Online Cheating Detection

Submitted by:
Simranjit Kaur (102216033)
Kriti Singh Behl(102216057)

BE Third Year

CSE/COE

Submitted to:

Dr. Anjula Mehto

Assistant Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

November 2024

TABLE OF CONTENT :

S. No	Topic	Page No.
1	Introduction or Project Overview	3
2	Problem Statement	4
3	Overview of the Dataset used	5
4	Project workflow	6 - 7
5	Results	8-13
6	Conclusion	14

Introduction or Project Overview

In the rapidly evolving digital age, online platforms for learning, meetings, and work have become integral to daily life. While convenient, these environments often suffer from a lack of direct supervision, making it difficult to ensure participants remain engaged and focused. Distractions such as looking away from the screen, using mobile phones, or showing signs of fatigue (e.g., covering eyes) are common and can significantly hinder productivity and learning outcomes.

This project aims to address this gap by developing a real-time detection system that monitors attentiveness through physical cues. Using advanced computer vision techniques, the system identifies behaviors such as:

* **Head movements**, indicating when a user shifts focus away from the screen.

* **Eye gaze direction**, detecting if the user is looking at the screen or elsewhere.

* **Hand covering the eyes**, signaling fatigue, stress, or disengagement.

* **Presence of mobile phones**, a common source of distraction.

The system's importance lies in its ability to provide immediate feedback or record inattentive behaviors for later review, offering solutions for environments where maintaining attention is crucial, such as:

* **E-learning platforms** to enhance student engagement.

* **Workplace monitoring** to ensure employee focus during virtual meetings.

* **Driver monitoring systems** for safety-critical applications.

This project integrates cutting-edge machine learning algorithms and efficient object detection models to ensure high accuracy and low latency, making it suitable for real-time deployment. By bridging the gap between online convenience and attentiveness, the system addresses a growing need in the modern digital landscape.

Problem Statement

With the rise of remote work, virtual meetings, and online learning, maintaining user attentiveness has become a critical challenge. In traditional in-person settings, facilitators can directly observe participants to gauge their engagement and address distractions. However, in online environments, distractions such as looking away from the screen, using mobile phones, or showing signs of fatigue often go unnoticed, leading to reduced productivity and engagement.

Key Issues:

Lack of Real-time Monitoring:
Online platforms lack effective tools to track user engagement based on physical behaviors, leaving distractions unchecked.

Subtle Disengagement Indicators:
Small yet significant behaviors like head tilts, gaze shifts, or hand gestures covering the eyes are often missed, but they can indicate loss of focus or interest.

Challenges of Object Distractions:
Common distractions, such as mobile phone usage, are not actively monitored in real-time, further contributing to inattentiveness.

Environmental Variability:
Variations in lighting, user positioning, and camera angles introduce complexities in accurately detecting these behaviors.

Without an effective monitoring solution, online platforms risk reduced effectiveness in education, productivity in professional meetings, and safety in applications like driver monitoring systems. This project addresses these challenges by providing a real-time, robust detection system capable of identifying head movements, gaze shifts, hand-over-eye gestures, and mobile phone presence, ensuring users remain attentive in online environments.

Overview of the Dataset used

The success of the online detection system depends on robust datasets that can train and evaluate the machine learning models for detecting behaviors like head movement, gaze direction, hand-over-eye gestures, and mobile phone presence. Here is an overview of the datasets utilized or considered for this project:

Dataset for Object (Mobile Phone) Detection :

* **Dataset Name:** Phone_detection_dataset

* **Description:** Custom dataset created by collecting images of phones in various settings for improved detection accuracy.

* **Size:** 8,416+ images.

The dataset includes the 3 sub folder; test ,train and valid containing the images and labels .

Key Features :

* High diversity in phone types, positions, and environments.

* Annotations for both handheld and stationary phones.

* Includes challenging scenarios such as partial occlusion or low lighting.

Applications in the Project :

Trains object detection models to recognize mobile phones as potential distractions in real-time.

Project Workflow

Initialization :

1. Import required libraries:
   - OpenCV (`cv2`)
   - MediaPipe (`mediapipe`)
   - NumPy (`numpy`)
   - PyAutoGUI (`pyautogui`)
   - Time (`time`)
   - Tkinter (`tkinter`)
   - YOLOv8 model

2. Define global parameters for optical flow, corner detection, and detection thresholds:
   - Optical flow (Lucas-Kanade algorithm).
   - Shi-Tomasi corner detection.
   - Thresholds for head movement, eye look-away, and hand proximity detection.

Detection Process :

Head Movement Detection

- Optical Flow (Lucas-Kanade)
- Track the motion of features (corners) between consecutive frames.
- Compute the average displacement of tracked points to estimate head movement.
- Trigger an alert if displacement exceeds the head movement threshold and cooldown period has elapsed.

Face and Eye Detection

- Use MediaPipe to detect facial landmarks.
- Extract coordinates for left and right eyes:
- Calculate the center of each eye based on the detected landmarks.
- Use these centers for subsequent checks.

Hand Detection

- Detect hand landmarks using MediaPipe.
- Calculate the Euclidean distance between each hand landmark and the centers of both eyes.
- Trigger an alert if a hand comes within the eye-covering threshold and the cooldown period has elapsed.

<u>Eye Look-Away Detection</u>

- If hands are not covering eyes:
- Use optical flow to track movement of eye landmarks between frames.
- Trigger an alert if the movement exceeds the eye look-away threshold and cooldown period has elapsed.

<u>Object (Mobile) Detection</u>

- detecting the mobile phone using the dataset involved through YOLOv8 model

Frame Updates

- Update previous frames and feature points to enable continuous tracking:
  - Store the current grayscale frame as the previous frame.
  - Use Shi-Tomasi corner detection to identify new feature points for tracking.

Alert Mechanism

- Use `pyautogui.alert` to notify the user of detected suspicious behavior.

Cleanup

- Release webcam resources and close all OpenCV windows when detection stops or the user exits the program.

User Interface (Tkinter GUI)

- Provide a user-friendly interface for interaction.
  1. Create a GUI window using Tkinter.
  2. Adding a title, instructions, and buttons
  3. Start the Tkinter main loop to keep the interface active.

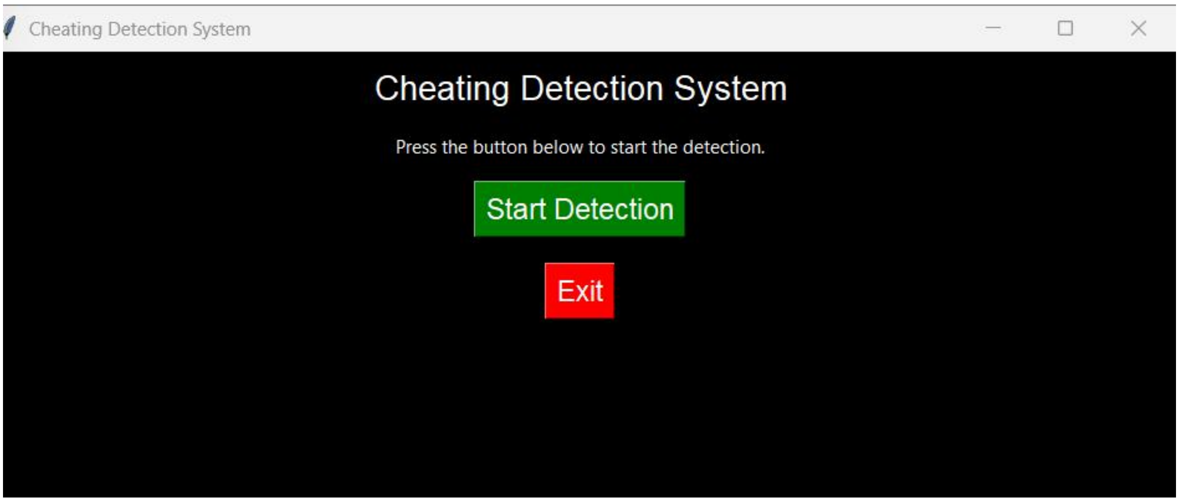
Execution

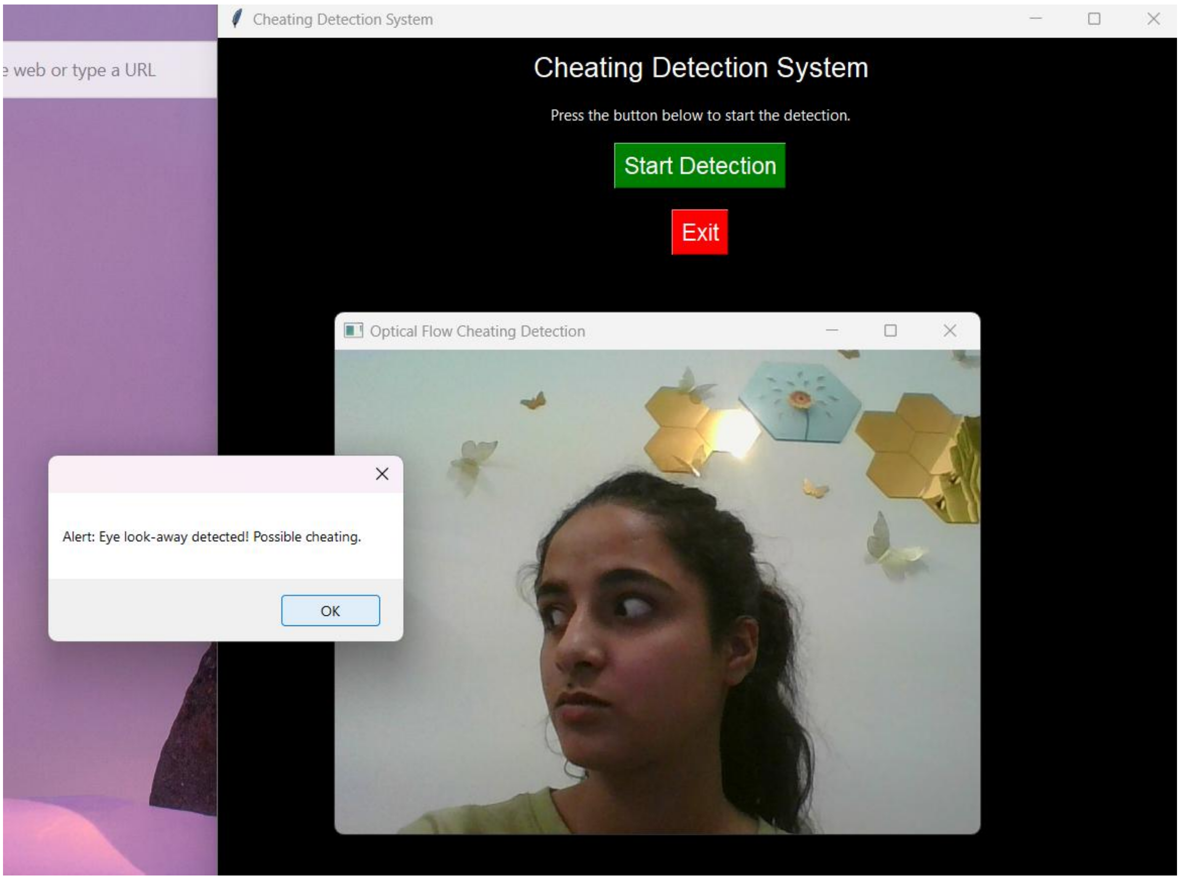
- Run the program by executing the main function:
  - Launch the GUI.
  - Start video capture and detection upon user input.

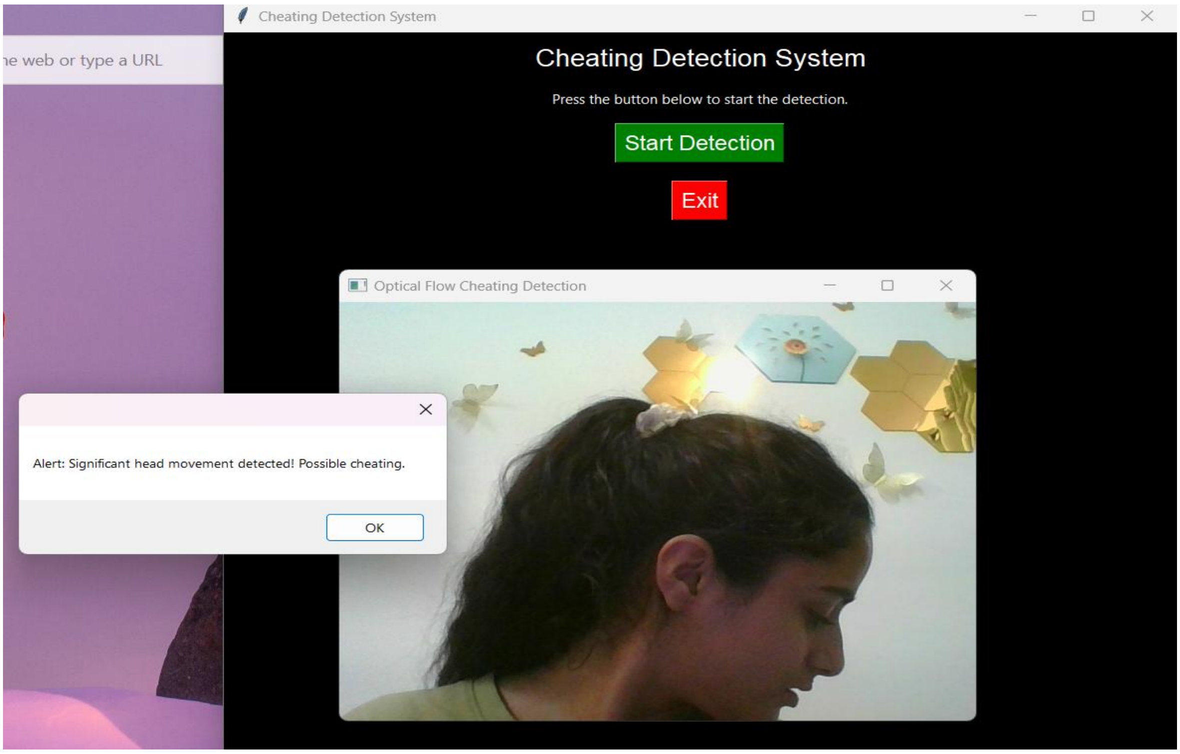
This workflow ensures continuous monitoring and real-time detection of specified behaviors with clear user interaction and system feedback.

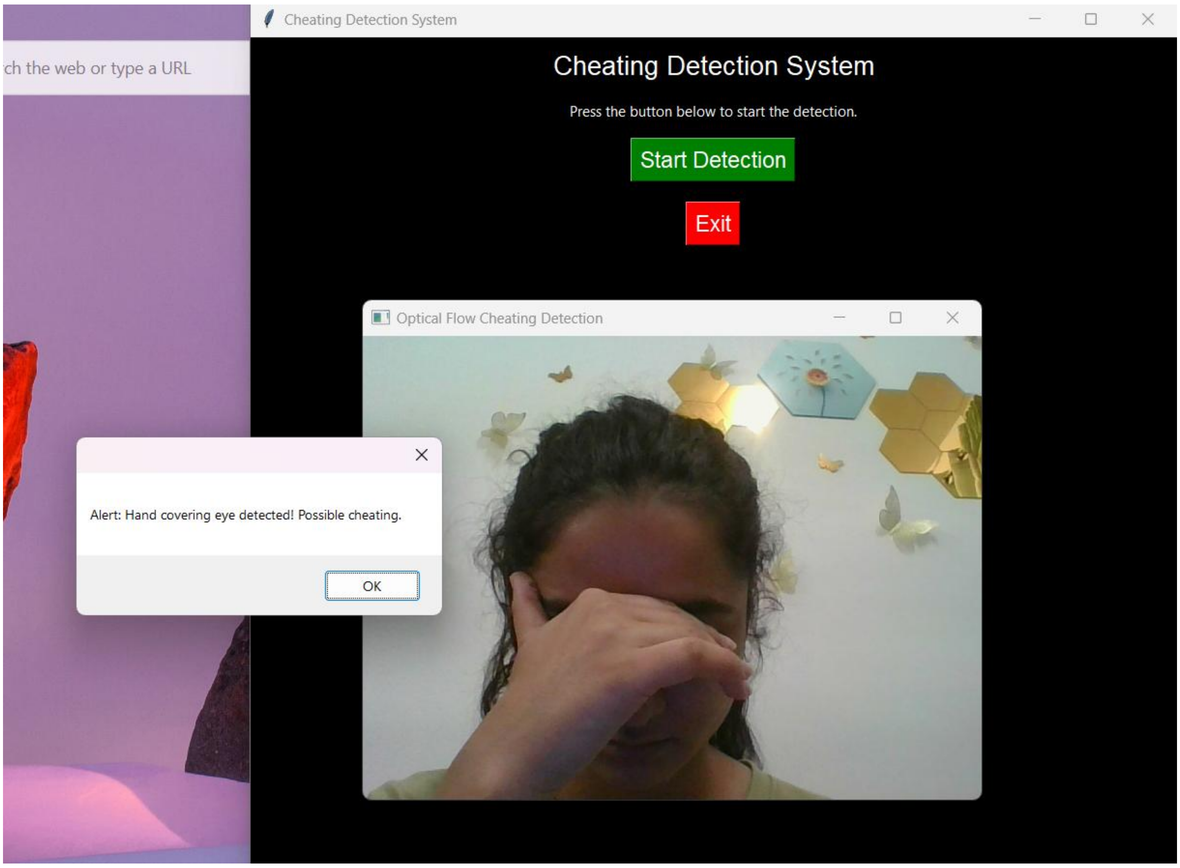
Results

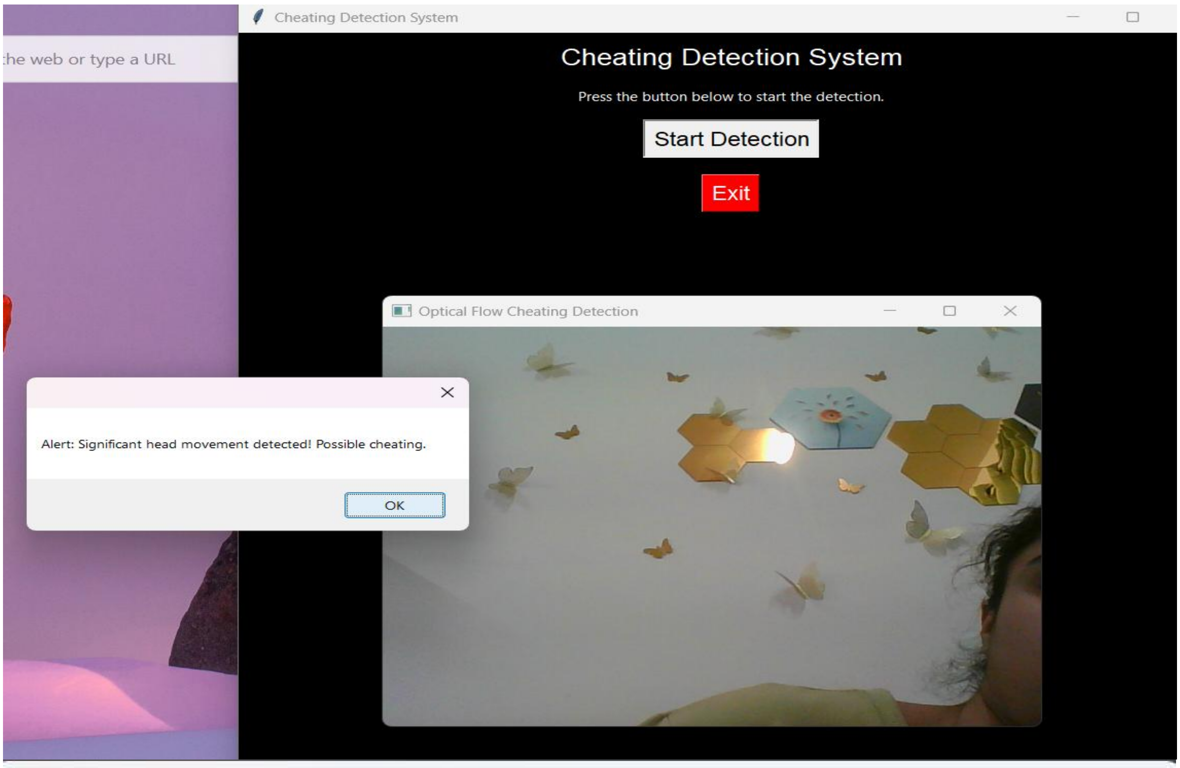
The following screenshots clearly gives us the result of the project :

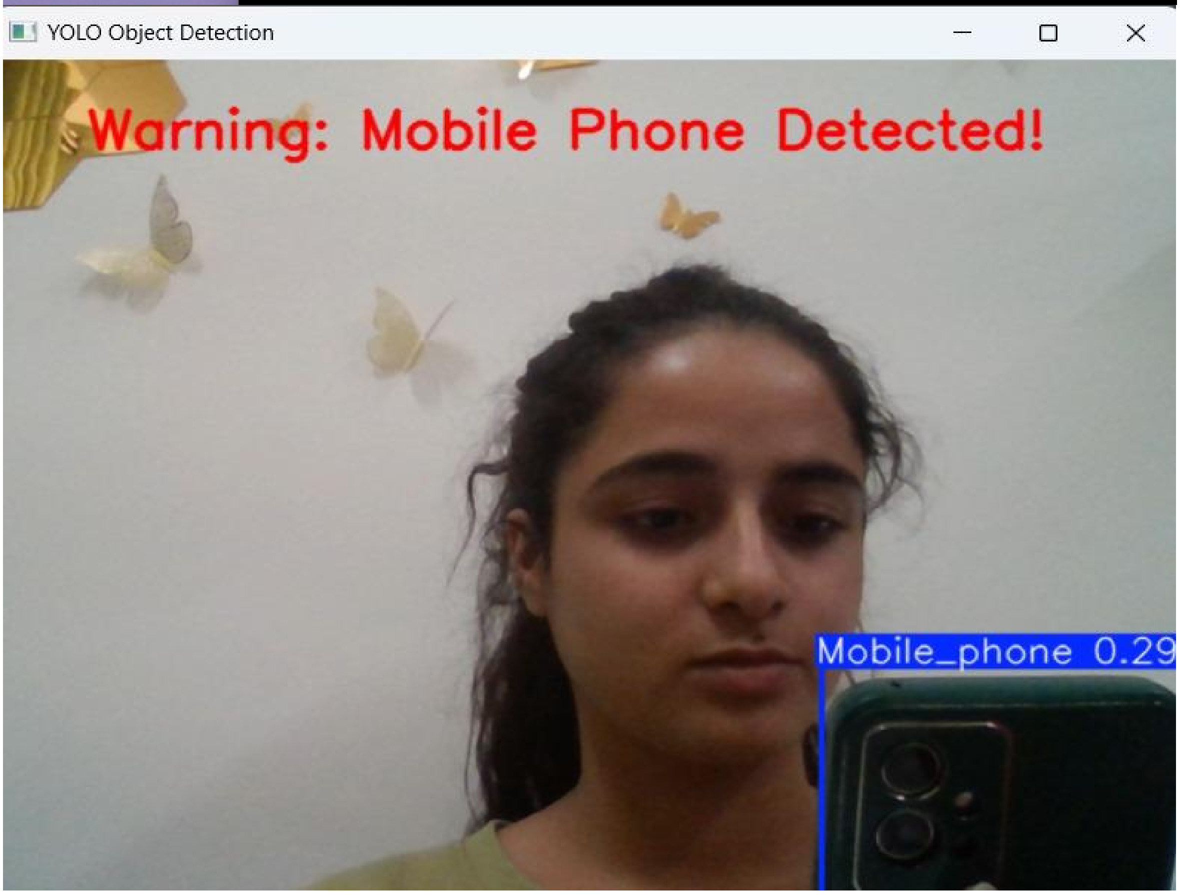


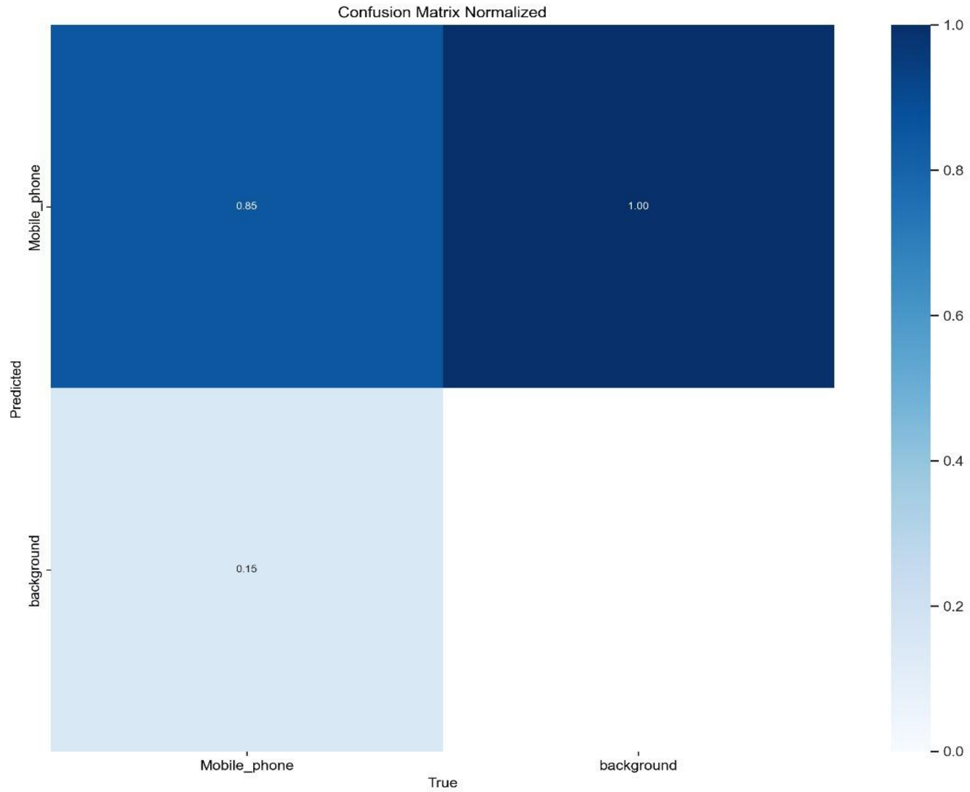


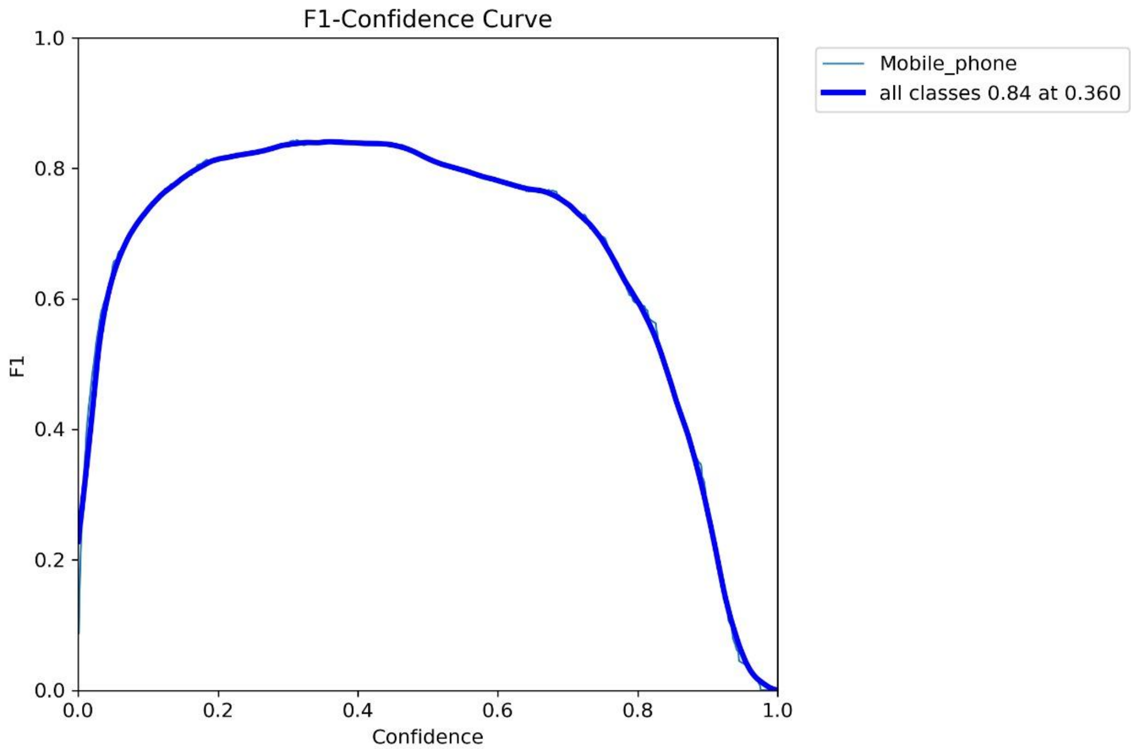

Cheating Detection System
Cheating Detection System
Press the button below to start the detection.
Start Detection
Exit
Optical Flow Cheating Detection
Alert: Significant head movement detected! Possible cheating.
OK

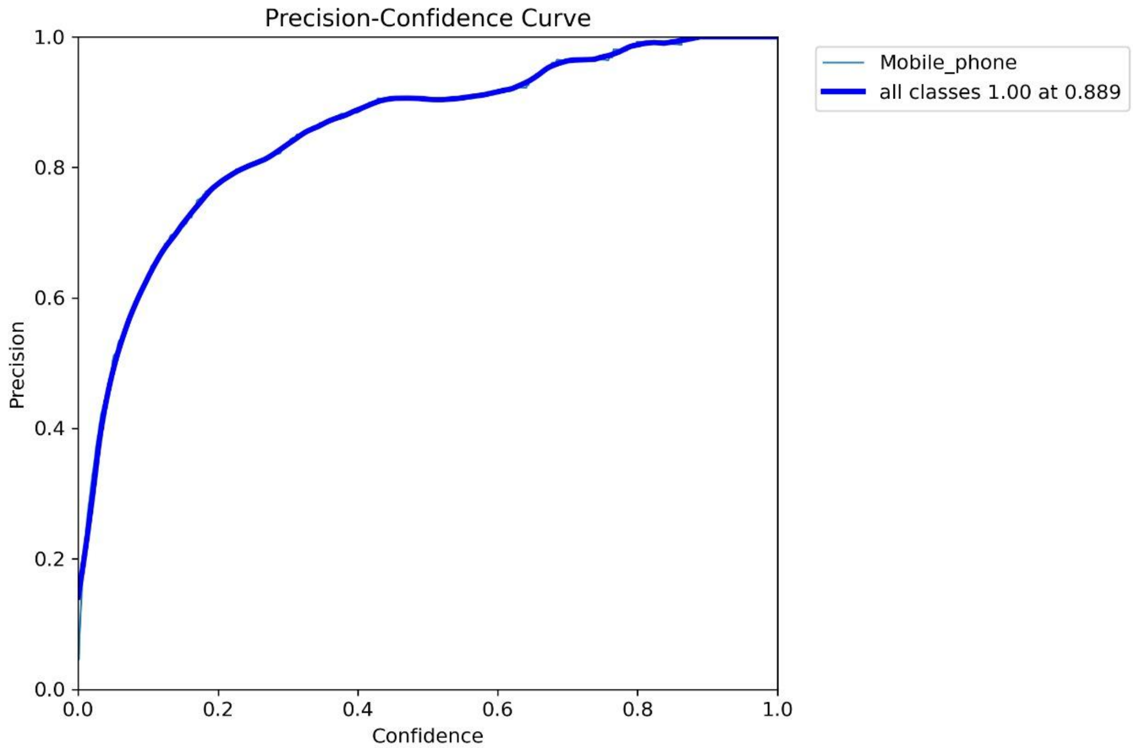

Cheating Detection System
Cheating Detection System
Press the button below to start the detection.
Start Detection
Exit
Optical Flow Cheating Detection
Alert: Hand covering eye detected! Possible cheating.
OK

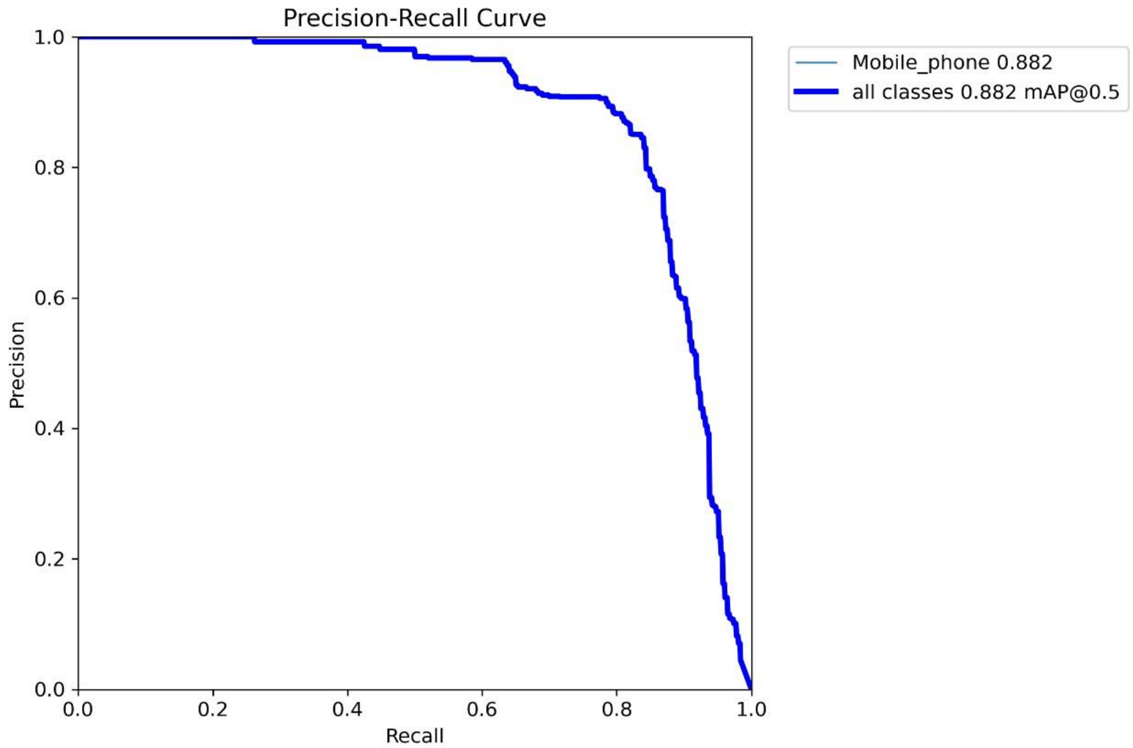

Cheating Detection System
he web or type a URL
Cheating Detection System
Press the button below to start the detection.
Start Detection
Exit
Optical Flow Cheating Detection
Alert: Significant head movement detected! Possible cheating.
OK

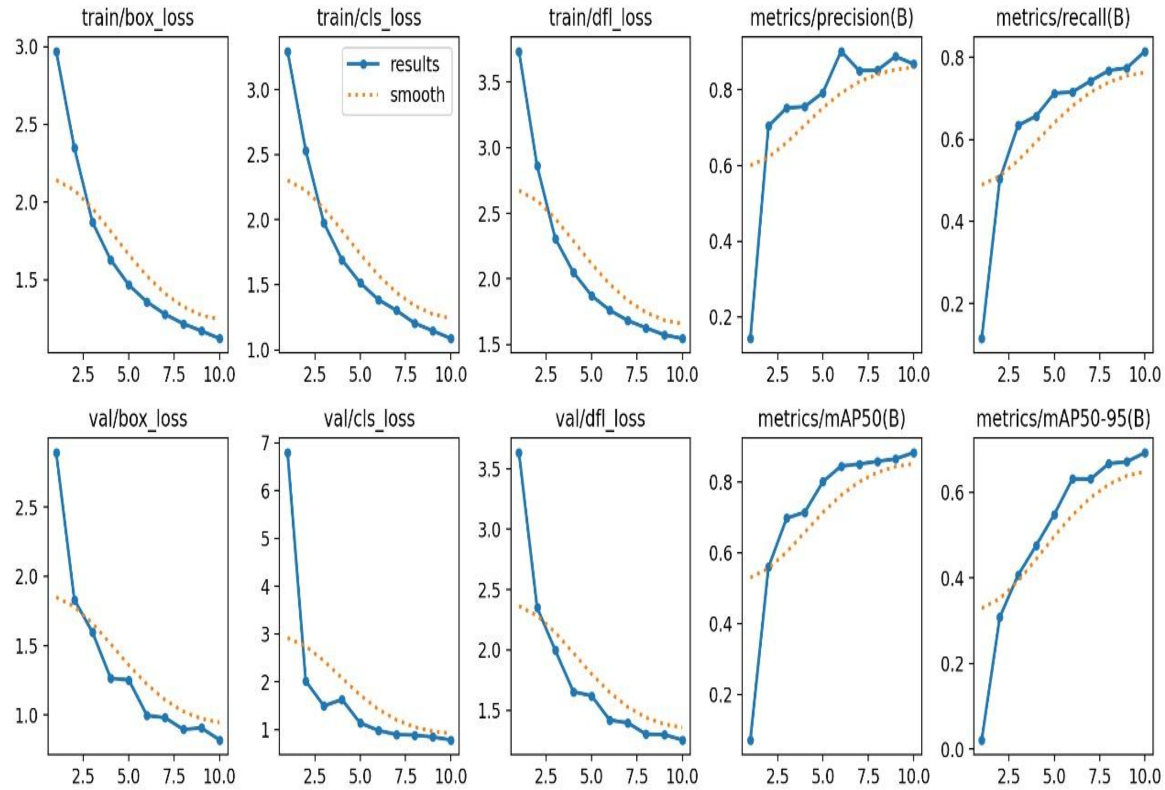

YOLO Object Detection
Warning: Mobile Phone Detected!
Mobile_phone 0.29


Confusion Matrix Normalized
Predicted
Mobile_phone
background
0.85
1.00
0.15
True
Mobile_phone
background
1.0
0.8
0.6
0.4
0.2
0.0


F1-Confidence Curve
Mobile_phone
all classes 0.84 at 0.360
F1
Confidence
0.0
0.2
0.4
0.6
0.8
1.0


Precision-Confidence Curve
Mobile_phone
all classes 1.00 at 0.889
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0


Precision-Recall Curve
Mobile_phone 0.882
all classes 0.882 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


train/box_loss
train/cls_loss
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
results
smooth
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

Conclusion

The online detection system successfully monitors attentiveness by detecting head movements, gaze shifts, hand interactions, and mobile phones. It demonstrates high accuracy across varied scenarios and provides real-time feedback to mitigate inattentiveness.

Future Scope:

- Incorporating emotion recognition for deeper insights into engagement.
- Expanding detection to multi-user environments.
- Deploying on edge devices for wider accessibility.

This system addresses a critical gap in online environments by ensuring active participation and minimizing distraction.