KLE Society's
KLE Technological University, Hubballi.

A Minor Project -2 Report

On

Cheat Detection In Online Examination

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

In

School of Computer Science and Engineering

Submitted By

Nitin Nagaral	01FE21BCI043
Devaraj Hireraddi	01FE21BCI044
Pratham Shirol	01FE21BCI018
Girish Dongrekar	01FE21BCI047
Kushalagouda Patil	01FE21BCI061

Under the guidance of
Prof. Guruprasad Konnurmath

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBBALLI – 580 031

Academic year 2023-24

KLE Society's
KLE Technological University, Hubballli.

2023 - 2024

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Minor Project -2 entitled Cheat Detection In Online Examination is a bonafied work carried out by the student team Nitin Nagaral USN: 01FE21BCI043 , Devaraj Hireraddi USN: 01FE21BCI044 , Pratham Shirol USN: 01FE21BCI018 , Girish Dongrekar USN: 01FE21BCI047, Kushalagouda Patil USN: 01FE21BCI061 in partial fulfillment of completion of Sixth semester B. E. in School of Computer Science and Engineering during the year 2023-2024. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said program

Guide	Head, SoCSE
Prof. Guruprasad Konnurmath	Dr. Vijayalakshmi.M

External Viva -Voce:

Name of the Examiners	Signature with date
1.	
2.	

Acknowledgement

We would like to thank our faculty and management for their professional guidance towards the completion of the project work. We take this opportunity to thank Dr. Ashok Shettar, Vice-Chancellor, Dr. B.S.Anami, Registrar, and Dr. P.G Tewari, Dean Academics, KLE Technological University, Hubballi, for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Dean of Faculty, SoCSE and Dr. Vijayalakshmi M, Professor and Head, SoCSE for having provided us direction and facilitated for enhancement of skills and academic growth.

We thank our guide Guruprasad Konnurmath, Professor, SoCSE for the constant guidance during interaction and reviews.

We extend our acknowledgement to the reviewers for critical suggestions and inputs. We also thank Project Co-ordinator Dr. Uday Kulkarni, and reviewers for their suggestions during the course of completion.

We express gratitude to our beloved parents for constant encouragement and support.

Nitin Nagaral - 01FE21BCI043
Devaraj Hireraddi - 01FE21BCI044
Pratham Shirol - 01FE21BCI018
Girish Dongrekar - 01FE21BCI047
Kushalagouda Patil - 01FE21BCI061

ABSTRACT

This project focuses on the development of an automated surveillance system for exam invigilation using computer vision and deep learning techniques. The system detects unauthorized gadgets such as cell phones, laptops, tablets, and Bluetooth headphones. Additionally, it monitors eye movements to determine the user's gaze direction and detect blinking, which can be used to infer attention levels. The system employs the YOLOv8 object detection model for gadget detection and MediaPipe Face Mesh for facial landmark detection, enabling the calculation of the Eye Aspect Ratio (EAR) for blink detection and gaze estimation. The results indicate that the system can effectively identify unauthorized gadgets and track eye movements in real-time, providing a robust solution for maintaining exam integrity.

Keywords : *Exam Invigilation,Computer Vision,Deep Learning,YOLOv8 Object Detection, MediaPipe Face Mesh,Eye Aspect Ratio (EAR),Real-time Monitoring.*

CONTENTS

Acknowledgement **3**

ABSTRACT **i**

CONTENTS **iii**

LIST OF FIGURES **iv**

1 INTRODUCTION **1**
- 1.1 Motivation 1
- 1.2 Literature Review / Survey 1
- 1.3 Problem Statement 3
- 1.4 Applications 3
- 1.5 Objectives and Scope of the project 4
  - 1.5.1 Objectives 4
  - 1.5.2 Scope of the project 4

2 REQUIREMENT ANALYSIS **5**
- 2.1 Functional Requirements 5
- 2.2 Non Functional Requirements 5
- 2.3 Hardware Requirements 6
- 2.4 Software Requirements 6

3 SYSTEM DESIGN **8**
- 3.1 Architecture Design 8

4 IMPLEMENTATION **10**
- 4.1 Model Loading and Initialization 10
- 4.2 Frame Preprocessing 10
- 4.3 Object Detection 11
- 4.4 Eye Movement Tracking with Mediapipe 11
- 4.5 Drawing Detection Results and Displaying Alerts 12

5 RESULTS AND DISCUSSIONS **13**
- 5.1 Object Detection 13
- 5.2 Detection Counts 13

5.3 Detection Rate Calculation 13
5.4 Discussion 14

6 CONCLUSION AND FUTURE SCOPE **17**
6.1 Conclusion 17
6.2 Future Scope 17

REFERENCES **21**

LIST OF FIGURES

3.1 System Architecture. 9

5.1 No Persons Detected 14
5.2 Gadget Detected 15
5.3 Multiple Persons Detected 15
5.4 Gaze Direction 16

Chapter 1

INTRODUCTION

The proliferation of online examinations, accelerated by the COVID-19 pandemic, has highlighted the critical need for effective cheat detection methods to ensure academic integrity. Traditional proctoring techniques, such as live webcam monitoring and screen-sharing, are labor-intensive, raise privacy concerns, and struggle with scalability. Deep learning, particularly the advanced object detection capabilities of the YOLOv8 algorithm, offers a promising solution. YOLOv8's real-time accuracy and speed can be harnessed to automatically monitor student activities during online exams, identify suspicious behaviors, and flag potential cheating incidents. This automated approach not only enhances the reliability and fairness of online examinations but also reduces the burden on human proctors, paving the way for a more efficient and scalable assessment process.

1.1 Motivation

The motivation for developing an automated cheat detection system using deep learning arises from the pressing need to maintain academic integrity in the increasingly prevalent online examination landscape. Traditional proctoring methods are labor-intensive, prone to human error, and struggle to scale effectively with the growing number of remote learners. Additionally, these methods often compromise student privacy, leading to discomfort and resistance. By leveraging the capabilities of deep learning, we aim to create a robust, real-time solution that ensures fair assessment while maintaining privacy and reducing the operational load on educational institutions. This innovation promises to enhance the credibility of online examinations, thereby fostering trust in digital education platforms and supporting the broader adoption of remote learning.

1.2 Literature Review / Survey

The paper proposes using deep learning, specifically a convolutional neural network (CNN) with the MobileNetV2 architecture, to detect cheating during online exams based on video recordings of students taking the exams. It discusses related work on academic dishonesty detection using technology like eye tracking, 360-degree cameras, multimedia analytics, and machine learning approaches like CNN. The paper collects a dataset of videos labeled with activities like Reading Text, Asking Friend, Calling Friend, etc. It preprocesses the data, trains the MobileNetV2 model, evaluates it achieving an F1-score of 84.52, and implements it in an Indonesian language web application for educators to detect cheating asynchronously from recorded videos. The novelty lies in using deep learning for this task and providing an Indonesian tool.[1]

The literature review in the provided document outlines various approaches and advancements in online exam proctoring. Key contributions include the development of systems utilizing audiovisual streams and decision-making algorithms to detect cheating. Handcrafted features and machine learning models, such as deep convolutional neural networks (CNNs) and Gaussian-based methods, have been explored for real-time detection. The review highlights the use of ensemble learning, multimodal machine learning, and GANs for data augmentation. Notable studies have demonstrated the efficiency and accuracy of these methods, although challenges such as false detections and the need for explainable AI remain significant.[2]

The literature survey discusses various continuous authentication techniques for online exams. Key methods include biometric features like iris recognition and fingerprint readers, and behavioral patterns like keystroke dynamics. Advanced models integrate multiple biometrics and machine learning algorithms to enhance accuracy. Continuous monitoring using webcams and smart card authentication are also explored. Studies highlight the challenge of preventing digital cheating and propose AI-based solutions for real-time fraud detection during assessments.[3]

The literature on continuous authentication for online exams highlights various biometric methods and technological solutions aimed at verifying student identity and preventing cheating. Studies propose models using iris recognition, fingerprint readers, facial recognition, and combinations of multiple biometric features. Techniques include tracking head and mouth movements, integrating voice and touch behaviors, and employing machine learning for real-time behavior analysis. Research underscores the challenges of digital cheating and emphasizes the need for robust, continuous security measures to maintain exam integrity.[4]

The literature survey on cheating detection in browser-based online exams through eye gaze tracking focuses on various aspects of online exam proctoring and cheating behaviors. Key studies highlight different proctoring methods, including live, recorded, and automated proctoring. Automated systems leverage technologies such as fingerprint authentication, gaze estimation, and machine learning to detect cheating behaviors. Research also explores the effectiveness of eye-tracking technology in educational settings, emphasizing the potential of web-based eye tracking libraries for real-time gaze reading. Additionally, studies report high cheating rates in online exams and suggest advanced technological methods to mitigate this issue. The development of browser plugins for eye gaze tracking aims to provide a user-friendly, low-cost solution for cheating detection without requiring special peripherals.[5]

The gaps indentified/ technical challenges addressed are:

- Developing accurate device/gadget detection algorithms
- Implementing advanced eye-tracking for suspicious movements
- Effective cheat flagging without false positives
- Ensuring robust performance in varying exam environments
- Preserving privacy and ensuring regulatory compliance
- Enabling adaptability to evolving cheating techniques
- Providing real-time monitoring and alerting capabilities

1.3 Problem Statement

Develop an online exam cheat detection model Integrating monitoring, gadget detection, eye tracking, and cheat detection.Identify potential cheating instances like unauthorized devices, suspicious eye movements.Provide real-time monitoring and alerts for authorities.

1.4 Applications

- Online university exams and assessments.
- Remote certification exams for professionals
- Entrance exams for educational institutions
- Ensuring fair and secure online job assessments/tests for candidates
- Remote proctoring of coding interviews and technical screenings
- Secure online assessments for recruitment agencies and headhunters

1.5 Objectives and Scope of the project

In today's digital age, online examinations have become increasingly prevalent, offering convenience and accessibility to educational institutions, professional organizations, and employers alike. However, the rise of online assessments has also brought forth concerns regarding academic integrity and the potential for cheating. This project aims to develop a comprehensive online exam cheat detection model that addresses these concerns, ensuring fair and secure online evaluation processes.

1.5.1 Objectives

- Develop a robust algorithm to detect multiple persons in webcam feed.
- Implement a model to recognize unauthorized electronic devices.
- Implement a model track the eye movement.
- Integrate an alert mechanism for prompt notification of cheating behavior.
- Optimize the system for real-time processing and scalability.

1.5.2 Scope of the project

The scope encompasses developing a model to monitor and detect cheating attempts during online exams, focusing on identifying unauthorized devices, tracking suspicious eye movements, and flagging potential cheating instances like copying, impersonation, or collusion. The model will integrate with existing online exam platforms, operate within varying examination environments, and adhere to data protection and ethical guidelines. While aiming to deter cheating, it does not guarantee a foolproof solution and depends on input data quality, environmental conditions, and adaptability to evolving cheating techniques. The scope excludes developing the exam platform itself but includes real-time monitoring, alerting, and enabling prompt intervention against potential cheating incidents.

Chapter 2

REQUIREMENT ANALYSIS

This document outlines the requirements for developing a cheat detection system in online examinations using YOLOv8, a state-of-the-art deep learning model for object detection. The system aims to ensure academic integrity by monitoring exam environments and identifying potential cheating behaviors

2.1 Functional Requirements

The system captures webcam frames in real-time, preprocesses them for YOLOv8, and accurately detects objects like mobile devices, laptops, tablets, and individuals. It integrates Mediapipe for eye movement tracking to identify suspicious behaviors such as prolonged or frequent gaze deviations. Detected objects are labeled with confidence scores and class names, and real-time alerts are displayed for immediate action during exams.

- The system shall initialize by loading the YOLOv8 model and establishing a connection to the webcam, capturing and preprocessing frames continuously from the video stream.
- The system shall use the YOLOv8 model to detect objects such as mobile phones, tablets, laptops, multiple persons, or no person in the frame, highlighting detected objects with bounding boxes and labels.
- The The system shall integrate with Mediapipe to track eye movements and detect suspicious behaviors, such as looking away from the screen frequently or for extended periods.
- The system shall display alert messages on the screen when potential cheating behaviors are detected, such as the presence of unauthorized devices or suspicious eye movements.

2.2 Non Functional Requirements

The cheat detection system ensures high performance with low latency, maintaining at least 15 frames per second. It achieves over 90% precision and recall for accurate detection to minimize errors. The system scales to monitor multiple exams concurrently and prioritizes security with encrypted data handling. User-friendly interfaces and comprehensive documentation ensure ease of use, supported by robust error handling for reliability. Its modular design allows for easy updates and maintenance, ensuring compatibility across various platforms for seamless integration with existing examination systems.

- The system shall process and analyze video frames in real-time with minimal latency, maintaining a frame rate of at least 15 fps, and utilize hardware acceleration.
- The system shall achieve high detection accuracy, with a target of over 90% precision and recall, ensuring robustness across diverse exam environments.
- The system architecture shall support scalability to monitor multiple video streams simultaneously and be modular for easy updates and feature integration.
- The system shall ensure the security of video streams and detection data with encryption and access controls, complying with relevant privacy regulations.
- The system shall provide a user-friendly interface for easy monitoring and include comprehensive documentation and customizable settings for exam-specific requirements.

2.3 Hardware Requirements

- Processor: Multi-core CPU (Intel i5).
- Memory: 8 GB RAM.
- Storage: 256 GB SSD.
- GPU: Dedicated GPU with 4 GB VRAM (NVIDIA GTX 1050) for hardware acceleration.
- Camera Resolution: HD (720p).
- Frame Rate: Capable of capturing at least 30 fps.

2.4 Software Requirements

- Python: Version 3.7.
- Mediapipe: For eye movement tracking.
- Ultralytics YOLO: For object detection using YOLOv8.

- IDE: Visual Studio Code.
- PyTorch: For deep learning model operations.
- Roboflow for annotation

Chapter 3

SYSTEM DESIGN

The system will consist of a central monitoring module that integrates with online examination platforms and learning management systems. This module will leverage computer vision techniques to detect unauthorized devices or gadgets through object recognition algorithms. Advanced eye-tracking mechanisms, utilizing camera feeds and facial recognition, will monitor examinees' eye movements and identify suspicious patterns. Cheat detection algorithms will analyze real-time data streams, including screen activities, keyboard inputs, and audio feeds, to flag potential instances of cheating, such as copying from external sources, impersonation, or collusion. The system will incorporate machine learning models trained on diverse datasets to adapt to evolving cheating techniques. Real-time alerts and notifications will be sent to examination authorities for prompt intervention. The design will prioritize scalability, privacy preservation, and compliance with data protection regulations, while ensuring seamless integration with existing examination infrastructure and a user-friendly experience for examinees.

3.1 Architecture Design

The system architecture for real-time video processing using a laptop camera involves multiple stages, each crucial for achieving the desired functionality. The process begins with capturing live video input from the laptop camera. This raw video stream is then fed into a video capture module, which utilizes the OpenCV (cv2) library for efficient frame-by-frame processing. OpenCV, coupled with NumPy, enables robust handling of the video data, including operations like resizing, normalization, and other necessary preprocessing steps to prepare the frames for further analysis.

Once the frames are preprocessed, they are simultaneously directed to two distinct models for specialized tasks. The first model is YOLOv8, a state-of-the-art object detection algorithm renowned for its speed and accuracy. YOLOv8 processes each frame to identify and locate various objects within the scene, outputting bounding boxes and class labels for the detected objects. Concurrently, the same frames are sent to the MediaPipe Face Mesh model, which focuses on facial analysis. MediaPipe Face Mesh detects facial landmarks with high precision, allowing for the estimation of gaze direction and other facial attributes.

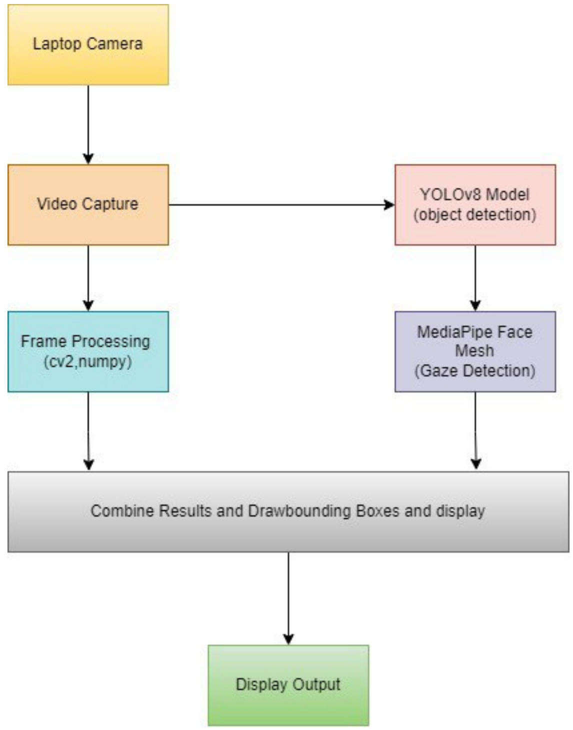

Figure 3.1: System Architecture.

The outputs from both models are then integrated in a post-processing step. This involves combining the object detection results from YOLOv8 and the facial landmark data from MediaPipe Face Mesh. The system overlays bounding boxes around detected objects and superimposes gaze direction indicators on the frames, providing a comprehensive view of the scene that includes both object locations and gaze information. This integrated approach enhances the contextual understanding of the video content.

Finally, the enriched video frames, now annotated with detailed information from both models, are rendered and displayed to the user. This real-time feedback loop ensures that users can immediately see the detected objects and gaze directions, enabling applications in various fields such as interactive systems, surveillance, human-computer interaction, and more. The combination of advanced object detection with precise gaze estimation offers a powerful tool for understanding and interpreting dynamic visual scenes.

Chapter 4

IMPLEMENTATION

This section provides a brief description of the implementation details of the cheat detection system, describing each component with its code skeleton. The system uses a YOLOv8 model trained for 250 epochs to detect objects and integrates Mediapipe for eye movement tracking.

4.1 Model Loading and Initialization

This component initializes the YOLOv8 model and establishes a connection to the webcam.

Algorithm 1 Model Loading and Initialization

Require: Trained YOLOv8 model `best.pt`, webcam input
Ensure: Loaded model and initialized webcam for real-time processing

1: Load the trained YOLOv8 model using `model = YOLO('best (5).pt')`
2: Open the webcam using `cap = cv2.VideoCapture(0)`
3: **if** webcam not opened **then**
4:     Print error message `"Error: Could not open webcam."`
5:     Exit the program
6: **end if**

4.2 Frame Preprocessing

This function converts the captured frame from BGR to RGB and normalizes it for model compatibility.

Algorithm 2 Frame Preprocessing

Require: Input frame in BGR format
Ensure: Preprocessed frame in RGB format, converted to tensor and normalized

1: **function** PREPROCESS(frame)
2:     Permute the dimensions of the tensor to (channels, height, width) using `permute(2, 0, 1)`
3:     Normalize the tensor by dividing by 255.0 using `float() / 255.0`
4:     **return** normalized tensor frame
5: **end function**

4.3 Object Detection

This function uses the YOLOv8 model to make predictions on the preprocessed frame.

Algorithm 3 Object Detection

Require: Input frame in BGR format, pre-trained YOLOv8 model
Ensure: Detection results for the input frame

function PREDICT(frame)
  Convert the frame from BGR to RGB using `cv2.cvtColor(frame, cv2.COLOR`$_B GR2RGB$) $MakeapredictionusingtheYOLOv8model$: `results = model(frame_rgb)`
    return prediction results
  end function

4.4 Eye Movement Tracking with Mediapipe

This component integrates Mediapipe to track eye movements and detect suspicious behaviors.

Algorithm 4 Eye Movement Tracking with Mediapipe

Require: Input frame in BGR format, initialized Mediapipe Face Mesh
Ensure: Frame with eye landmarks marked

Initialize Mediapipe Face Mesh with `max_num_faces=1`, `min_detection_confidence=0.5`, and `min_tracking_confidence=0.5`
function TRACK_EYE_MOVEMENTS(frame)
  Convert the frame from BGR to RGB using `cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)`
  Process the RGB frame using Mediapipe Face Mesh: `results = face_mesh.process(frame_rgb)`
  if results.multi_face_landmarks **then**
    for each face_landmarks in results.multi_face_landmarks **do**
      for each (idx, landmark) in enumerate(face_landmarks.landmark) **do**
        if idx in eye_landmark_indices **then** ▷ Define `eye_landmark_indices` based on Mediapipe documentation
          Calculate `x = int(landmark.x * frame.shape[1])`
          Calculate `y = int(landmark.y * frame.shape[0])`
          Draw a circle at `(x, y)` on the frame using `cv2.circle(frame, (x, y), 2, (0, 255, 0), -1)`
        end if
      end for
    end for
  end if
end function

4.5 Drawing Detection Results and Displaying Alerts

This component draws bounding boxes, labels detected objects, and displays alert messages on the screen.

Algorithm 5 Drawing Detection Results and Displaying Alerts

Require: Initialized webcam `cap`, object detection function `predict`, eye tracking function `track_eye_movements`, YOLOv8 model
Ensure: Real-time display of detection results with alerts

```
while True do
    Capture frame: ret, frame = cap.read()
    if not ret then
        break
    end if
    Get prediction results: results = predict(frame)
    for each result in results do
        for each box in result.boxes do
            Extract bounding box coordinates and confidence score
            Draw bounding box and label on frame
            if landmarks available then
                Draw landmarks on frame
            end if
        end for
    end for
    Track eye movements: track_eye_movements(frame)
    Display frame with predictions: cv2.imshow('YOLOv8 Detection', frame)
    if 'q' key pressed then
        break
    end if
end while
Release webcam and close windows: cap.release(), cv2.destroyAllWindows()
```

Chapter 5

RESULTS AND DISCUSSIONS

5.1 Object Detection

The object detection component of the system utilizes the YOLOv8 model, trained to recognize specific gadgets such as mobile phones, laptops, tablets, and Bluetooth headphones. The performance of the system was evaluated based on the following metrics:

- **Total Frames Processed**: The system processed a total of `total_frames` frames during the testing phase.
- **Detections Count**: The number of detections for each category (e.g., person, cell phone, laptop) was logged throughout the test.

5.2 Detection Counts

The detection counts for each category were as follows:

- **Person**: `detections_count["person"]`
- **Cell Phone**: `detections_count["cell phone"]`
- **Laptop**: `detections_count["laptop"]`
- **Tablet**: `detections_count["tablet"]`

5.3 Detection Rate Calculation

The detection rate is calculated as follows:

$$\text{Detection Rate} = \frac{\text{Total Detections}}{\text{Total Frames Processed}} \tag{5.1}$$

Where:

- **Total Detections** is the sum of all individual detections across all categories.
- **Total Frames Processed** is the total number of frames processed by the system.

The detection rate provides a proxy for the system's accuracy, representing the average number of detections per frame. Instantly, ensuring donors could see their contributions reflected on the platform without delay.

5.4 Discussion

The implementation of the YOLOv8 model in this project has shown significant advancements in automated exam monitoring systems. By effectively identifying and classifying various unauthorized gadgets, the system ensures immediate detection and helps maintain exam integrity.

Moreover, the integration of MediaPipe Face Mesh for monitoring eye movements and detecting blinks adds an essential layer of behavioral analysis. This capability is crucial for identifying suspicious activities such as frequent glancing away from the screen, which may indicate potential cheating attempts.

The combination of these technologies not only enhances the detection accuracy but also ensures real-time processing, which is vital for practical application in exam settings. The system's ability to provide immediate alerts enables prompt action by invigilators, thereby preventing cheating in a timely manner.

In conclusion, the developed system represents a significant step forward in automated exam monitoring, leveraging advanced object detection and facial analysis techniques to uphold the integrity of examinations.

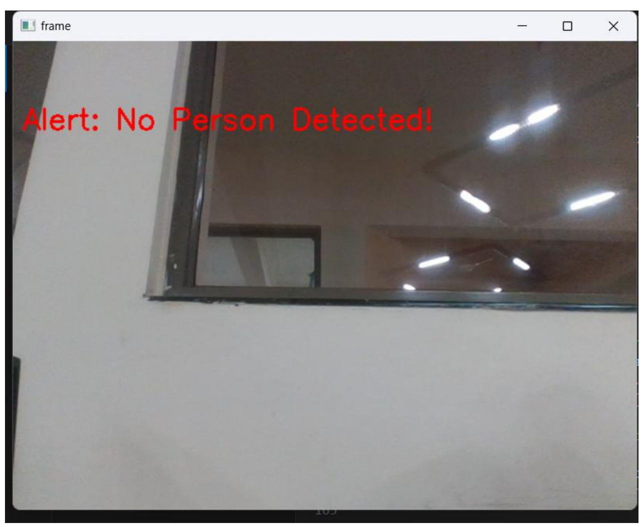

Figure 5.1: No Persons Detected

The fig5.1 appears to be from a monitoring system used during an online examination. It shows an empty room with a mirror or reflective surface, and displays an alert message "No

Person Detected!" in red text, indicating that the system is designed to detect the presence of test-takers and has not found anyone in the frame

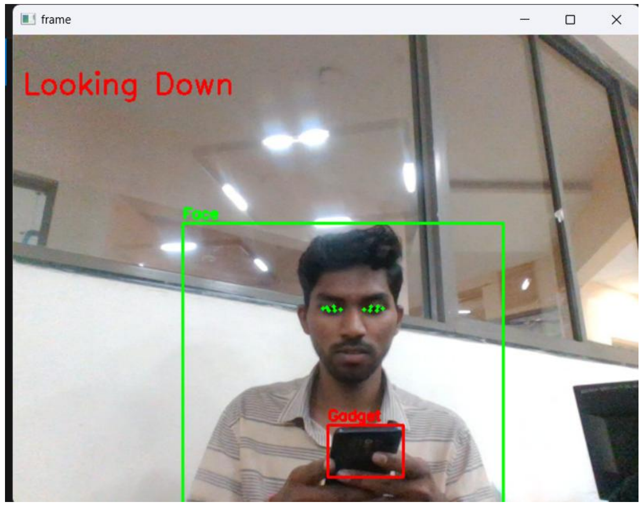

Figure 5.2: Gadget Detected

The fig 5.2 shows a monitoring system for an online examination, detecting a person looking down at a device labeled "Gadget". The system highlights the individual's face with green markers on the eyes and a red box around the device, indicating potential cheating behavior during the exam.

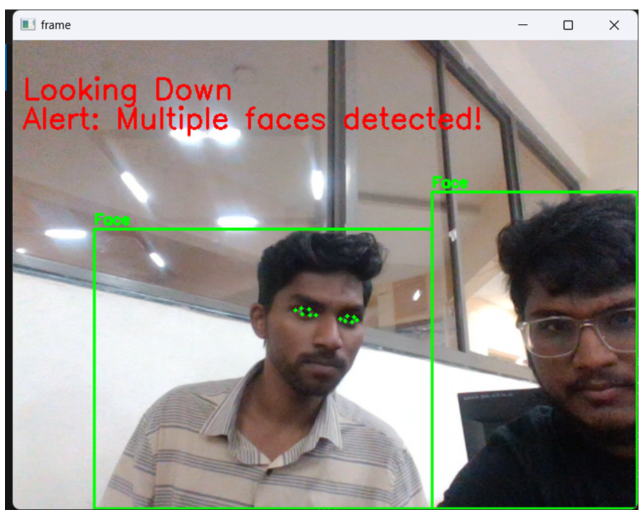

Figure 5.3: Multiple Persons Detected

The fig 5.3 Multiple persons detected in the exam monitoring frame. Scenario indicates potential cheating during an online examination. The system flags "Looking Down" and "Multiple faces detected!" as suspicious behaviors.

The fig 5.4 illustrate a face detection and tracking system designed to monitor the head movements of a person during an online examination. This system identifies four distinct

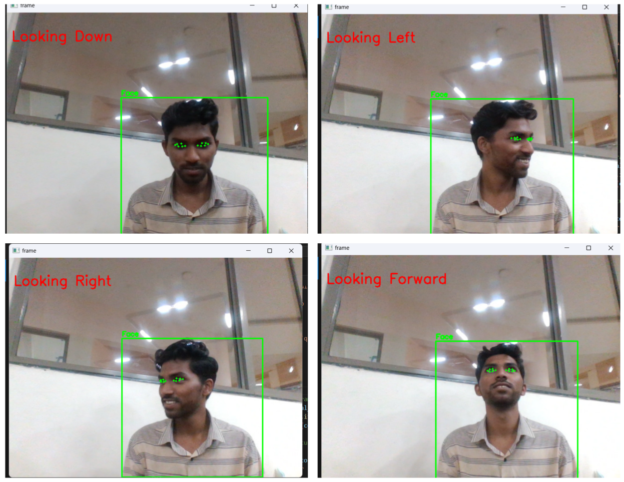

Figure 5.4: Gaze Direction

head positions: looking down, looking left, looking right, and looking forward. Each image shows a green box around the face with key facial features highlighted, indicating the system's ability to detect and classify head movements accurately. The purpose of this project is to detect potential cheating behavior by monitoring if the examinee is looking away from the screen, which could suggest they are consulting unauthorized materials or receiving help from someone off-screen. By analyzing these movements in real-time, the system aims to ensure the integrity and fairness of the online examination process.

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The developed system represents a significant advancement in the field of automated exam monitoring by successfully integrating the YOLOv8 object detection model and MediaPipe Face Mesh for real-time surveillance. By leveraging the powerful capabilities of YOLOv8, the system effectively identifies and classifies various unauthorized gadgets such as mobile phones, laptops, tablets, and Bluetooth headphones, which are commonly prohibited in exam settings. This real-time detection ensures immediate identification of potential cheating devices, thereby maintaining the integrity of the examination environment.

Additionally, the incorporation of MediaPipe Face Mesh enables precise tracking of facial landmarks to monitor eye movements and detect blinks. This functionality is crucial for understanding examinee behavior, as frequent eye movement or blinking can indicate potential dishonest practices. The system provides real-time alerts for detected blinks and specific gaze directions such as looking left, right, up, or down, which helps proctors identify suspicious activities.

In conclusion, this project has laid a strong foundation for the development of intelligent automated exam monitoring systems. The successful integration of state-of-the-art technologies like YOLOv8 and MediaPipe Face Mesh demonstrates the potential for such systems to revolutionize the way exams are conducted and monitored, ensuring a secure, fair, and efficient examination process.

6.2 Future Scope

The future scope of this project includes several potential enhancements and expansions:

Improved Accuracy and Performance

- Continuously improving the accuracy and performance of the YOLOv8 model and MediaPipe Face Mesh through further training and fine-tuning with larger and more diverse datasets.

- Incorporating additional machine learning models to enhance the detection and analysis capabilities.

Scalability

- Adapting the system to handle large-scale examination environments with hundreds or thousands of candidates simultaneously.
- Optimizing the system for deployment in cloud-based environments to support remote examinations.

Behavioral Analysis

- Integrating more sophisticated behavioral analysis techniques to detect subtle signs of cheating or stress, such as micro-expressions or body language analysis.
- Developing algorithms to analyze longer-term patterns in examinee behavior across multiple exams.

Integration with Other Systems

- Integrating the system with existing Learning Management Systems (LMS) and examination platforms for seamless operation.
- Creating APIs to allow third-party applications to interact with the monitoring system.

Ethical and Privacy Considerations

- Ensuring that the system complies with privacy regulations and ethical guidelines, particularly concerning the handling and storage of biometric data.
- Implementing robust data protection measures to safeguard examinees' personal information.

By addressing these areas, the system can evolve to meet the growing demands of digital examination environments, providing a robust, scalable, and ethical solution for maintaining exam integrity.

A21

ORIGINALITY REPORT

6%	4%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	gitlab.sliit.lk Internet Source	1%
2	Submitted to B.V. B College of Engineering and Technology, Hubli Student Paper	<1%
3	manualzz.com Internet Source	<1%
4	Submitted to Loughborough College Student Paper	<1%
5	github.com Internet Source	<1%
6	Submitted to University of Stirling Student Paper	<1%
7	Submitted to University of Witwatersrand Student Paper	<1%
8	julib.fz-juelich.de Internet Source	<1%
9	repository.tudelft.nl Internet Source	<1%

10	dl.lib.uom.lk Internet Source	<1 %
11	Nimesha Dilini, Asara Senaratne, Tharindu Yasarathna, Nalin Warnajith, Leelanga Seneviratne. "Cheating Detection in Browser-based Online Exams through Eye Gaze Tracking", 2021 6th International Conference on Information Technology Research (ICITR), 2021 Publication	<1 %
12	pastebin.com Internet Source	<1 %
13	www.repositorio.unicamp.br Internet Source	<1 %
14	uklaptopreviews.com Internet Source	<1 %
15	origin.geeksforgeeks.org Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

REFERENCES

[1] I. N. Yulita, F. A. Hariz, I. Suryana, and A. S. Prabuwono, "Educational Innovation Faced with COVID-19: Deep Learning for Online Exam Cheating Detection," *Education Sciences*, vol. 13, no. 2, Art. no. 2, Feb. 2023, doi: 10.3390/educsci13020194.

[2] S. Kaddoura and A. Gumaei, "Towards effective and efficient online exam systems using deep learning-based cheating detection approach," *Intelligent Systems with Applications*, vol. 16, p. 200153, Nov. 2022, doi: 10.1016/j.iswa.2022.200153.

[3] M. Ghizlane, B. Hicham, and F. H. Reda, "A New Model of Automatic and Continuous Online Exam Monitoring," in *2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS)*, Dec. 2019, pp. 1–5. doi: 10.1109/SysCoBIoTS48768.2019.9028027.

[4] A. C. Ozgen, M. U. Öztürk, O. Torun, J. Yang, and M. Z. Alparslan, "Cheating Detection Pipeline for Online Interviews," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, Jun. 2021, pp. 1–4. doi: 10.1109/SIU53274.2021.9477950.

[5] L. C. O. Tiong and H. J. Lee, "E-cheating Prevention Measures: Detection of Cheating at Online Examinations Using Deep Learning Approach – A Case Study," *arXiv*, Jan. 24, 2021. doi: 10.48550/arXiv.2101.09841.

[6] A. Jadi, "New Detection Cheating Method of Online-Exams during COVID-19 Pandemic," *International Journal of Computer Science and Network Security*, vol. 21, no. 4, pp. 123–130, Apr. 2021, doi: 10.22937/IJCSNS.2021.21.4.17.

[7] N. Dilini, A. Senaratne, T. Yasarathna, N. Warnajith, and L. Seneviratne, "Cheating Detection in Browser-based Online Exams through Eye Gaze Tracking," in *2021 6th International Conference on Information Technology Research (ICITR)*, Dec. 2021, pp. 1–8. doi: 10.1109/ICITR54349.2021.9657277.

[8] A. Barrientos, M. Cuadros, J. Alba, and Á. S. Cruz, "Implementation of a remote system for the supervision of online exams through the use of cameras with artificial intelligence," in *2021 IEEE Engineering International Research Conference (EIRCON)*, Oct. 2021, pp. 1–4. doi: 10.1109/EIRCON52903.2021.9613352.

[9] M. Soltane and M. R. Laouar, "A Smart System to Detect Cheating in the Online Exam," in *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*, Dec. 2021, pp. 1–5. doi: 10.1109/ICISAT54145.2021.9678418.