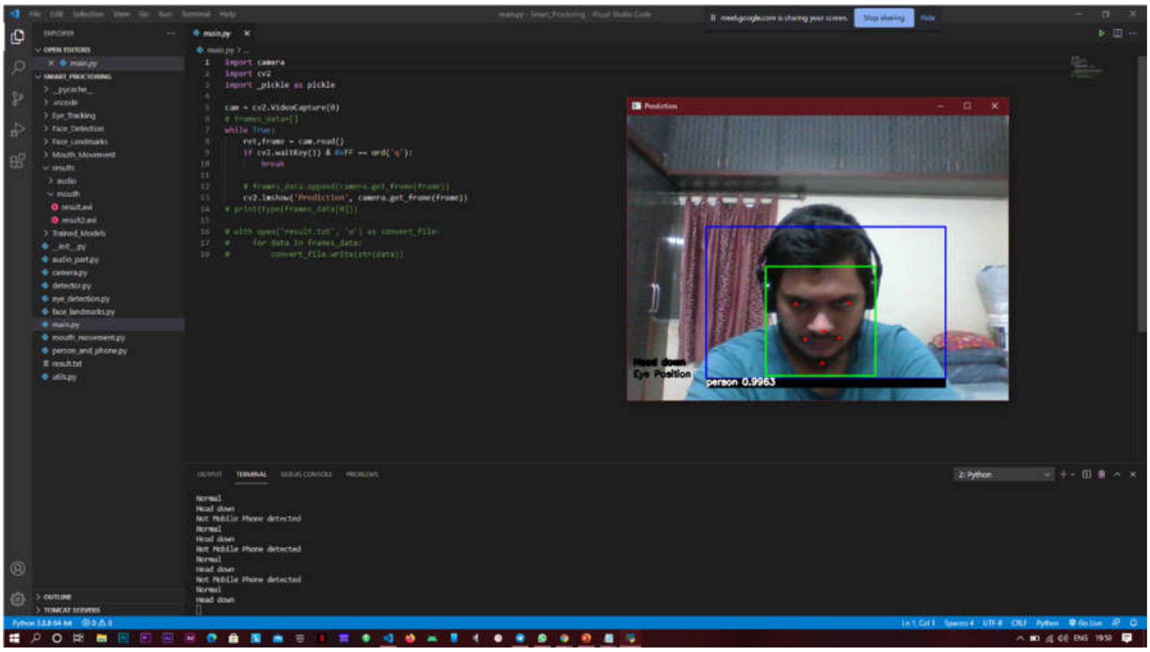

Fig 8.1.6 Detecting Head pose

VI. CONCLUSION

n conclusion, the intelligent web-based tracking system powered by artificial intelligence represents a significant advancement in the field of tracking and monitoring. By combining AI algorithms with a user-friendly web interface, the system enhances accuracy, automates data processing, and provides valuable insights for decision-making. While challenges such as data privacy and integration need to be addressed, the potential benefits of implementing such a system are substantial. As organizations increasingly seek to leverage AI technologies, this research contributes to the development of smarter, more efficient tracking solutions across various domains, ultimately improving operational effectiveness and user satisfaction. Future work will focus on expanding the system's capabilities and exploring additional applications to further enhance its utility.

10. REFERENCES

[1] Neelesh Chandra M, Piyush Sharma, Utkarsh Tripathi, Ujwal Kumar and Dr. G.C. Bhanu Prakash, 'Automating Online Proctoring Through Artificial Intelligence' IRJET, Volume: 08 Issue: 01, Jan 2021.

[2] Weiqing Wang, Kunliang Xu, Hongli Niu, Xiangrong Miao," Emotion Recognition of Students Based on Facial Expressions in Online Education Based on the Perspective of Computer Simulation", Complexity, vol. 2020, Article ID 4065207, 9 pages, 2020. https://doi.org/10.1155/2020/4065207

[3] S. Prathish, A. N. S. and K. Bijlani, "An intelligent system for online exam monitoring," 2016 International Conference on Information Science (ICIS), 2016, pp. 138-143, doi: 10.1109/INFOSCI.2016.7845315.

[4] "Fraud detection in video recordings of exams using Convolutional Neural Networks", Aiman Kuin, University of Amsterdam, June 20, 2018.

[5] "e-Invigilator: A biometric-based supervision system for eAssessments", N.L Clarke, P.Dowland, S.M. Furnell International Conference on Information Society (iSociety 2013), 24-26 June 2013.

[6] Wolfram Research," FER-2013" from the Wolfram DataRepository (2018).

[7] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journalof Machine Learning Research 10, pp. 1755-1758, 2009

[8] Bradski, G., Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library." O'Reilly Media, Inc."

[9] https://arxiv.org/abs/1804.02767v1

[10] Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham.