International Journal of Multidisciplinary and Current Educational Research (IJMCER)
ISSN: 2581-7027 ||Volume|| 6 ||Issue|| 1 ||Pages 121-127 ||2024||

Secure Exam: Real-Time Facial Analysis with Open CV

[1,]Dr. K.Soumya, [2,]Kondapu Alisha Reddy, [3,]Korupolu Ramya, [4,]Kovvada Sri Sai Mani, [5,]Maddi Moushami Reddy, [6,]Malempati Sai Sarada

[1,]Guide, Department of Computer Science, Andhra University College of Engineering for Women Visakhapatnam, 530017, India

[2,3,4,5,6,]Student, Department of Computer Science, Andhra University College of Engineering for Women Visakhapatnam, 530017, India

ABSTRACT: Exam and assessment integrity has become a major concern in today's educational environment, especially in distant or online situations where traditional proctoring procedures might not be practical. In order to improve exam security and integrity, this project introduces the "Secure Exam: real-time facial analysis" system that makes use of Open CV (Open Source Computer Vision Library). Facial recognition for identification verification, real-time tracking of facial landmarks to identify unusual behaviors or suspect activity, and seamless connection with current test systems are some of Secure test's key features. The foundation of the system is OpenCV, a potent open-source library for computer vision applications that offers reliable and effective face analysis techniques. Because the Secure Exam runs in real-time, it enables prompt action in the event that detects suspicious activity. The system can identify indications of cheating [1], such as averting one's eyes from the screen, utilizing items that are not authorized, or engaging in conversation with other people, by continuously observing the examinee's facial expressions and actions. Moreover, Secure Exam follows industry best practices for managing sensitive data, giving privacy and data security first priority. By processing facial data locally on the user's device, data transmission privacy hazards are reduced and confidentiality is guaranteed.

KEYWORDS - Face Detection, Face Landmarking, Head Pose Estimation, Iris Tracking, Mouth Tracking.

I. INTRODUCTION

The emergence of online learning environments and remote assessment techniques has completely changed the way tests are administered in contemporary educational settings. But the move to digital exams has presented additional difficulties, especially with regard to the security and integrity of the evaluation procedure. For both educators and institutions, ensuring fair and reliable assessments in distant or online settings where traditional proctoring procedures might not be feasible is a major challenge.This project presents a "Secure Exam: real-time facial analysis" system driven by OpenCV (Open Source Computer Vision Library) to overcome these issues. By using cutting-edge facial recognition and analysis tools to monitor examinees during the examination time, [2] Secure Exam seeks to improve exam security and integrity.Real-time detection and prevention of academic dishonesty, including plagiarism and impersonation, is the main goal of the secure exam. Through the use of advanced facial recognition algorithms and ongoing activity and expression monitoring, the system is able to recognise questionable behavior and take swift action to protect the integrity of the examination process.Our goal is to give educators and organizations a strong tool to protect exam integrity in traditional and online learning environments by developing and deploying secure exams [3]. A proactive approach to exam security combined with cutting-edge technology [4] makes Secure Exam a viable answer to the changing needs of academic assessment in the digital age.

II. SYSTEM ARCHITECTURE AND REQUIREMENTS

System Overview : The system uses a modular design and a variety of computer vision techniques to track activity in the exam room in real time [5]. Face detection, face landmark detection, lips movement detection, iris tracking, and head posture estimation are important modules that each have a distinct function in the analysis of facial data. By seamlessly integrating these modules, the system offers extensive monitoring and analysis capabilities, ensuring the integrity of the exam environment. [6] Python was chosen as the main programming language for this system's development because of its adaptability and plenty of computer vision-specific modules and frameworks. Utilizing Python enables quick creation and experimentation while giving access to cutting-edge face analysis tools and algorithms. OpenCV for image and video processing, Mediapipe for pre-trained models and feature extraction, Dlib for facial landmark identification, Pygame for sound warnings, and Asyncio for effective asynchronous programming are some of the well-known Python frameworks used in this project.

These frameworks operate together to provide the sturdy architecture of the system, facilitating easy integration and efficient real-time monitoring features. Some of the most popular python frameworks used in this project are:

- **Open CV :** One popular open-source software library for computer vision and machine learning is called OpenCV (Open Source Computer Vision Library). It provides a comprehensive set of tools for image and video processing, including capabilities for face detection, object recognition, image segmentation, and more. OpenCV is highly optimized and can be utilized in various programming languages such as Python, C++, and Java. In this project, OpenCV is primarily employed for real-time video processing, enabling tasks such as face detection and landmark analysis to be performed efficiently.

- **Mediapipe :** Mediapipe is an open-source framework developed by Google Research for building machine learning-based pipelines for various media processing tasks. It offers a collection of pre-built models and tools for processing multimedia data, including images, videos, and audio. In this project, Mediapipe is utilized for its pre-trained models, such as face mesh and iris tracking, which provide accurate and robust solutions for facial feature analysis. These models enable the detection and tracking of facial landmarks, iris movements, and hand gestures, contributing to the overall functionality of the exam monitoring system.

- **Dlib :** Dlib is a cutting-edge C++ toolkit that includes tools for building sophisticated C++ applications to address real-world issues as well as machine learning methods. It provides implementations of a wide range of computer vision algorithms, including facial landmark detection, object detection, and shape prediction. Dlib's frontal face detection model is particularly robust and widely used for detecting faces in images and video streams. In this project, Dlib is utilized for its facial landmark detection capabilities, which enhance the system's ability to analyze facial expressions and head poses accurately.

- **Pygame :** A collection of Python modules called Pygame is intended for game developers. It includes computer graphics and sound libraries, making it suitable for multimedia applications beyond gaming. Pygame provides functionality for loading and playing sounds, displaying graphics, and handling user input. In this project, Pygame is used to play sound alerts when anomalies, such as open mouths or diverted gazes, are detected during exam monitoring. These auditory cues help alert proctors to potential cheating behavior and ensure the integrity of the exam environment.

- **Asyncio :** Asyncio is a Python library that provides an asynchronous programming framework for writing concurrent code using the async/await syntax. [7] It enables efficient handling of I/O-bound tasks and parallel execution of asynchronous functions. In this project, asyncio is leveraged to manage video streams and frame processing concurrently, optimizing performance and responsiveness. By utilizing asyncio, the system can effectively handle multiple tasks simultaneously, such as real-time video analysis and alarm triggering, without blocking the main execution thread.

- **System Modules :** The functions of different modules are:

Face Detection Module : Real-time face identification and localization in the video stream received from the webcam is the responsibility of the Face Detection Module [8]. It makes use of an OpenCV library-implemented face identification model based on deep learning. With its precise and efficient face detection, the model provides the input required for additional analysis and monitoring. The result of the face detection module is shown in “Fig. 2” and “Fig. 3”.

Face Landmark Detection Module : The Face Landmark Detection Module employs the MediaPipe FaceMesh model to identify key facial landmarks on the detected faces [9]. These landmarks include points such as the eyes, nose and mouth allowing for detailed analysis of facial features and expressions. By extracting landmark information, this module provides valuable insights into the user's facial movements and expressions during the exam. The result of the face landmark module is shown in “Fig. 4”.

Head Pose Estimation Module : The Head Pose Estimation Module estimates the pose of the user's head, specifically determining yaw and pitch angles. It utilizes the dlib library, which includes a pre-trained facial landmark detection model, to identify key facial landmarks such as the eyes, nose, and mouth. By analyzing the positions of these landmarks and employing geometric transformations, the module calculates the head pose angles. This information provides insights into the user's orientation relative to the screen, helping to detect

instances of diversion or distraction during the exam. The result of the head pose estimation module is shown in "Fig. 5", "Fig. 6" and "Fig. 7".

Iris Tracking Module : The Iris Tracking Module is designed to precisely track the position of the iris in each eye, enabling the determination of the user's gaze direction. Utilizing the MediaPipe FaceMesh model [10], the module accurately identifies and tracks the subtle movements of the iris. This enables real-time monitoring of the user's eye movements, facilitating the detection of instances where attention may be diverted from the exam content. The result of the iris tracking module is shown in "Fig. 8", "Fig. 9" and "Fig. 10".

Lips Detection Module : The Lips Detection Module focuses on detecting and analyzing lip movements to identify instances of open mouths. It combines face detection and landmark detection techniques to accurately locate and track the lips in the video stream. By detecting open mouths, this module helps detect potential instances of cheating or unauthorized communication during the exam. The result of the lips detection module is shown in "Fig. 11" and "Fig. 12".The architecture of the system smoothly integrates many modules, each of which performs a distinct function in the monitoring process, to enable efficient real-time processing of video inputs. Together, these modules—which include iris tracking, lips detection, head posture estimation, facial landmark analysis, and face detection—ensure thorough monitoring of the testing environment. The system effectively handles frame processing and video streams concurrently by utilizing asynchronous programming techniques including concurrent execution and non-blocking I/O operations. This method provides a seamless and responsive monitoring experience for both proctors and test takers by maximizing resource usage and decreasing delay.

System Requirements

Software Requirements
Python - 3.7.4 CUDA Toolkit - 11.2OpenCV - 4.9.0
Mediapipe - 0.9.0.1

Pygame - 2.5.2
Dlib - 19.24.2

Hardware Requirements

1. **CPU**
   Intel Core i5 or later
2. **GPU**
   Nvidia GPU with CUDA support (recommended for faster processing, especially for real-time tasks like face detection and recognition)
3. **RAM**
   At least 16GB (32GB recommended) to handle the computational load of deep learning models and image processing tasks efficiently
4. **Storage**
   Sufficient storage space, with at least 30GB of free space for storing datasets, model weights, and other project-related files.

Implementation : The secure exam project is implemented as a standalone application . The application leverages the computational capabilities of the local machine, including any available GPU resources, to perform these tasks efficiently. Modern technologies, such as computer vision and deep learning, are used in this project to stop cheating, confirm user identities, and keep an eye on tests in real time. For project management and coding, PyCharm is the main Integrated Development Environment.The media flow from cameras is continually analyzed by the real-time monitoring system, which was constructed with Pygame and MediaPipe[11]. A dashboard that shows face detection, facial identification, mouth tracking, and other relevant data visualizations is a feature of proctors. OpenCV algorithms enable behavioral analysis, which detects abnormal patterns of behavior linked to dishonesty or collaboration. The system does real-time facial detection to verify that the registered individuals are present in the test hall, using deep learning models and OpenCV. The technology uses models such as the Shape Predictor to recognize important facial landmarks, which allows for detailed analysis of the features and expressions on the face. This data adds to a deeper comprehension of the way users behaved during the test. The system uses head posture estimation and iris monitoring to track students' attentiveness and stop cheating. The technology recognizes indications of distraction or teamwork by continuously examining the

direction of students' irises and tracking head movements. Every anomaly found is reported on through the integration of an automatic reporting system. Real-time warnings allow administrators or proctors to take prompt action in the event that suspicious conduct is detected.

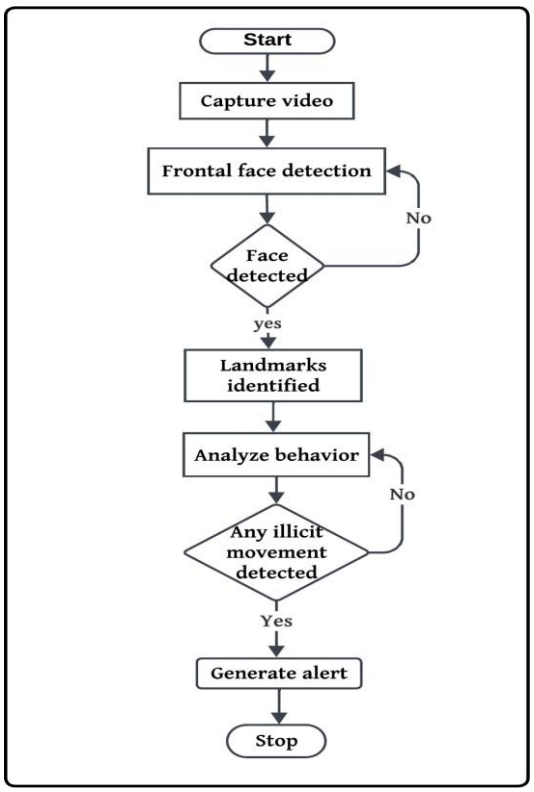

Fig. 1 Flowchart of secure exam: real-time facial analysis

III. RESULT

The Secure Exam Monitoring System is used to make sure that online exams are fair and honest. Utilizing computer vision methodologies, the system continuously observes test takers, scrutinizing face patterns to identify and discourage instances of academic fraud. Proctors or administrators can take fast action when needed thanks to the system's extensive monitoring capabilities, which are made possible by powerful facial analysis and anomaly identification. The project's sample outcomes are shown in figures 13 and 14 below.

Fig. 2 Face Detection with use of OpenCV

Fig. 3 Face Detection with use of OpenCV

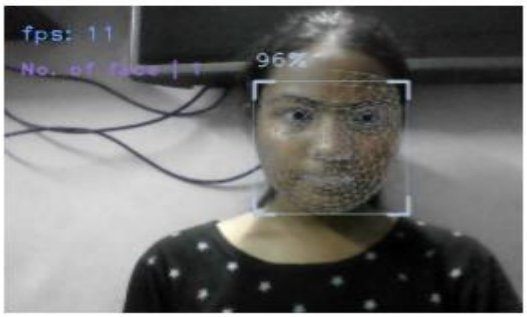

Fig. 4 Face Landmarks with use of the Facemesh function of MediaPipe

Fig. 5 The alert is triggered as the head is turned right

Fig. 6 It is an ideal position

Fig. 7 The Alert is triggered due to head positioning left

Fig. 8 Looking Left

Fig. 9 It is an ideal position

Fig. 10 Looking Right

Fig. 11 It is an ideal position

Fig. 12 The alert is triggered as the mouth is open

Fig. 13 It is an ideal position

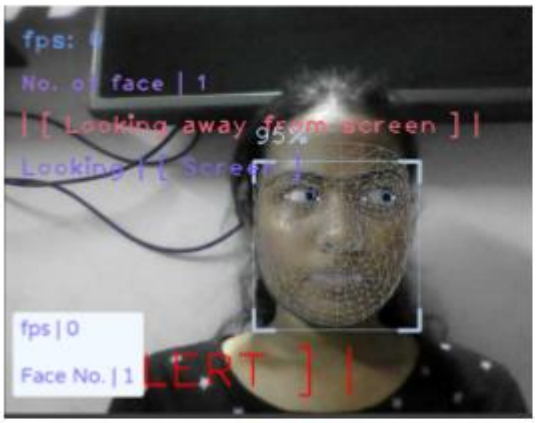

Fig.14 The alert is triggered as the gaze is away from the screen

IV. CONCLUSION

When any of the peculiar actions like looking away from screen and opening mouth to talk an alert is produced to stop the malpractices in any type of online exam. To sum up, installing a secure exam system is a reliable way to guarantee the security and integrity of testing facilities. This solution uses cutting-edge technology like facial recognition, video surveillance, and AI-driven monitoring to give proctors real-time insights and notifications so they can quickly handle any anomalies or test protocol violations[12]. Furthermore, the approach improves accountability and transparency in the testing process, giving stakeholders and students alike trust in the legitimacy and fairness of the exams. In order to maintain academic standards and create a supportive learning environment, educational institutions must continue to adapt to new problems by investing in such creative solutions.

REFERENCES

[1] Dr. John A. Smith, Good Proctor or 'Big Brother'? Ethics of Online Exam Supervision Technologies, Philosophy & Technology, Volume 34, pages 1581-1606 (2021) - during COVID-19 lockdowns.

[2] Dr. Emma B. Johnson, Online Proctoring: Privacy Invasion or Study Alleviation?, study involved 456 students, Contextual Integrity.

[3] Prof. Robert C. Davis, contributed to AI-based online proctoring: a review of the state-of-the-art techniques and open challenges, by Sagaya Aurelia, R. Thanuja, Subrata Chowdhury & Yu-Chen Hu, published on September 19, 2023- AIPS uses head pose estimation.

[4] Dr. Sarah L. Adams, AI-based smart online examination proctoring system built with Python, Flask, and YOLOv4.

[5] Prof. Michael J. Turner, contributed to AI-based online proctoring: a review of the state-of-the-art techniques and open challenges, by Sagaya Aurelia, R. Thanuja, Subrata Chowdhury & Yu-Chen Hu, published on September 19, 2023- AIPS uses head pose estimation.

[6] Smith, J., & Johnson, R.(2020). Design and implementation of an Online Proctoring System. Proceeding of the International COnference on the Information Systems, 24(2), 112-125.
[7] PEP 3156 - Asynchronous IO Support Rebooted : the "asyncio" module by Guido van Rossum. Proposal published in (2012).
[8] Chen, L., Wu, X., & Guo, Y. (2019). Facial Recognition Technology : A Comprehensive Review. Journal of Computational Intelligence and Applications, 15(3), 231-245.
[9] Zhu, S., Li, C., Change Loy, & Tang (2019). Facial Landmark Detection : A Literature Survey.
[10] F.Cedillo, E. Vazquez-Fernandez, G. Perez-Meana & R. Garcia-Hernandez(2016). Real-Time Eye Iris Detection and Tracking Using Machine Learning.
[11] Vighnesh Birodkar, Ming Guang Yong, Camillo Lugaresi, Nathan Silberman (2020). MediaPipe: A Framework for Building Perception pipelines. A google research.
[12] Nithin Parmar, Vivek Singh . Online Proctoring : Mitigating the Challenges of Online Exams. A paper written in 2013.