جامعة خليفة
Khalifa University

Computer Science Department

COSC 497 | Senior Design Project II

Spring 2025

Final report

Project Title:

"Classroom Occupancy Monitoring and Behavior Analysis System"

Submitted to:

Dr. Naoufel Werghi

Dr. Muzammal Nasser

Dr. Maregu Asefa

Examiners: Dr. Panos Liatsis Dr. Sajid Javed

Submitted by:

Alanood Alharmoodi	**100053854**
Rodha Alhosani	**100058376**
Shaikha Alhammadi	**100058710**
Yasmine Benkhelifa	**100059531**

Table of Contents

Abstract

This report goes over the beginning phases of implementing an AI-driven Classroom Occupancy and Behavior Analysis system designed to enhance safety and optimize the learning experience and resource utilization in educational institutes. The system integrates object detection and behavior analysis using advanced computer vision techniques. The utilization of YOLOv8s was a key component in providing high-speed and accurate detections of students, illegal objects, and defining illegal areas. In the first phase of the project the system is able to detect illegal objects such as cups, mug, bottles in the laboratories and alert unauthorized access in restricted areas. In the second phase, the system will be able to identify disruptive behaviors such as eating, drinking, phone calls, texting, cheating as well as accurately detect occupancy levels. Data was collected from CCTV cameras, annotated using MakeSense, and trained using diverse datasets using techniques such as augmentation. A user-friendly dashboard visualizes real-time alerts, logs events, generates email alerts, and provides CSV logs with coordinates of bounding boxes. The system went through an intensive validation process, through metrics such as the mean Average Precision (mAP), precision, recall, and more. The innovative solution presented by the system reduces dependency on manual monitoring, minimizes errors, and aids in providing a more productive learning environment.

Keywords: AI-driven system, classroom monitoring, behavior analysis, YOLOv8s, object detection, computer vision, educational safety, illegal area detection, illegal object detection, machine learning.

Chapter 1: Introduction

1.1 Background and Problem Statement

Educational institutes around the world rely on maintaining an environment in classrooms that aid in nurturing future leaders. Optimal classroom settings are crucial for providing students and instructors with a safe and effective learning experience. However, challenges arise with issues such as overcrowding and disruptive student behaviors. Actions such as bringing food or beverages pose risks to costly equipment. Moreover, behaviors like cheating, unauthorized phone usage, texting, and entering restricted areas without proper safety equipment (e.g safety jackets and helmets) further complicate classroom management. Relying on traditional methods to monitor classroom occupancy and analyze behavior is time consuming, prone to human error, and inconsistent. These limitations call for a more modern and innovative approach, which makes use of CCTV cameras, automation, and artificial intelligence.

1.2 Goals and Objectives

The primary goal of this project is to develop a sophisticated AI- driven Classroom Occupancy and Behavior Analysis System. The team is focused on implementing an innovative solution to an ongoing problem; to enhance classroom management whilst providing a safe and productive learning environment. By adopting an automated approach, educational institutions will gain valuable insights into classroom occupancy and student behaviors, leading to improved educational management practices. The key objectives include:

1. Occupancy Monitoring: Automatically monitor and detect the number of students present in each classroom to optimize space and resource utilization.
2. Behavior Analysis: Analyze the behavior of students to recognize and notify signs of distress, disruptive behavior, or partaking in prohibited activities such as eating, drinking, cheating, texting, or phone calls. Additionally, automated emails are triggered for incidents deemed as critical.
3. Object Detection: Detect and alert staff about the presence of unauthorized objects in the classroom with their specific x and y coordinates, such as bottles, mugs, and phones. Additionally, detect safety equipment such as safety jackets and helmets to ensure staff and students alike are following safety regulations.
4. Illegal Area Detection: Detect and notify if an unauthorized individual enters areas classified as restricted, checking specifically for safety jackets and helmets, ensuring compliance with safety policies and classroom regulations.
5. Safety Enhancement: Enhance overall classroom safety by promptly identifying unauthorized access and unusual behaviors, ensuring rapid and appropriate intervention.
6. User Interface: Provide an intuitive and user-friendly interface for educators, offering valuable tools such as real-time classroom monitoring, data visualization, immediate alerts and notifications, which includes automated email alerts for critical incidents.

1.3 Motivation

The motivation behind this project stems from the increasing need for efficient and modern ways to manage classrooms in educational institutes such as universities. The team has recognized the shortcomings of traditional monitoring methods that are inadequate especially in laboratories, where inconsistencies and failure to identify certain behaviors or objects can turn into costly or potentially dangerous situations. Factors such as large classroom sizes, the need for real-time insights, immediate detection and intervention highlight the need for an AI-driven system. Key motivations include:

1. Enhancing Learning Outcomes: Individual and group learning is improved when student engagement is monitored and behavioral issues are identified and improved early on. Traditional monitoring methods often fail to capture behaviors immediately and effectively. Behavior analysis techniques are critical in maintaining classroom order and providing a safe environment [1].

2. Ensuring Safety: Automating the detection of abnormal behaviors, unauthorized access to restricted areas, and the presence of prohibited objects significantly enhances classroom safety. In the second phase of the project, safety was further strengthened with email alerts for critical behaviors.
3. Reducing Manual Labor: Reducing the dependency on human observations will allow instructors to place their efforts in lecturing. This also reduces the risk of errors and missed detections.

1.4 Methodology

The project follows a structured and systematic methodology approach to ensure that the system is successfully deployed and achieves all of the target goals and objectives. This chapter focuses on giving a brief rundown on the steps the team took, whereas the specifics and details of implementation will be clarified in the upcoming chapters of the report.

1.4.1 System Design and Requirement Analysis

The initial phase included gathering requirements and defining key components to ensure the success of the system. This includes determining which specific behaviors and areas to deem as illegal, and which objects to flag when found present in a classroom. The system architecture was also briefly discussed as well as a high-level overview of the project to define the video processing pipeline.

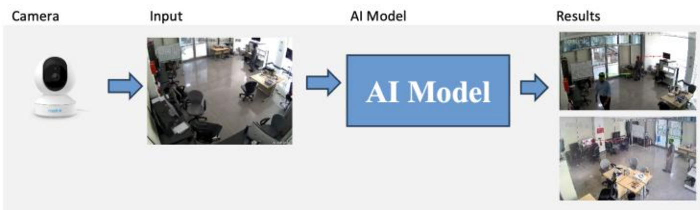


Figure 1 A High-Level Overview Taken from the Project Proposal

1.4.2 Data Collection and Preparation

The system will utilize the existing CCTV cameras that are already installed in classrooms in order to collect video data. The collected footage includes various scenarios ranging from simple to complex situations to reflect real-life situations. Simpler scenarios include illegal items placed on desks that are in clear sight, which makes it easy to flag and detect. More complex scenarios include illegal objects that are further away from the camera and behind or between different objects which make flagging more difficult.

In Phase 2, the dataset was expanded to include advanced behavioral classes such as eating, drinking, texting, phone calls, and different cheating contexts. It also includes proper versus improper safety compliance situations, such as entering the restricted area with or without the required helmet and safety jacket. The data collected reflects both regular classroom conditions and exam settings, where specific behaviors such as phone usage are reclassified as cheating.

Once collected, the video footage is preprocessed using techniques such as frame extraction, resizing, annotation, and noise reduction to prepare the input for model training. This process will be defined and explained in further detail in the subsequent chapters.

Figure 2 A Visual of Complex Vs. Simple Situations- The Mugs in Clear View is Easy to Flag Whereas the Black Mug Behind the Black Bag is Difficult to Detect.

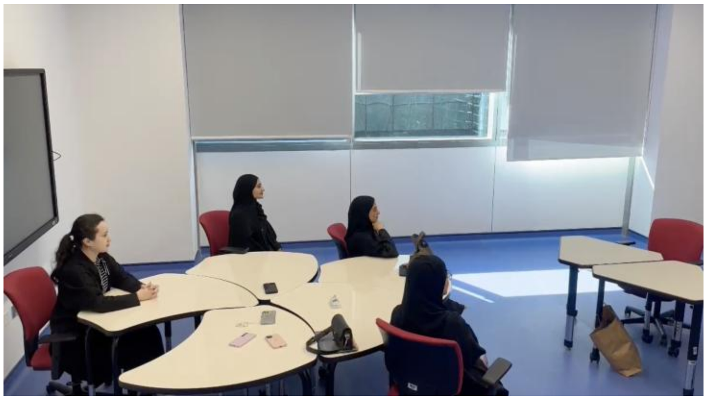


Figure 3 A Visual of Simple Scenario - Phones in Clear Sight - In a Classroom Setting

1.4.3 Data Annotation

The data annotation phase includes annotating the video data. Frames from the video were used to label and identify illegal objects and areas. The frames were obtained from the training dataset, and were manually annotated using the online website MakeSense. Rectangular shapes were utilized to label illegal objects and areas in the frame. Finally, the annotated images were exported as text files in YOLO format, which generated separate label files for each image. The same process was repeated for validation and testing images. These annotations will serve as ground truth labels for training and evaluating the machine learning model.

1.4.4 Model Selection and Development

The team will make use of object detection algorithms, such as YOLO (You Only Look Once) for real-time detection of illegal objects and areas. While selecting models, factors such as accuracy, efficiency, and real-time performance were taken into consideration. Initially, the team experimented with YOLOv5, where Figure 3 below shows what objects were able to be detected before any training, fine-tuning, or specifications on which objects to flag. As the project progressed, the team recognized the need for an upgraded model to simplify training. Factors such as the package size of each YOLO model, accuracy, and precision were taken into consideration when deciding on which version to upgrade to. The team then decided on implementing YOLOv8s which provided a good trade-off between model performance and package sizes. The model selection, made in Phase 1, continues to serve as the foundation in Phase 2 due to its reliable performance, fast inference speed, and compatibility with the expanded detection requirements introduced in the second phase.

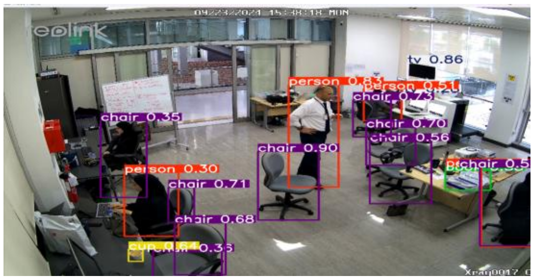


Figure 4 YOLOv5 Detection Prior to any Training or Specifications

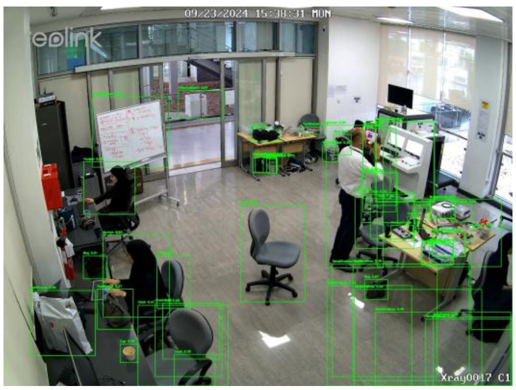


Figure 5 YOLOv8s Detection Capabilities After Minimal Training

1.4.5 Model Training and Evaluation

The annotated dataset is divided into three subsets which include training, testing, and validation. When inputting the training dataset into the model, supervised learning techniques were used. In Phase 2, the dataset was expanded to include additional behavioral classes, object classes, safety compliance situations, and context-specific actions. As a result, multiple YOLOv8s models were trained or fine-tuned for specific tasks. To assess the performance of these models when it comes to detecting illegal objects behaviors, and restricted area violations, various evaluation metrics were implemented. These include F1-score, precision, recall, and accuracy. The results were visualized using confusion matrices to better understand model effectiveness and potential areas for improvement.

1.4.6 Integrating Computer Vision Techniques

To enhance the detection capabilities of our system, pre-existing computer vision techniques were integrated. This includes making use of algorithms such as object classification and tracking. The purpose of doing this is to ensure our system is functional even in dynamic environments where students are constantly moving around. These techniques became even more essential in Phase 2 with the introduction of behavior detection and safety compliance monitoring. The use of tracking supports consistent identification across frames and helps reduce false positives in both live CCTV deeds and recorded video analysis.

1.4.7 System Development and Integration

The next phase involves integrating videos from the CCTV cameras that are installed in classrooms. The system supports both live video streams and pre-recorded footage, providing flexibility for real-time monitoring and offline analysis. The pipeline processes the incoming video, applies detection models, and sends the results to a centralized monitoring interface. Detected violations and objects are logged, and in critical cases, an email is automatically triggered. This integration ensures the system operations seamlessly in various classroom scenarios including exam-mode proctoring and standard monitoring.

1.4.8 Real-Time Monitoring

A user-friendly interface is included in the system, allowing educators to visualize the output in real time. The dashboard displays detections related to student behaviors, object presence, safety compliance, and restricted area violations. The system is capable of supporting both standard and exam-specific monitoring modes. The interface also includes an automated alert system for critical incidents, and logs all detections into a structured Excel report for future review and documentation. Key design principles such as ease of use, reliability, clarity, and responsiveness were prioritized to ensure the interface remains intuitive and effective.

1.4.9 Documentation and Deployment

The final phase includes documenting the entire project lifecycle, including data collection, preprocessing, model training, multi-model integration, and system development. This documentation captures the reasoning behind design decisions such as selecting YOLOv8s, supporting both live and recorded videos, and building distinct interfaces for standard monitoring and exam proctoring. Finally, the team aims to create a comprehensive user guide and technical documents to assist target users and provide a smooth deployment of the system in educational institutes.

1.5 Overview of the Technical Area

1.5.1 Introduction to Classroom Monitoring Systems

In educational environments it is essential to harbor a safe and structured setting; which is why monitoring student behaviors, classroom occupancy, and objects brought into a classroom is of extreme importance. Manual observations of such behaviors is an inconvenience, and takes valuable time away from the learning process [2]. It becomes especially impractical when certain factors arise, such as large numbers of students, or classrooms with certain restrictions such as illegal areas one should not get close to. With the advancement of technology, there has been a clear shift towards a more innovative solution for such problems, the deployment of automated systems. Incorporating techniques such as computer vision, artificial intelligence, data analysis, and machine learning will allow real-time insights of student behaviors which can be further analyzed by educators [2].

1.5.2 The Current State of Classroom Monitoring Systems

The current state of classroom monitoring systems includes deploying techniques such as manual supervision and basic CCTV cameras. However, there are several limitations found in these methods. In the case of manual supervision, there is a heavy dependency on human observation. Not only is this impractical in big classrooms with a large number of students, it is also inconsistent and prone to error. This is because humans are bound to miss certain disruptive behaviors, and cannot detect every object that is brought into the classroom. Additionally, it is labor intensive and is taking away from time that can be spent on things more beneficial to both the students and instructors.

In the case of basic surveillance systems such as standard CCTV cameras, it is beneficial in terms of security but it does not provide aid to educators and educational institutes. The footage is captured, but there is no real-time insights, object detection, or area detection. Additionally it cannot analyze classroom occupancy or flag disruptive behaviors. Even with more innovative approaches in the application of CCTV cameras in classrooms, such as the integration of PIR sensors for proactive surveillance [3], the activities still need to be monitored and analyzed manually. For those reasons, there is a growing need for advanced, AI-driven solutions that provide consistent, beneficial analytics beyond capturing videos and manually monitoring students.

1.5.3 Evolution of Automated Behavior Analysis in Classrooms

With the rise of computer vision techniques and machine learning, there has been a significant advancement in the field of automated behavior analysis and object detection. Earlier attempts in the field of simple motion detection involved techniques such as supervised background subtraction. This method makes use of neural networks and supervised learning to improve detection accuracy. However, it relies on objects, so with a growing number of features the complexity significantly increases [4]. Issues will arise from such methods in classrooms where there are a large amount of objects and bodies overlapping.

Currently, rapid developments in deep learning aid in the performance of automated behavior analysis systems. The following techniques play a crucial role in the improvements of this field:

1. Object Detection with Deep Learning: As the team is implementing YOLOv8s, the contributions of YOLO in the object detection and behavior analysis field are of utmost importance. YOLO has the ability to accurately identify numerous objects in one pass. Additionally, the fact that models are pre-trained and can be fine-tuned to specific datasets make YOLO an attractive solution for object detection. The YOLOv8 algorithm is split into three crucial parts: the backbone, neck, and head. This can be visualized with the help of Figure 1 below. The backbone is responsible for feature extraction from the provided input. The neck is responsible for collecting the mappings of features from the backbone. Finally, the head is responsible for the output which includes confusion matrix, confidence scores, and bounding boxes for the detected objects [6].

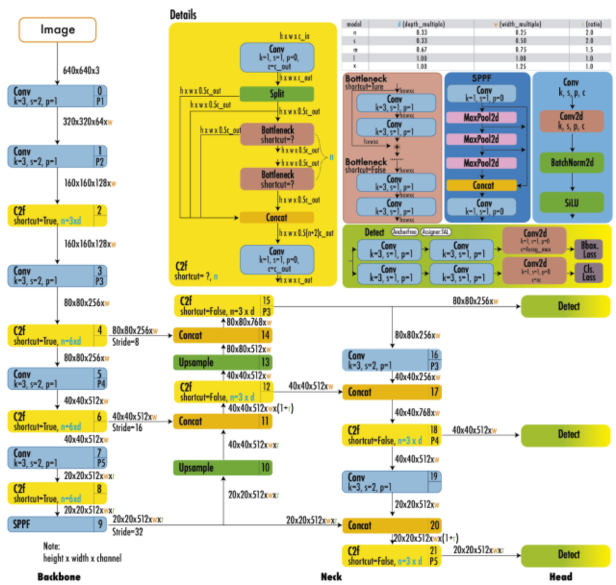


Figure 6 YOLOv8 Architecture [5]

2. Multi-Camera Integration: By integrating footage from various angles into automated behavior systems, deep learning models are able to continuously improve accuracy.

1.5.4 Challenges in Classroom Monitoring

The section above highlights advancements in the field of computer vision and automated systems, however despite these improvements, challenges still exist when it comes to implementing a classroom occupancy and behavior system. These include:

1. Overlapping Bodies: In classrooms, students can overlap with one another or obstruct the visibility of illegal objects. This makes it difficult for a system to capture all illegal objects and behaviors. Additionally, detecting who enters an illegal area becomes more difficult the more crowded a classroom becomes. Another challenge is to identify individuals performing subtle actions like glancing at another students' exam paper (cheating) during assessments.
2. Varied Lighting Conditions: When the lighting in a classroom is too bright or too dim, it could potentially lead to a negative impact in the performance of a system using computer vision. This requires an intense preprocessing and training phase to prepare a system for such conditions, which could be challenging.
3. Behavior Variability: Detecting illegal behaviors such as eating, drinking, or other disruptive actions in a classroom can vary from student to student, and is context-dependent. The challenge is familiarizing a system with numerous cases of illegal behaviors and ensuring the system can distinguish between when a behavior is illegal and when it is tolerable.
4. Object Variability: Illegal objects such as mugs, water bottles, and food items come in many different shapes and sizes. The system must have previous knowledge of many different food and drink items which can lead to large datasets, which in turn increases difficulty.
5. Live vs. Offline Processing: Ensuring consistent accuracy between live CCTV feeds and pre-recorded footage includes technical hurdles such as frame rate variation, buffering, and resource constraints. These challenges must be handled during deployment.

1.5.5 Relevant Literature Review

After conducting an intensive literature review on computer vision systems present in educational settings, several key findings were uncovered which include:

1. Attendance Monitoring: New approaches in attendance monitoring that implement face detection algorithms integrated with Learning Management Systems (LMS) are being explored [7]. The objective is to automatically detect and report the attendance of students present in a lecture.
2. Behavior Analysis: Methods are being tested and explored to detect student engagement in classrooms based on factors such as body posture and facial expressions [8]. The objective of such a system is to automatically classify whether students are engaged or not during lectures to aid in the learning process.

3. Occupancy Detection: Implementing YOLO models to count and keep track of occupancy levels inside of a classroom is another way computer vision is implemented in educational settings. The benefits of deploying this in a classroom include improving energy efficiency and adhering to safety protocols [9].

1.5.6 Patent Search

Searching for relevant patents to discover AI-driven technologies targeting behavior analysis and occupancy allow for the identification of unique contributions in this field. It provides a better understanding of existing solutions whilst highlighting what is missing from the market. The research conducted revealed patents related to surveillance systems with behavior analysis, automated attendance systems, and engagement monitoring. However, the methods mentioned do not provide insight into illegal objects and areas, disruptive behaviors, and are not specifically tailored to educational settings. After analyzing existing patents, it was evident that there was a need for a more tailored solution.

1.5.7 Identified Research Gaps

There were several gaps identified despite the advancements in computer vision technologies and automated systems. The first gap noticed is that there is a lack of context-specific solutions, meaning although surveillance systems exist to monitor occupancy they are not catered to classrooms. This could lead to lower detection and accuracy rates since the setting of a classroom is unique and varies greatly from other environments. Additionally, although behavior analysis exists it is mostly aimed at detecting engagement levels by posture and facial expressions. This lacks the detection of disruptive behavior, and activities that are not permissible in a classroom such as eating or drinking. Lastly, the team did not come across a system tailored for detecting whether someone was in an area deemed illegal in a classroom. The lack of this solution is concerning especially in laboratories where entering certain areas could be dangerous or lead to catastrophic outcomes.

1.6 Overview of the Report

The report systematically breaks down the project into key components and tasks, each chapter aligns with the overall goal of creating an AI-driven automated system to monitor classrooms in educational institutes to improve the learning experience.
Chapter 1 presents a detailed overview of the development of an advanced system for classroom occupancy and behavior analysis using computer vision and machine learning techniques. A clear outline of the objective of the project along with the goals placed is also found in this chapter. Additionally, the key roles of existing techniques such as YOLO are emphasized. Moreover, the methodology and technical overview including the steps taken to successfully create the automated system. Chapter 2 provides a detailed overview of the system architecture and specifications. It

outlines both functional and non-functional requirements and presents the hardware and software configurations that support both live and offline CCTV analysis. Chapter 3 dives into dataset construction, highlighting the configuration and limitations of the data, the process of data acquisition from CCTV sources, incorporation of Roboflow datasets, and the inclusion of object-specific examples. Techniques such as data augmentation, labeling, and handling overfitting/underfitting are also discussed in depth. Chapter 4 walks through the step-by-step development process, including problem definition, model selection, and training workflow. This chapter also explains how iterative refinement and validation processes were used to improve the performance of our models. This leads to Chapter 5, where concept generation and design evaluation are introduced. Here, the decisions made throughout system development and interface design are highlighted. Additionally, this chapter explores how different configurations (normal mode vs. exam mode) were incorporated to adapt to varying classroom use cases. Chapter 6 focuses on validation, presenting detailed evaluation metrics such as accuracy, precision, recall, and F1-score. Results from training and testing phases are analyzed, along with failure cases that highlight challenges like occlusion and behavior variability. Chapter 7 concludes the report with a summary of work completed, the challenges faced, project constraints, IEEE standards, and an outline of next steps for future improvements.

Chapter 2: System Overview

2.1 Requirements

The goal of the Classroom Occupancy and Behavior Analysis system is to create an advanced AI-driven system that surpasses traditional observation methods and reduces the need of manual monitoring. To achieve this, the system is designed based on a set of functional and non-functional requirements that guide the development processes and serves as a foundational blueprint for implementation.

2.1.1 Functional Requirements

1. Illegal Object Detection: The system must detect illegal objects in the classroom such as mugs, bottles, and phones, and clearly identify their locations using x and y coordinates.
2. Illegal Area Detection: The system must be able to define restricted area zones within the classroom and trigger an alert when an individual enters these areas without proper safety gear.
3. Event Logging and Reporting: All detected objects and events must be logged and flagged through a user-friendly interface. The system should also generate Excel logs and send automated email alerts for critical events.
4. Occupancy Monitoring: Accurately detect and count the number of students in a classroom using footage from CCTV cameras.

5. Behavior Analysis: The system must identify disruptive behaviors and flag them, this includes eating or drinking in the classroom. After identifying such behaviors, an alert should be triggered.
6. Mode Switching (Normal vs. Exam Mode): The system must support two distinct operational modes. In normal mode, only direct cheating behavior is flagged (looking at neighboring students). In exam mode, additional behaviors such as texting and phone calls are also classified as cheating.
7. Live and Offline Video Support: The system must support both real-time monitoring from live CCTV camera feeds and offline analysis of recorded footage.
8. GUI Visualization: The system must display all detections in real time through a user interface, including bounding boxes, labels, and status logs.

2.1.2 Non-Functional Requirements

1. Accuracy: The team aims to achieve high accuracy in student, object, and behavior detection. Part of accuracy is ensuring consistent performances across various classroom setups and environments which is a crucial part in successfully deploying the system.
2. Latency: Having a low latency is important in terms of real-time processing. The team plans to ensure low latency between video input processing and alert generation in the user interface, especially in time-sensitive scenarios such as exams or safety breaches.
3. Scalability: The team aims to ensure the system can handle classrooms of all sizes, with minimal to no degradation in performance even when the number of students increases.
4. Robustness: The system should operate effectively with classrooms of different layouts and under various lighting conditions. It should also be able to perform just as effectively even when camera angles change.
5. Usability: An intuitive and friendly user interface should be provided.
6. Efficiency: Video streams should be processed efficiently without using excessive computational resources such as storage.
7. Generalizability: The automated system should be compatible with desktops and laptops of different models.
8. Privacy and Ethics: Ensure that the system collects data and recordings in a way that complies with legal and ethical standards.
9. Cost Effectiveness: In order to ensure that various educational institutes with all types of budgets can gain the benefits of implementing our system, the team aims to balance functionality with affordability.
10. Documentation and Training: When the system will eventually be ready for deployment, user manuals and technical guides must be created to facilitate system adoption.

	Accuracy	Latency	Scalability	Robustness	Usability	Efficiency	Generalizability	Privacy	Cost	Documentation	Score
Accuracy	0	1	1	1	1	1	1	0	1	1	8
Latency	0	0	0	0	0	0	1	0	1	1	3
Scalability	0	1	0	0	0	1	1	0	1	1	5
Robustness	0	1	1	0	1	1	1	0	1	1	7
Usability	0	1	1	0	0	1	1	0	1	1	6
Efficiency	0	1	0	0	0	0	1	0	1	1	4
Generalizability	0	0	0	0	0	0	0	0	1	1	2
Privacy	0	1	1	1	1	1	1	0	1	1	9
Cost	0	0	0	0	0	0	0	0	0	0	0
Documentation	0	0	0	0	0	0	0	0	1	0	1

Table 1 PCC Table for Non-Functional Requirements

2.2 System Specification

2.2.1 System Architecture

The system follows a modular architecture with the following key components:

1. Input Layer: The system accepts video streams from existing CCTV cameras installed in classrooms. It supports both live feeds and pre-recorded video footage, providing flexibility for real-time and offline monitoring.
2. Processing Layer: The input video is first preprocessed using OpenCV to enhance frame clarity, reduce noise, and normalize the format. This layer then routes the frames through three specialized YOLOv8s models, the first model detected in the first phase of our project, and the second model is used to identify behaviors such as cheating and safety equipment.
3. Output Layer: When violations, events, or objects are detected, they are displayed in a real-time user interface with labeled bounding boxes, confidence scores, and positional data. The system also triggers automated email alerts for critical behaviors and generates an Excel-based detection log for reporting and review. Additionally, the interface supports switching between normal and exam modes.

2.2.2 Hardware Specification

The CCTV cameras should have a minimum resolution of 1080p to ensure high quality video outputs that are clear, in order for our models to accurately process the input. Additionally, sufficient storage space is required in order to flag, download, and look back at previously processed videos.

2.2.3 Software Specification

The system is built using a combination of machine learning frameworks, computer vision techniques, and user interface tools to support object detection, behavior analysis, and real-time event monitoring. YOLOv8s is used as the core detection model, enabling the identification of students, behaviors, and illegal objects after an intensive training and fine-tuning process. In Phase

2, the system was expanded to incorporate behavioral detections such as eating, drinking, cheating, texting, and phone calls, along with zone entry monitoring that checks for safety equipment. The system supports both live CCTV feeds and pre-recorded videos, which offers flexibility. Below are the key software tools and frameworks used in the development of the system:

1. **OpenCV**
   OpenCV was essential in the project, enabling real-time video stream processing and frame analysis. It seamlessly integrates with YOLO to support detection, dynamic tracking, and live video frame visualization within the interface.

2. **Ultralytics**
   The Ultralytics library simplified the implementation of YOLOv8. Providing pre-trained models and an adaptable framework for fine-tuning accelerated development, allowing the team to customize models effectively for the specific use case.

3. **Tkinter**

   Tkinter was used to create a user-friendly graphical interface that allows users to interact with the system intuitively—whether it is for live monitoring, receiving alerts, or managing system settings. Figure 7 showcases an example of the Tkinter interface, illustrating its simplicity and effectiveness.

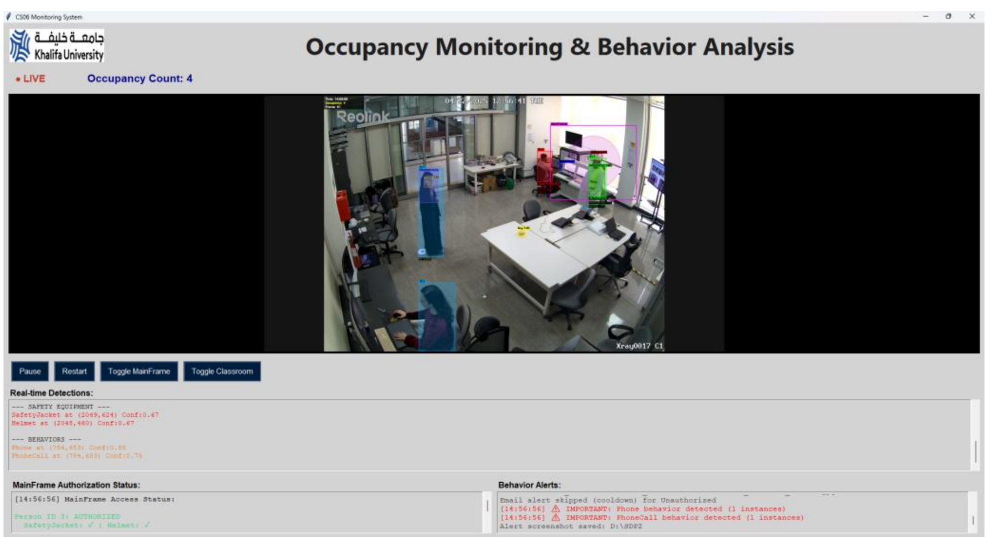


Figure 7 Showcasing the Simplicity of Tkinter

4. **PyTorch (Python 3.10.15)**

   PyTorch served as the backbone of the model training process. The advanced features for managing large-scale datasets and resource-intensive tasks made it the ideal choice for our needs. By leveraging GPU acceleration, training times were significantly reduced through parallel computations, which was essential given the size of the data.

   Furthermore, PyTorch's dynamic computation graph provided the flexibility needed to experiment with various model architectures and refine training strategies efficiently. Its compatibility with Python 3.10.15 ensured smooth integration with other tools and libraries [10].

In addition to the main libraries mentioned above, the following tools were integrated to enable advanced features such as automated email alerts, event logging, and live responsiveness.

1. **smtplib, email.message, email.mime**

   The smtplib in Python allows the system to make use of an SMTP connection, in the case of our system, this allows us to make a connection to Gmail [11].

2. **Threading**

   The threading module played a key role in handling multiple tasks at the same time, without slowing down the main application. It allowed the system to run background processes like capturing frames or sending alerts, all while the user interface remained responsive[12].

3. **Queue**

   The queue model was used to manage the flow of data between threads safely. For example, one thread could keep adding video frames to a queue, while another thread processes those frames for analysis. This allows the system to avoid crashes especially when multiple threads are attempting to access shared resources[13].

4. **PIL (Pillow)**

   Pillow converts OpenCV images to a format that is compatible with Tkinter, additionally it aids in the preparation of snapshots for display and email delivery [14].

By integrating these advanced tools and methodologies, the system will be designed to deliver insights that enhance safety and improve the learning experience across educational institutes for instructors and students alike.

Chapter 3: Dataset Construction

3.1 Data Configuration

Constructing a high-quality dataset is a foundational step in ensuring the success of any machine learning system. The accuracy and reliability of model predictions are directly influenced by the relevance and diversity of the training data. This chapter outlines how the datasets were prepared for two distinct models used in the project—SDP I and SDP II—each tailored to different objectives within the classroom monitoring environment.

SDP I – Object Detection Model (Model 1)

The first model, developed during the SDP I phase, was designed to detect commonly encountered objects within the classroom setting. The dataset for Model 1 included the following seven object classes:

```yaml
nc: 7
names:
  0: Cup
  1: MainFrame
  2: Bottle
  3: Human
  4: Snack
  5: Mug
  6: Biscuit
```

Figure 8 SDP I Model Data.yaml classes

This selection focused on identifying physical items and individuals in the environment. The inclusion of the “MainFrame” class specifically represented a restricted zone in the classroom, which was critical for flagging violations related to unauthorized access.

SDP II – Behavior Detection Model (Model 2)

In the SDP II phase, the focus shifted from object detection to recognizing behaviors and safety compliance. A second model, Model 2, was trained using a different dataset that introduced a new set of classes. These included both wearable safety equipment and observable behaviors relevant to the classroom context:

```
7: SafetyJacket
8: Helmet
9: Eating
10: Drinking
11: Classroom
12: Phone
13: PhoneCall
14: Texting
15: Cheating
```

Figure 9 SDP II Model data.yaml Classes

This model enabled the system to detect and log specific human behaviors—such as eating, drinking, or phone usage—as well as ensure safety compliance by identifying the presence of helmets and safety jackets. Unlike Model 1, which focused purely on object detection, Model 2 was designed to infer human activity and assess behavioral norms or violations in real time.

3.2 Limitations & Assumptions

While the project's dataset and training methodology were carefully developed to support accurate detection in both lab and classroom environments, certain limitations and foundational assumptions may affect the system's overall robustness and scalability. Recognizing these limitations is essential to contextualize the system's performance and to guide future iterations.

3.2.1 Limited Diversity in Imaging Conditions

In the early phases of the project, the primary dataset was captured within a controlled laboratory setup using three fixed camera angles. While this setup provided consistency and clarity during training, it inherently lacked exposure to diverse environmental contexts. Variations such as different classroom layouts, furniture types, background noise, or lighting conditions — which are common in real-world deployments — were underrepresented. This lack of diversity means the model may perform well in similar structured environments but struggle to maintain accuracy in unfamiliar scenarios. While the team incorporated Roboflow datasets to introduce external variability, these were still limited in scope compared to the unpredictable nature of real-world settings.

In the next phase of the project, the team expanded diversity in imaging conditions. Footage from various classrooms around the university campus were included. Additionally, videos recorded from different sources such as mobile phones and external camera sources were provided. This drastically helped improve environmental variability and simulate more realistic conditions. Despite these improvements, certain limitations still remain, for example highly cluttered classrooms and unusual furniture layouts may still be underrepresented.

For broader adoption, the model will require exposure to a wider range of physical conditions to build visual resilience and maintain consistent performance across different schools, labs, or institutions.

3.2.2 Generalizability to Real-World Scenarios

Generalization refers to the model's ability to apply what it learned during training to new, unseen environments. This is not just a desirable trait — it is essential for real-world utility. A model that performs perfectly in its training space but fails in a live classroom or lab is neither reliable nor scalable.

Real-world deployment introduces dynamic challenges. Students may move unpredictably, lighting can vary dramatically throughout the day, and obstructions such as bags, chairs, or peers may partially block critical objects or behaviors. Moreover, behavior variability exists; subtle behaviors such as texting or cheating can differ dramatically from one person to another, both in posture and context.

As mentioned in the section above, to mitigate these challenges the team incorporated videos from various settings. This has already enhanced the model's exposure to realistic classroom settings. However, generalization remains to be an ongoing challenge, especially for low-frequency, high-risk behaviors such as cheating, which exhibit high variability across users and different contexts.

For these reasons, generalization is crucial for:

- **Reducing false positives** in unpredictable settings
- **Ensuring fairness and accuracy** across different users
- **Making the system adaptable** to new deployments without constant retraining

In future iterations, the system could benefit from adaptive learning techniques and real-world user feedback loops to continue improving generalization. By maintaining a strong focus on this goal, the team aims to build a system that is not just a working prototype, but a deployable and trustworthy solution.

3.2.3 Class Imbalance in Object and Behavior Detection

A significant limitation encountered during both phases of development — SDP I (object detection) and SDP II (behavior recognition) — was the issue of class imbalance. This refers to the unequal distribution of labeled examples across different classes, which can cause the model to overfit on frequently seen categories while underperforming on rarer but equally important ones.

In SDP I, classes like “Human” and “MainFrame” were highly represented in the dataset due to their frequent occurrence in lab footage. In contrast, smaller objects such as “Cup”or “Mug” were less common, resulting in poorer detection accuracy during validation and testing. These items were especially prone to being missed in cluttered scenes, occluded views, or under low lighting.

To address this imbalance, the team:

- Performed **targeted data augmentation** (rotations, scaling, occlusions) on low-frequency object classes.
- Adjusted **labeling priorities** during manual annotation to increase representation of small object instances.
- **Retrained the model multiple times**, fine-tuning hyperparameters and loss weighting to improve sensitivity toward underrepresented objects.

In SDP II, the imbalance was even more impactful due to the subtle nature of certain behaviors. Classes like “Eating”, “Drinking”, and “Texting” occurred far less frequently compared to more visible and consistent classes such as “Cheating”, “SafetyJacket”, or “Classroom.”

This created a challenge in training the model to detect nuanced behaviors that are high-risk but low-frequency. These behaviors often vary between individuals (e.g., texting styles, phone holding angles), making them harder for the model to learn from limited samples.

To mitigate this, the team:

- **Manually extracted more behavioral frames** from lab videos focused on edge cases and hard-to-detect actions.
- Integrated **behavior-specific Roboflow datasets** to supplement the rare classes.
- Applied **aggressive augmentation** to simulate different contexts for underrepresented behaviors (e.g., lighting, body posture, occlusion).
- Iteratively validated and retrained the model to correct misclassifications, especially for behaviors that trigger alerts.

While these efforts greatly improved model robustness, class imbalance remains a key limitation, especially when it comes to real-time performance in complex or unpredictable environments. Future improvements will focus on incorporating adaptive learning strategies, real-world feedback loops, and potentially synthetic data generation to further balance the detection landscape.

3.3 Data Acquisition

3.3.1 Data Collection

The data collection process was carried out within the laboratory environment using three strategically installed cameras. This multi-angle configuration enabled the team to capture video data that reflected real-world operating conditions, including diverse perspectives, variable lighting, dynamic human interactions, and differing object arrangements. This setup was essential for acquiring high-quality footage for both object detection (e.g., cups, bottles, snacks) and behavior recognition (e.g., eating, phone usage, wearing safety gear).

In the second phase, data collection was expanded to include multiple classroom environments across campus, captured using both mobile phone cameras and external handheld cameras. This extension significantly increased the diversity of camera perspectives and background clutter. This helped simulate a broader range of scenarios which aided in the preparation of the system for deployment.

To further enhance variability and address edge cases, the team continued to incorporate pre-annotated datasets from Roboflow. This technique was implemented in both phases of the project in order to mitigate class imbalance issues.

Frames were then extracted from the collected video streams and manually annotated to label relevant objects and behaviors, aligning with the requirements of the two models. By combining real-world lab footage with curated external data, the final dataset was both context-specific and adaptable to broader environments. This hybrid approach laid a strong foundation for training both object-based and behavior-based detection models, improving their accuracy and practical effectiveness during real-time deployment.

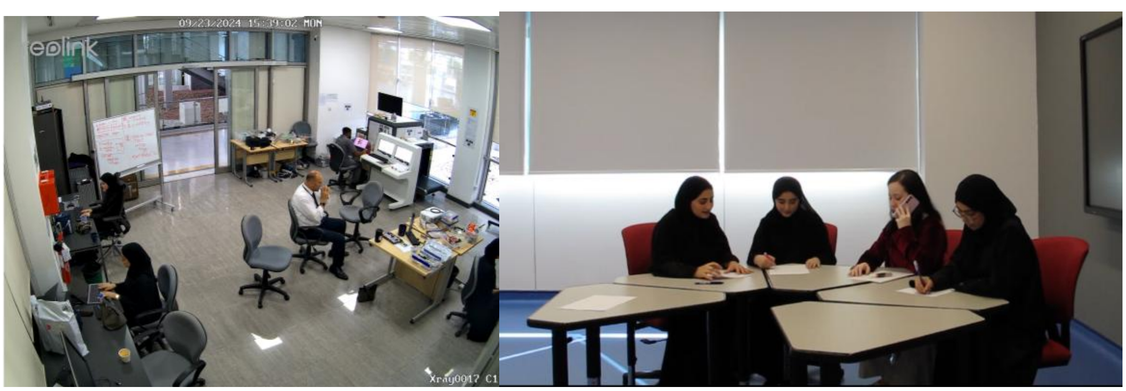


Figure 10 Data Collection from Multiple Cameras in the Lab

3.3.2 Incorporating Roboflow Datasets

To further enrich the training data and improve model generalization, the team integrated supplementary datasets from Roboflow, a widely recognized platform for sharing and managing computer vision datasets. These datasets included high-quality, pre-annotated images featuring various objects such as cups, mugs, bottles, eating, drinking and other relevant classes in a wide range of shapes, angles, and environmental contexts.

Roboflow's ready-to-use datasets, particularly those formatted for YOLOv8, significantly accelerated the dataset preparation process by reducing the time and effort required for manual annotation. The diversity in object appearance and background conditions provided by Roboflow complemented the lab-collected data and introduced variability critical for preventing overfitting and underfitting during model training.

By incorporating Roboflow images into the augmented dataset, the team ensured that the models could recognize target objects in unfamiliar or complex settings—such as different lighting conditions, backgrounds, or object orientations—not explicitly present in the laboratory environment. This strategic enhancement boosted the models' robustness and performance in real-world scenarios beyond the lab.

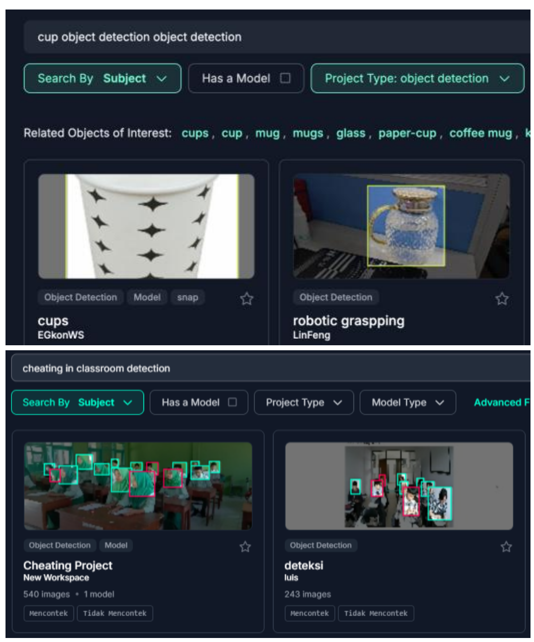


Figure 11 Datasets collected from Roboflow

3.3.3 Object-Specific Data Inclusion and Augmentation

SDP I Model

Case 1: Cup

Reason for Inclusion:
Cups are a frequently occurring object in the lab environment, varying in design, size, and material. Accurate detection of cups across diverse scenarios is vital for the system's effectiveness.

Data Augmentation:
To improve the model's ability to generalize across different situations, the team incorporated cups in various:

- **Colors:** Red, blue, white, and transparent to reflect variations in appearance.
- **Shapes and Sizes:** Examples included large mugs, small coffee cups, and disposable cups.
- **Orientations:** Rotations (e.g., 45°, 90°) simulated real-world viewing angles.
- **Lighting Conditions:** Frames with both dim and bright lighting conditions were added.
- **Background Variability:** Frames included cluttered tables, clean surfaces, and cups partially occluded by other objects to mimic challenging scenarios.

Case 2: MainFrame (Restricted Area)

Reason for Inclusion:
The 'MainFrame' represents a restricted area requiring constant monitoring to ensure no unauthorized access occurs. Accurate detection of humans entering this zone is crucial.
Data Augmentation:

- **Perspective Changes:** The MainFrame zone was labeled from various camera angles to provide diverse viewpoints.
- **Lighting Variability:** Frames with both bright and dim lighting conditions were added to simulate different times of day.
- **Edge Cases:** Examples where the MainFrame boundary was partially obscured by objects were included to test detection robustness.

Case 3: Bottle

Reason for Inclusion:
Bottles are commonly found in the lab and can be confused with similar items like glasses or jugs.

Data Augmentation:

- **Class Consistency:** All water bottle labels were remapped to 'Bottle' for uniformity.
- **Shapes and Sizes:** Included cylindrical bottles, flat bottles, and sport-style bottles.
- **Rotations:** Examples of bottles lying on their side, upright, or tilted.
- **Color Variations:** Transparent, green, and blue bottles were added

Case 4: Human

Reason for Inclusion:
Humans are the primary entities for monitoring. YOLOv8's pre-trained models already exhibit high accuracy in human detection.

Enhancements:

- Video frames from the lab environment were integrated to ensure reliable detection of humans in the specific context.
- Varied postures such as standing, sitting, and bending were included to reflect real-world scenarios.

Case 5: Mug

Reason for Inclusion:
Mugs, while similar to cups, differ in shape, size, and function. Including mugs as a separate class ensures precise detection.

Data Augmentation:

- **Diverse Designs:** Examples included plain, patterned, and branded mugs.
- **Handle Variations:** Mugs with handles oriented to the left, right, or hidden due to occlusion were added.
- **Perspective Changes:** Frames captured mugs from top-down, side-on, and angled views to simulate real-world appearances.

By addressing each object class with tailored data augmentation and diversity, the team ensured a balanced and comprehensive dataset. This approach enhances the detection model's ability to generalize and perform reliably in dynamic, real-world environments. The inclusion of the 'MainFrame' restricted area demonstrates the system's dual capabilities in object detection and spatial monitoring, making it a robust tool for lab environments.


Figure 12 Visual Representation of Data Augmentation

SDP II Model

Case 1: SafetyJacket

Reason for Inclusion:
Safety jackets are essential for monitoring safety compliance in the lab. Detection of these jackets helps ensure that lab personnel are following safety protocols, especially in hazardous areas.
Data Augmentation:

- **Occlusion Variations:** Frames where the jacket is partially covered by equipment or other objects, simulating real-world scenarios.
- **Lighting Conditions:** Examples with both bright and low lighting to test robustness under varying conditions.

Case 2: Helmet

Reason for Inclusion:

Helmets are another important piece of safety equipment. Monitoring their presence ensures that safety measures are being adhered to in environments that require protective gear.
Data Augmentation:

- **Rotation:** Helmets were shown from various angles to ensure the system could detect them from different perspectives.
- **Lighting Variability:** Images in both bright and dim lighting were included to cover various environmental conditions.
- **Partial Occlusion:** Frames where helmets are partially obscured by other objects or individuals, testing the model's detection robustness.

Case 3: Eating

Reason for Inclusion:

Eating in the lab is prohibited for student and staff safety, as food and drinks can create hazards and interfere with experiments. Monitoring this behavior helps ensure safety protocols are followed.

Data Augmentation:

- **Postures:** Examples were included where people are eating while standing, sitting, or bending.
- **Food Variations:** Different types of food (e.g., sandwiches, snacks, and drinks) were included to increase the model's ability to recognize eating behavior in varied contexts.
- **Lighting Conditions:** Frames with both bright and low lighting to simulate different times of day.
- **Occlusion:** Instances where the person's hands or food items are partially blocked by objects or other people, testing the model's resilience in detecting eating behavior despite partial occlusion.

Case 4: Drinking

Reason for Inclusion:

Drinking in the lab is prohibited for student and staff safety, as it can create spills, damage equipment, or cause distractions. Detecting this behavior ensures compliance with lab safety protocols.

Data Augmentation:

- **Containers:** Examples included people drinking from bottles, mugs, and cups adding variety in the types of containers.
- **Postures:** Drinking while standing, sitting, or walking to simulate various real-world situations.
- **Angles and Rotations:** Frames showing people drinking from different angles, such as side views and top-down views.
- **Lighting Conditions:** Both bright and dim lighting scenarios were included for generalization.
- **Partial Occlusion:** Drinking instances where the drink is obscured by other objects or people, testing detection in cluttered settings.

Case 5: Classroom

Reason for Inclusion:

Defines the physical boundaries of the lab space where human activity (i.e., occupancy) is monitored. Tracking entry and exit from this designated space is essential for accurate occupancy counting.

Data Augmentation:

- **Lighting Variability:** Examples with varying lighting conditions, from natural light during the day to artificial lighting at night.
- **Angles and Rotations:** Frames showing classroom from different angles, such as side views and top-down views.

Case 6: Phone

Reason for Inclusion:

Essential for monitoring students during exams, especially to detect potential cheating activities. Phones can be used to access unauthorized materials, communicate with others, or search for answers during exams. This class helps the system recognize when a student is using a phone in a prohibited environment.

Data Augmentation:

- **Device Variations:** Different phone models and colors were included to reflect the diversity of devices.
- **Hand and Pocket Detection:** Examples where phones are held in hands or stored in pockets were added to increase generalizability.
- **Occlusion Variability:** Frames with phones partially obscured by hands, books, or other objects, to simulate common real-world conditions.
- **Lighting Conditions:** Frames captured under different lighting scenarios, including dim and bright conditions.

Case 7: PhoneCall

Reason for Inclusion:

Crucial for detecting instances where students may make or receive phone calls during an exam. Phone calls in such a context could allow students to cheat by discussing answers with others, making this class essential for maintaining exam integrity.

Data Augmentation:

- **Postures and Movements:** Instances where individuals are seen talking on the phone while standing, sitting, or walking.
- **Phone Placement:** Frames where the phone is held to the ear.
- **Background Variability:** Frames from different classroom environments with varying levels of clutter.
- **Lighting Conditions:** Both bright and dim lighting were considered for robustness.

Case 8: Texting

Reason for Inclusion:

Focuses on detecting students who are sending or receiving text messages during exams, which is another form of cheating. Texting allows students to communicate with others to share or receive answers, making it a high-priority behavior to track in exam environments.

Data Augmentation:

- **Phone Types:** A range of phone models and colors were used to ensure variety in the dataset.
- **Lighting Variability:** Examples of texting in different lighting conditions, from bright daylight to low-light environments.
- **Partial Occlusion:** Instances where the phone or hands are partially obstructed, testing the model's robustness to occlusion.

Case 9: Cheating

Reason for Inclusion:

Detecting cheating behavior is crucial for ensuring integrity in the lab or classroom, where unauthorized activities may occur.

Data Augmentation:

- **Postures:** Instances of students looking at notes using mobile phones during an exam were included.
- **Lighting Variability:** Frames with both bright and dim lighting to account for different lighting setups in classrooms or examination halls.
- **Occlusion:** Frames where cheating behavior is partially concealed by desks, other students, or classroom equipment, challenging the model's detection ability.
- **Activity Diversity:** Various forms of cheating were captured, from copying answers to using hidden devices.

By applying targeted data augmentation strategies for each object and behavior class, the team ensured a comprehensive and diverse dataset for SDP II Model. This not only improved the accuracy and reliability of the detection system but also enabled the model to handle complex, real-world scenarios with greater generalization.

3.3.4 Feature Engineering Techniques

In our project, feature engineering was crucial in optimizing the performance of the YOLOv8 model for real-time object detection and tracking in video feeds. Several feature engineering techniques were used to extract meaningful information from the video frames and improve detection accuracy:

1. **Bounding Box Features:** The bounding box coordinates, which define the position and dimensions of detected objects, were central to our feature extraction. These coordinates were essential for localizing objects in each frame. Additionally, attributes like the area and aspect ratio of the bounding boxes were derived to help differentiate objects with varying sizes and shapes.
2. **Confidence Scores:** YOLOv8 outputs a confidence score for each detected object, reflecting the model's certainty about the presence of the object within a bounding box. These confidence scores were utilized as a feature to filter out detections that were less reliable, prioritizing high-confidence predictions for further analysis.
3. **Object Class Labels:** The model's predicted object class labels, such as person, car, or animal, were used to categorize detected objects. These class labels were important for distinguishing between different object types, enabling the system to apply relevant processing techniques depending on the object detected.
4. **Behavior Recognition Features**: Behavior detection, such as identifying eating and drinking actions, was handled similarly to object detection. We extracted frames from recorded lab videos and manually labeled each frame based on the behavior occurring. Additional behavior-specific images were sourced from Roboflow to enrich the dataset and increase diversity. Bounding boxes and class labels were assigned to behaviors just like objects, allowing the model to learn and generalize from visual cues associated with human actions in the lab environment.

Together, these engineered features enabled the YOLOv8 model to better understand both static object presence and dynamic behavior patterns, ultimately improving detection accuracy and system responsiveness.

3.3.5 Overfit and Underfit

Overfitting and underfitting are key challenges in machine learning that can limit a model’s ability to generalize effectively to unseen data. Overfitting occurs when a model becomes too tailored to the training data, memorizing noise and irrelevant details instead of identifying meaningful patterns. This results in excellent performance on the training data but poor results when applied to new, unseen data. On the other hand, underfitting happens when a model is too simplistic to recognize the underlying patterns in the data, leading to poor performance on both the training set and new data.

Striking a balance between overfitting and underfitting is crucial for building a model that generalizes well. This involves finding the right level of complexity in the model to ensure it learns relevant features while avoiding unnecessary details. Generalization—the ability to apply what the model has learned from the training data to unseen data—depends heavily on the quality and diversity of the dataset, as well as the model's architecture and training process [15].

To tackle these challenges, the team utilized a carefully curated and diverse dataset, which combined data collected in a lab environment with high-quality, pre-labeled datasets from Roboflow. The addition of lab video frames allowed the model to capture real-world scenarios with unique lighting conditions, camera angles, and specific classroom objects. The Roboflow dataset contributed variety by including items such as cups, mugs, and bottles of different shapes, sizes, and colors, captured in various environments. Additionally, for behavior classification tasks such as detecting actions like eating and drinking, the team labeled frames from lab-recorded videos and supplemented them with behavior-specific images from Roboflow. This enriched the dataset with varied behavioral contexts, helping the model learn to identify subtle action patterns across different individuals and scenarios.

This dual-source approach created a balanced dataset that reduced the risks of overfitting and underfitting. By combining structured data (from Roboflow) and unstructured data (from lab video frames), across both object and behavior categories, the team ensured that the model could generalize effectively to new, unseen data. Moreover, data augmentation techniques were employed to further diversify the dataset, preventing the model from memorizing training examples and enhancing its adaptability to real-world situations. This methodology established a solid foundation for achieving a model that performs reliably across diverse conditions and complex environments.

Figure 13 Visual Representation of Underfitting and Overfitting

3.4 Data Augmentation

To increase the diversity and volume of the training dataset, the team applied data augmentation to the extracted frames. Augmentation is a critical step that ensures the model is exposed to a variety of scenarios, improving its generalization capability and robustness against unseen conditions.

Frame Extraction

The team developed a Python script to systematically extract 18 frames from each video at regular intervals. This approach was carefully designed to capture diverse moments within the video without introducing redundancy. By doing so, the dataset retained the uniqueness of each instance while covering various angles, lighting conditions, and object positions within the videos. Each extracted frame underwent one randomly selected augmentation from a set of transformations, which included rotation, flipping, brightness adjustment, scaling and cropping.

Why Augmentation Was Applied

1. **Prevent Overfitting**: Augmentations exposed the model to diverse scenarios, reducing the risk of memorizing training data and instead encouraging generalization.
2. **Improve Real-World Adaptability**: Augmentation ensured that the model could detect objects across varying conditions such as lighting changes, different object orientations, and partial occlusions.
3. **Expand Dataset Volume**: The transformations increased the dataset size without requiring additional data collection efforts, creating a cost-effective and efficient way to enhance model performance.
4. **Enhance Detection Accuracy**: By training on augmented images, the model's capacity to recognize objects in unpredictable settings was significantly improved.

Impact of Augmentation

The application of augmentation techniques substantially increased the diversity of training examples. This played a crucial role in making the model more resilient to real-world complexities, such as detecting cups, bottles, and restricted areas in dynamic environments with variations in perspective, lighting, and occlusions. Ultimately, this step helped the system achieve higher detection accuracy and adaptability, even in challenging and unstructured settings [16].

3.5 Data Labeling

The data labeling process was conducted meticulously to ensure the highest quality annotations for training the object detection model. The steps were as follows:

1. **Uploading Frames:** Extracted frames from the dataset were uploaded to the MakeSense.AI interface, an intuitive tool designed for efficient data annotation.
2. **Manual Labeling:** Team members manually labeled each frame by drawing bounding boxes around objects or areas of interest. Each bounding box was assigned the corresponding class label, such as "Cup" or "MainFrame", ensuring accurate categorization of all detected entities.
3. **Exporting Annotations:** The completed annotations were saved in the YOLO format, a lightweight structure that integrates seamlessly into the training pipeline for object detection models.

To maintain consistency and avoid annotation errors, each labeled frame was reviewed by at least one other team member. This peer-review step helped ensure that bounding boxes were correctly placed and class labels were accurately assigned across all frames.

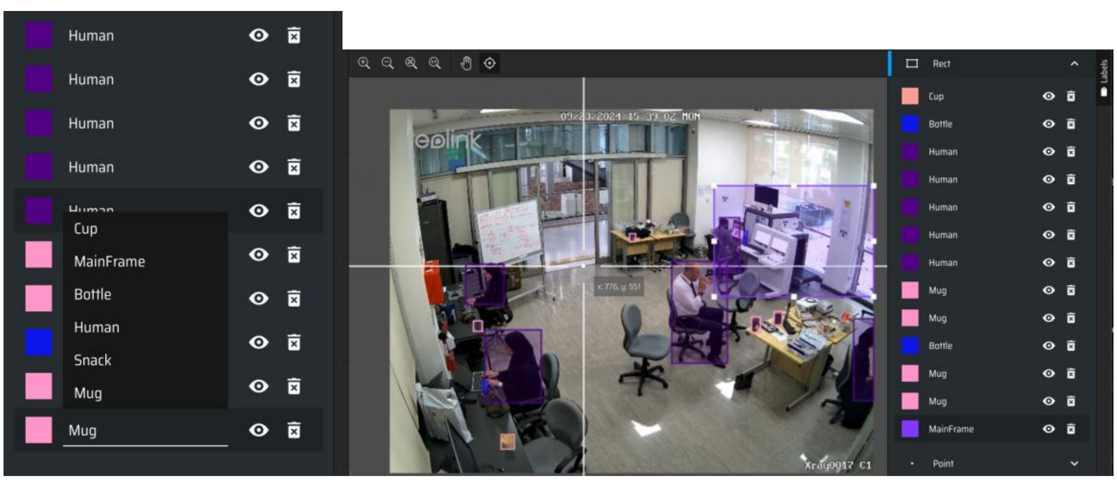


Figure 14 Manual Annotation Using MakeSense

This manual labeling process ensured the annotations were not only accurate but also consistent across the dataset. These high-quality annotations were vital for optimizing the performance of the YOLOv8 model, allowing it to learn effectively from the labeled data and achieve robust detection capabilities in real-world scenarios.

3.6 Training, Validation, and Testing Split

To ensure the model's generalization and reliability, the dataset was divided into Training, Validation, and Testing sets with the following distribution:

- **Training Set (70%):**

  This set was used to train the model by providing labeled images for parameter optimization. The large proportion of training data ensured that the model could learn effectively from diverse examples.

- **Validation Set (20%):**

  During training, this set was used to monitor the model's performance and fine-tune hyperparameters. Validation helped prevent overfitting by providing an independent dataset to assess the model's adaptability to unseen data.

- **Testing Set (10%):**

  Reserved exclusively for final evaluation, the testing set allowed the team to measure the model's performance on entirely unseen data. This phase provided an objective assessment of the model's accuracy, precision, and recall in real-world scenarios.

This distribution ensured that the model had sufficient data for learning while maintaining dedicated subsets for validation and testing, resulting in a robust and generalizable detection system.

Chapter 4: Steps of Development

4.1 Steps of Development

Machine learning is the key technology for this project, serving as the foundation for real-time object detection and behavior analysis. The development process follows a structured pipeline that begins with problem definition, followed by data collection and annotation. The core of the system is built on the YOLOv8 framework, known for its high speed and accuracy in detection. The essential steps include model training, validation, and testing, which ensure the system's reliability and ability to adapt to dynamic environments. This approach allows for effective object detection and alert mechanisms, even in complex scenarios.

4.2 Problem Definition

The system is designed to address a range of real-time monitoring challenges within a lab environment. These include both object detection and behavior recognition to enhance safety, enforce regulations, and automate surveillance tasks:

1. **Real-time Object Detection and Classification:** Accurately identifying and classifying objects in a dynamic lab setting to monitor key items such as cups, bottles, and mugs. This enables effective inventory management and violation detection (e.g., drinking in restricted zones).

2. **Detection of Unauthorized Proximity to Restricted Areas:** Monitoring and identifying breaches of designated restricted zones (e.g., the "MainFrame") to ensure compliance with access protocols. This includes detecting whether individuals entering these zones are wearing the required safety gear, such as helmets and jackets.

3. **Behavior Recognition for Rule Enforcement:** The system detects specific human behaviors to identify violations, including:

    - **Cheating incidents** such as using a phone for calls or texting during restricted times.
    - **Non-compliance with lab safety protocols**, like entering without a helmet or jacket.
    - **Prohibited actions** such as eating or drinking in the lab when not allowed.

By addressing these multiple challenges, the system enhances lab safety, rule enforcement, and overall situational awareness through real-time, intelligent video monitoring.

4.3 Model Selection

YOLOv8 is a state-of-the-art deep learning framework for object detection, chosen for its real-time performance and high accuracy. Its capability to handle multiple objects in a single frame makes it ideal for dynamic environments like laboratories.

During the development phase, the team also evaluated YOLOv5 as a potential model. Although YOLOv5 showed good performance, its detection accuracy and speed were not as robust as those of YOLOv8 when applied to our dataset, especially for smaller object categories such as "Cup" and "Mug," or in cases of overlapping objects. YOLOv8's anchor-free architecture and enhanced dynamic detection head contributed to better generalization and improved performance in these situations. Therefore, YOLOv8 was selected as the final model for this project.

In SDP II, as the project expanded beyond object detection to include behavior recognition (such as identifying cheating, safety violations, and eating or drinking in prohibited areas), YOLOv8 was still used due to its adaptability and strong detection capabilities. Behaviors were treated similarly to objects, with bounding boxes and class labels applied to frames depicting specific actions, enabling the model to learn and generalize these human behaviors effectively.

Figure 14 compares the performance of different versions of YOLO (You Only Look Once) in terms of latency and accuracy on the COCO dataset. The graph plots the COCO mAP50–95 (mean Average Precision at IoU thresholds from 0.5 to 0.95) against the latency, measured in milliseconds per image, for various YOLO models. The blue line represents YOLOv8, which consistently outperforms the earlier versions (YOLOv7, YOLOv6-2.0, and YOLOv5-7.0) in terms of accuracy while maintaining lower latency [17].

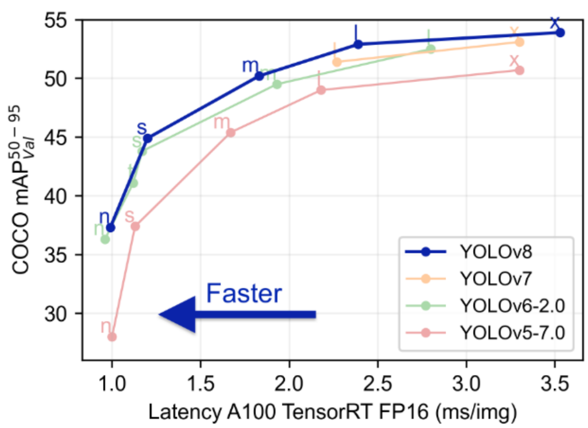


Figure 15 Performance of Various YOLO Models

4.4 Training Process

The training process was designed with careful consideration of the dataset's structure, model requirements, and the computational resources available. The goal was to develop a robust object detection system capable of accurate predictions across diverse scenarios. This section outlines the key elements of the training process, including the rationale behind the chosen configurations.

4.4.1 Training Data Split

A significant portion of the data, 70%, was allocated for training to ensure the model had sufficient exposure to varied examples. This decision was made to enable the model to learn a wide range of patterns, object variations, and scenarios effectively.

4.4.2 Batch Size and Epochs

- **Batch Size:** The batch size was set at 16 to optimize GPU utilization without risking memory overflow. This size provided a balance between computational efficiency and stability during convergence.
- **Epochs:** The model was trained for 60 epochs, a number determined after initial experimentation. This duration allowed the model adequate time to learn from the data without overfitting. Early stopping mechanisms based on validation metrics were employed to halt training if the model began to exhibit signs of overfitting. The choice of 60 epochs ensured sufficient iterations for effective convergence, especially for larger objects like "Human" and "MainFrame," while highlighting areas for improvement in detecting smaller objects such as "Cup" and "Mug."

4.4.3 Training Workflow

Model Initialization:

The team leveraged a combination of pre-trained COCO weights, a custom lab-collected dataset, and curated data from Roboflow to enhance robustness and adaptability. The lab-collected dataset included annotated video frames representing real-world classroom and exam scenarios, captured from varied angles and lighting conditions. These frames featured not only key objects like "Cup," "Mug," and the "MainFrame" restricted area (used in SDP I), but also human behaviors such as "Eating," "Drinking," "Phone Use," and safety compliance markers like "Helmet" and "SafetyJacket" (used in SDP II). The Roboflow datasets contributed further variability, with high-quality images of both objects and behaviors across different environments. This multi-source, multi-purpose approach minimized the risks of overfitting and underfitting and supported both object-based and behavior-based detection tasks.

Loss Function Optimization:

The model's composite loss function included:

- **Bounding Box Loss** – for precise localization of both objects and behaviors
- **Classification Loss** – for accurate class labeling
- **Objectness Loss** – to reduce false positives and focus on relevant detections

Learning Rate Scheduling:

A cyclical learning rate scheduler was used to dynamically adjust learning rates during training. This helped the model balance exploration of new parameters and convergence speed.

Augmentation and Preprocessing:

Random rotations, flips, brightness variations, cropping, and scaling were applied to both object and behavior classes. For example, drinking actions were captured in various postures and lighting conditions, while cups and bottles appeared with occlusion and angle variations. This exposure improved the model's adaptability to unpredictable real-world environments.

Evaluation During Training:

Key metrics such as mean Average Precision (mAP), precision, and recall were monitored per epoch. These guided dynamic adjustments in learning rate and augmentation parameters to improve detection performance across both object and behavior categories.

Results and Observations:

- **Performance on Larger Objects/Actions:** The model achieved high accuracy in detecting large, prominent elements such as "Human" and "MainFrame," and well-defined behaviors like "Cheating".
- **Challenges with Small or Subtle Objects:** Detection rates for smaller items (e.g., "Cup", "Mug") and nuanced behaviors (e.g., "phone usage") required targeted fine-tuning and specialized augmentations to improve recognition accuracy.

The training process successfully integrated a diverse dataset, optimized hyperparameters, and comprehensive augmentation techniques to support both object detection (SDP I) and behavior recognition (SDP II). By allocating 70% of the data for training and actively monitoring validation performance over 60 epochs, the team developed a robust and flexible model capable of generalizing across classroom and exam monitoring scenarios.

4.5 Validation Process

The validation process played a key role in evaluating model performance and guiding improvements during training. A dedicated validation set, comprising 20% of the total dataset, was used to assess the model's ability to generalize to unseen data. YOLOv8's built-in evaluation tools provided metrics such as Mean Average Precision (mAP), Precision, and Recall — which collectively measured the system's effectiveness in detecting, classifying, and localizing both objects and human behaviors.

- In SDP I, validation focused on traditional object classes such as "Cup," "Bottle," "Mug," and "MainFrame."
- In SDP II, the validation set was expanded to include behavior-based classes such as "Cheating," "PhoneCall," "Eating," "Helmet," and "Classroom." These new additions required the model to handle more abstract and variable features, like motion, posture, and contextual cues.

4.5.1 Iterative Refinement

Feedback from the validation process enabled iterative improvements across both object and behavior detection tasks. Key refinement strategies included:

- **Underperforming Classes:**

  Early validation runs showed lower detection accuracy for smaller items (e.g., "Cup", "Mug") and subtle behaviors (e.g., "Texting", "Eating"). The team applied targeted data augmentation — including rotation, brightness shifts, and occlusion scenarios — to boost performance on these challenging classes.
- **Hyperparameter Tuning:**

  Parameters were fine-tuned based on validation trends:
  - **Learning Rate:** Dynamically adjusted using cyclical patterns to stabilize learning while improving convergence.
  - **Batch Size:** Tested across multiple values to maximize GPU efficiency while avoiding memory overload.
- **Behavior-Specific Fine-Tuning:**

  For SDP II, behaviors like "Cheating" and "PhoneCall" required fine-tuning on annotated lab scenarios. The team added diverse behavioral frames showing different camera angles, lighting conditions, and posture variations to help the model learn to generalize.

4.5.2 Impact of Validation and Refinement

The iterative validation and refinement cycle significantly improved the system's overall robustness. Notable improvements included:

- **Higher mAP scores across both object and behavior classes.**

- **Improved precision and recall** for difficult detections such as small objects and overlapping behaviors.
- **Reduced overfitting**, as confirmed by stability in validation metrics across varied conditions.
- **More consistent detection** under challenging lighting, occlusion, or angle distortions — especially for behaviors like "Eating" or "Texting."

By leveraging feedback from validation and making systematic adjustments, the team ensured the development of a generalized, real-time monitoring system capable of detecting both static objects and dynamic behaviors across classroom and lab environments.

4.6 Testing Process

The testing phase was conducted using 10% of the full dataset, exclusively reserved for final evaluation. This subset consisted of unseen videos, ensuring an unbiased assessment of the model's generalization capability across both object detection (SDP I) and behavior recognition (SDP II) tasks.

Key Objectives of Testing

1. Robust Object Detection (SDP I):

- The model's ability to detect objects such as "Human", "Cup", "Bottle", "Mug", and "MainFrame" was tested in dynamic, real-time settings.
- Particular focus was placed on smaller objects (like "Cup" and "Mug"), where consistent detection is typically more difficult.

2. Behavior Recognition and Rule Violation Detection (SDP II):

- Cheating Behaviors: Actions such as "PhoneCall", "Texting", and "Cheating" were tested to ensure accurate identification under different lighting, angles, and partial occlusions.
- Safety Compliance: Detection of "Helmet" and "SafetyJacket" was evaluated for robustness in ensuring that lab safety protocols were met.
- Prohibited Actions: The system was tested on its ability to recognize "Eating" and "Drinking" actions during restricted times, with a focus on different postures and container types.

3. Spatial and Occupancy Monitoring:

- The classroom bounding box was used to test occupancy tracking. The system was evaluated for:
    - Accurate increment/decrement of the occupancy count as people entered and exited.

    - Reliability across varying body positions and movement speeds.
- The "MainFrame" restricted zone alert mechanism was validated using an IoU-based logic to detect unauthorized human presence without safety gear.

Metrics Captured During Testing

- **Precision and Recall:** Assessed for both object and behavior classes to determine the rate of accurate detections vs. missed ones.
- **False Positive Rate:** Measured across all categories, especially alert-triggering behaviors like **Phone Use** and **Cheating**, to ensure the system remains practical and non-intrusive.
- **mAP Scores:** Calculated for behavior-specific classes to validate performance parity with object detection.
- **Occupancy Consistency:** Stability of the entry/exit detection mechanism was tracked over time to minimize count drift.
- **Alert Response Time:** Time between detection and alert triggering (screenshot or recording) was measured for responsiveness.

Observations and Insights

- **Strong Performance on Larger or Well-Defined Objects/Actions:**

  High accuracy was achieved in detecting “Human”, “MainFrame”, “PhoneCall”, and “Helmet”, even under varied lighting and partial occlusions.

- **Challenges in Subtle Behaviors and Small Items:**

  Occasional missed detections occurred for “Texting”, “Phone”, and “Mug”, particularly under fast hand movements or visual clutter.

- **Cheating Detection Reliability:**

  The system effectively captured cheating-related behaviors, triggering screen recordings or screenshots as configured, and organizing the evidence for post-session review.

- **Occupancy and Restricted Access Accuracy:**

  The classroom and MainFrame monitoring functions were consistently accurate, supporting both automated attendance tracking and safety enforcement.

- **Real-Time Capability:**

  The system sustained an average inference time of ~30 ms per frame, maintaining smooth real-time performance across both detection pipelines.

The comprehensive testing phase confirmed the system’s ability to generalize to new, real-world scenarios while maintaining strong detection accuracy, alert responsiveness, and real-time processing speed. These outcomes validated the reliability of both SDP I and SDP II systems in supporting safety, surveillance, and behavior monitoring in lab and classroom environments. Testing insights directly informed fine-tuning efforts — especially in improving detection for smaller objects and refining the behavior alert system for more nuanced cases.

Chapter 5: Concept Generation & Design Evaluation

5.1 User Interface Design Approach

The user interface design for the Classroom Occupancy Monitoring and Behavior Analysis System was developed with a focus on simplicity, clarity, and effective task execution. Rather than creating a complex, multi-tabbed application, we opted for a clear separation of functions through a two-system approach.

5.1.1 System Selection Interface

The design selection with SDP II began with the creation of a streamlined selection interface that serves as the entry point to our monitoring systems. This primary interface features:

- Three clearly labeled buttons: "Behavior System", "Cheating System", and “Attendance System”
- Brief descriptions explaining the purpose of each system
- Clean, minimalist design to reduce cognitive load
- Consistent visual styling that carries through to both systems

This approach allows users to immediately select the appropriate tool for their monitoring needs, eliminating confusion and reducing the learning curve. The separation into two distinct systems recognizes that classroom behavior monitoring, and exam proctoring have different requirements and priorities.

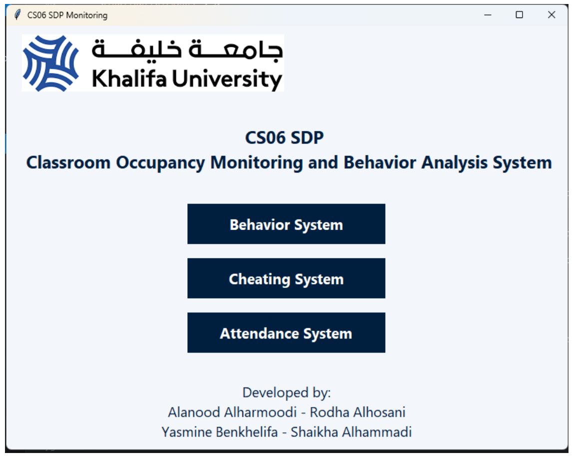


Figure 16 Main Window UI

5.1.2 Behavior Monitoring System Interface

The Behavior Monitoring System interface was designed specifically for monitoring safety compliance and general classroom behavior. Key design elements include:

- Prominent video display showing the classroom environment with detection overlays
- Clearly marked restricted areas (particularly the MainFrame zone)
- Visual indicators showing safety equipment detection status
- Alert panels that highlight violations with timestamp information
- Screenshot preview area for reviewing captured safety violations
- Email status indicators confirming when alerts have been sent
- Simple controls for adjusting detection sensitivity
- Saves detection logs as CSV files where these logs contain detailed information including timestamps, frame numbers, object classes (Human, Helmet, SafetyJacket, Phone, PhoneCall, MainFrame, etc.), confidence scores, bounding box coordinates (x1, y1, x2, y2), and center point locations.

The interface uses color coding to clearly communicate status information:

- Red highlights for safety violations
- Green indicators for proper safety compliance

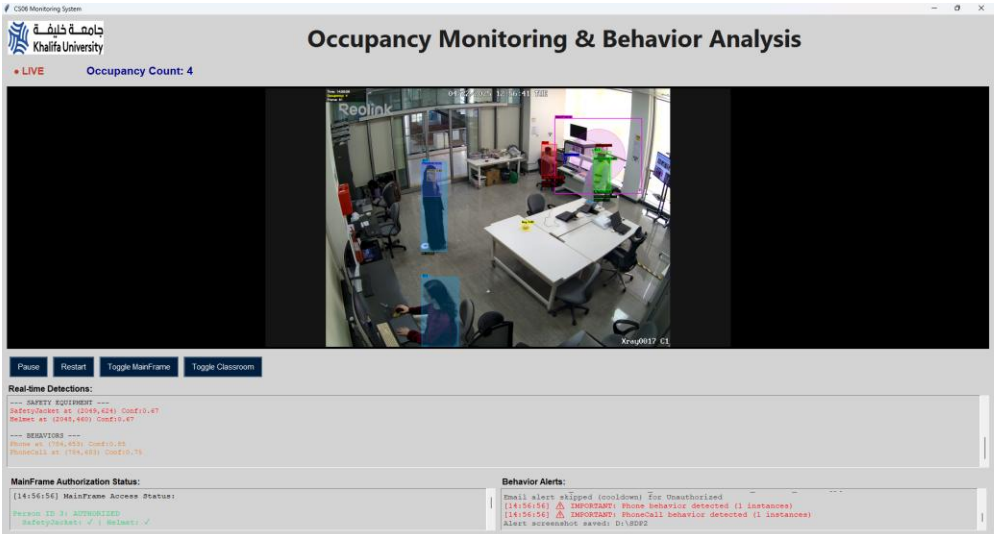


Figure 17 Behavior Monitoring System UI

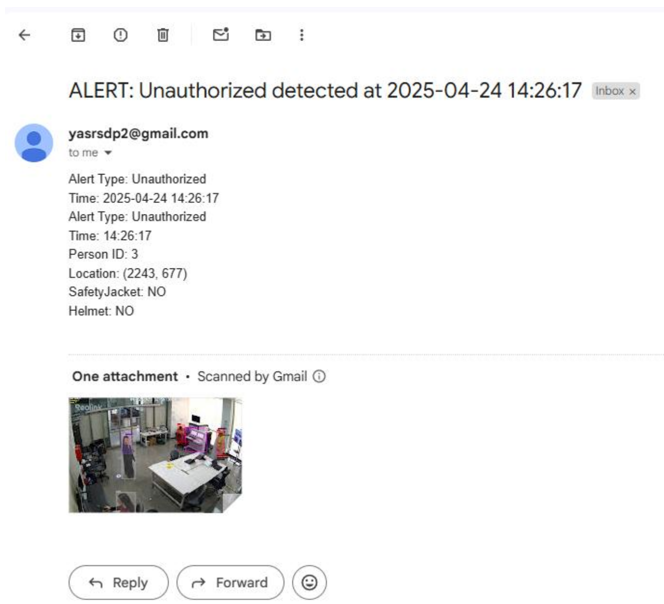


Figure 18 Behavior Monitoring System Email Alert

timestamp	frame	class	conf	x1	y1	x2	y2	center_x	center_y
2025-04-24 14:56:43	3	Phone	0.754206538	820	628	883	696	851	662
2025-04-24 14:56:43	3	Helmet	0.708012342	2207	476	2295	533	2251	504
2025-04-24 14:56:43	3	SafetyJacket	0.63432014	2202	575	2302	763	2252	669
2025-04-24 14:56:43	3	Human	0.873081744	781	563	984	1213	882	888
2025-04-24 14:56:43	3	Mug	0.818095148	1455	1003	1520	1054	1487	1028
2025-04-24 14:56:43	3	Human	0.752548397	708	1406	986	1909	847	1657
2025-04-24 14:56:43	3	MainFrame	0.695767403	1721	463	2036	811	1878	637
2025-04-24 14:56:43	3	Human	0.640309691	2184	505	2308	929	2246	717
2025-04-24 14:56:43	3	Human	0.59703213	1605	413	1718	649	1661	531
2025-04-24 14:56:43	3	PhoneCall	0.741876185	787	567	932	835	859	701
2025-04-24 14:56:43	4	Phone	0.766621709	824	628	885	698	854	663
2025-04-24 14:56:43	4	Helmet	0.708709121	2207	476	2295	533	2251	504
2025-04-24 14:56:43	4	SafetyJacket	0.650940239	2204	575	2302	741	2253	658
2025-04-24 14:56:43	4	Human	0.869117379	783	565	981	1207	882	886
2025-04-24 14:56:43	4	Mug	0.817542136	1455	1003	1520	1054	1487	1028
2025-04-24 14:56:43	4	Human	0.739915669	706	1406	986	1909	846	1657
2025-04-24 14:56:43	4	MainFrame	0.676300526	1718	463	2036	810	1877	636
2025-04-24 14:56:43	4	Human	0.599470973	1605	413	1718	649	1661	531
2025-04-24 14:56:43	4	Human	0.597827315	2174	503	2304	936	2239	719
2025-04-24 14:56:43	4	Mug	0.372929752	296	917	339	950	317	933
2025-04-24 14:56:43	4	Bottle	0.311252266	1771	505	1792	566	1781	535
2025-04-24 14:56:43	4	PhoneCall	0.739186764	787	565	933	838	860	701
2025-04-24 14:56:43	5	Phone	0.778170168	824	628	886	698	855	663
2025-04-24 14:56:43	5	Helmet	0.706822991	2207	476	2295	533	2251	504
2025-04-24 14:56:43	5	SafetyJacket	0.658565044	2202	575	2301	733	2251	654
2025-04-24 14:56:43	5	Human	0.858445108	781	562	983	1212	882	887
2025-04-24 14:56:43	5	Mug	0.806096494	1455	1003	1520	1054	1487	1028
2025-04-24 14:56:43	5	Human	0.732377708	706	1406	986	1908	846	1657
2025-04-24 14:56:43	5	MainFrame	0.652079403	1722	463	2037	808	1879	635
2025-04-24 14:56:43	5	Human	0.608000994	2188	504	2305	879	2246	691
2025-04-24 14:56:43	5	Human	0.599672198	1605	413	1718	650	1661	531
2025-04-24 14:56:43	5	Mug	0.346140355	296	918	338	950	317	934
2025-04-24 14:56:43	5	Bottle	0.300390691	1771	505	1792	566	1781	535
2025-04-24 14:56:43	5	PhoneCall	0.742256284	787	564	936	840	861	702
2025-04-24 14:56:43	6	Phone	0.781186581	824	628	888	699	856	663
2025-04-24 14:56:43	6	Helmet	0.706436098	2207	476	2295	533	2251	504

Sheet1

Figure 19 Behavior Excel Log

5.1.3 Cheating Detection System Interface

The Cheating Detection System interface was optimized for exam monitoring scenarios where academic integrity is the primary focus:

- Expanded video display optimized for detecting subtle cheating behaviors
- Recording status indicators showing when screen recording is active
- Evidence panel displaying thumbnails of captured cheating incidents
- Playback controls for reviewing recorded evidence
- Email confirmation indicators showing alert delivery status
- Excel log preview and export controls
- Confidence score display for detected behaviors

The interface organization prioritizes rapid identification of potential academic integrity violations while maintaining comprehensive evidence collection for later review.

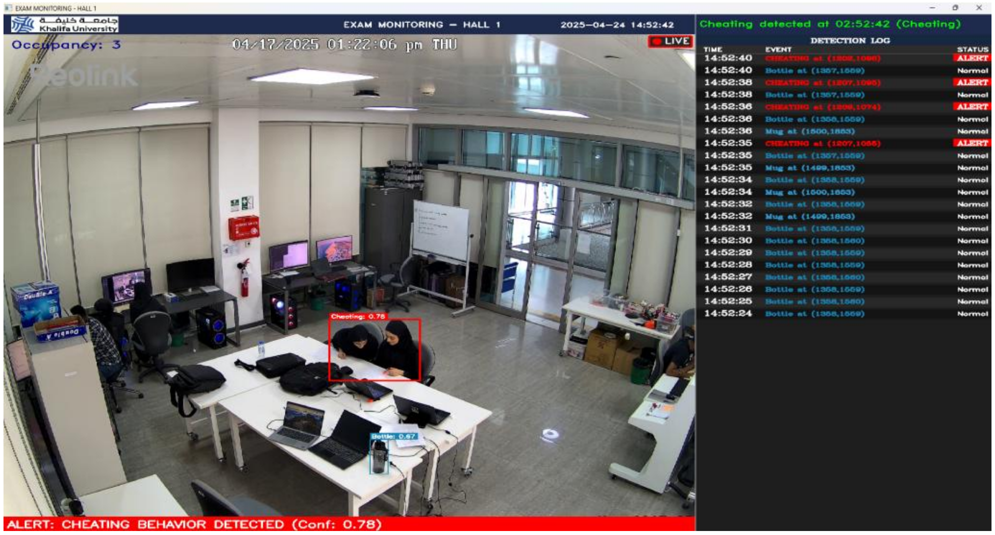


Figure 20 Cheating Detection System UI

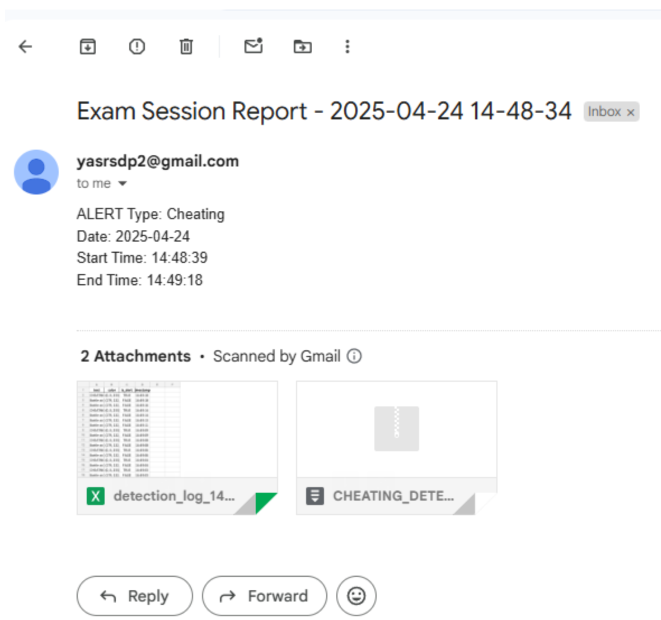


Figure 21 Cheating Detection System Email Alert

	A	B	C	D
1	text	color	is_alert	timestamp
2	CHEATING at (1232,1077)	(0, 0, 255)	TRUE	14:54:47
3	Bottle at (1359,1560)	(179, 132, 25)	FALSE	14:54:47
4	Bottle at (1358,1560)	(179, 132, 25)	FALSE	14:54:45
5	CHEATING at (1221,1083)	(0, 0, 255)	TRUE	14:54:43
6	Bottle at (1359,1560)	(179, 132, 25)	FALSE	14:54:43
7	Bottle at (1358,1560)	(179, 132, 25)	FALSE	14:54:42
8	Bottle at (1358,1558)	(179, 132, 25)	FALSE	14:54:40
9	CHEATING at (1216,1089)	(0, 0, 255)	TRUE	14:54:38
10	Bottle at (1358,1558)	(179, 132, 25)	FALSE	14:54:38
11	CHEATING at (1216,1084)	(0, 0, 255)	TRUE	14:54:36
12	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:36
13	CHEATING at (1218,1076)	(0, 0, 255)	TRUE	14:54:34
14	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:34
15	CHEATING at (1209,1071)	(0, 0, 255)	TRUE	14:54:32
16	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:32
17	CHEATING at (1207,1072)	(0, 0, 255)	TRUE	14:54:30
18	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:30
19	CHEATING at (1211,1072)	(0, 0, 255)	TRUE	14:54:28
20	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:28
21	CHEATING at (1209,1083)	(0, 0, 255)	TRUE	14:54:26
22	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:26
23	CHEATING at (1209,1097)	(0, 0, 255)	TRUE	14:54:25
24	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:25
25	CHEATING at (1207,1097)	(0, 0, 255)	TRUE	14:54:23
26	Bottle at (1357,1559)	(179, 132, 25)	FALSE	14:54:23
27	CHEATING at (1202,1098)	(0, 0, 255)	TRUE	14:54:21
28	Bottle at (1357,1559)	(179, 132, 25)	FALSE	14:54:21
29	CHEATING at (1207,1095)	(0, 0, 255)	TRUE	14:54:20
30	Bottle at (1357,1559)	(179, 132, 25)	FALSE	14:54:20
31	CHEATING at (1209,1074)	(0, 0, 255)	TRUE	14:54:18
32	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:18
33	Mug at (1500,1853)	(204, 153, 51)	FALSE	14:54:18
34	CHEATING at (1207,1055)	(0, 0, 255)	TRUE	14:54:17
35	Bottle at (1357,1559)	(179, 132, 25)	FALSE	14:54:17
36	Mug at (1499,1853)	(204, 153, 51)	FALSE	14:54:17
37	Bottle at (1358,1559)	(179, 132, 25)	FALSE	14:54:15

Sheet1

Figure 22 Cheating Excel Log

5.1.4 Attendance System Interface

The Attendance System interface was designed to efficiently track and document classroom attendance. Key design elements include:

- Real-time video display with person detection highlighted by green bounding boxes
- Prominent occupancy counter showing current number of students in the room
- University branding and course identification elements
- Live status indicator and timestamp showing monitoring is active
- Video playback controls for reviewing attendance patterns
- Automated attendance report generation with session details
- Maximum occupancy tracking with precise timestamps
- Analytics access for reviewing historical attendance data

The interface organization prioritizes efficient tracking of classroom occupancy while maintaining comprehensive attendance records for administrative review.

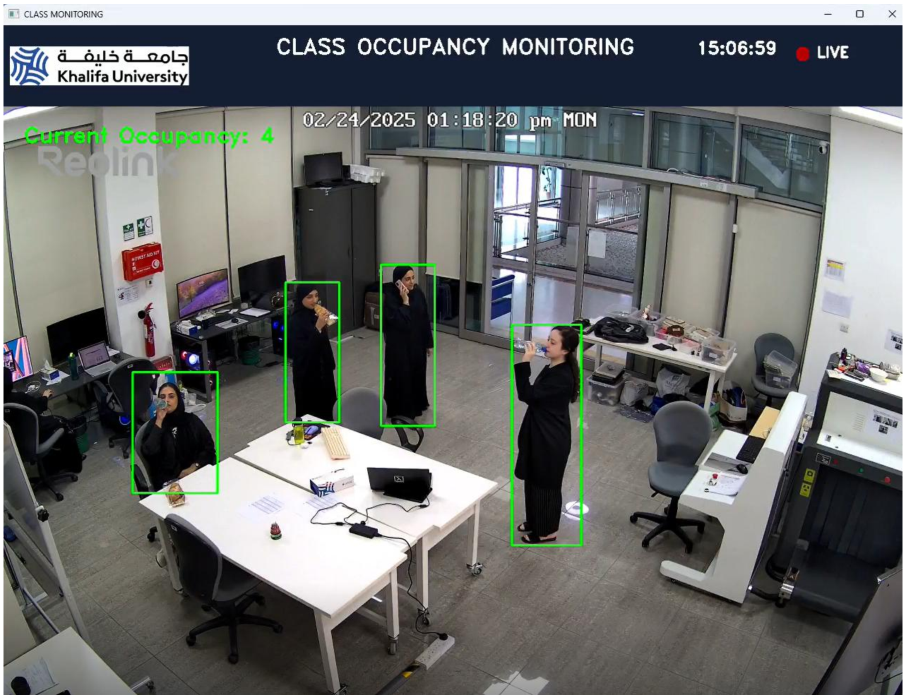


Figure 23 Attendance System UI

Figure 24 Attendance System Slip

5.2 Alert System Design

The alert system was designed to provide immediate notification of critical incidents while maintaining a comprehensive record of all detected events. The system implements different alert mechanisms based on the type and severity of detected behaviors.

5.2.1 Safety Violation Alert Mechanism

For the Behavior Monitoring System, alerts are primarily triggered when an individual enters a restricted area without proper safety equipment. The alert mechanism functions as follows:

1. When a person is detected entering the MainFrame restricted area, the system immediately checks for the presence of required safety equipment (helmet and safety jacket).
2. If either or both safety items are missing, the system:
   - Captures a screenshot of the violation
   - Highlights the individual in the interface with a red bounding box
   - Displays a warning message specifying which safety equipment is missing
   - Triggers an email alert process
3. The alert panel within the interface updates to show:
   - Timestamp of the violation
   - Specific safety equipment status (e.g., "Helmet: No, Safety Jacket: Yes")
   - A thumbnail of the captured screenshot
   - Email delivery status

This design ensures that safety officers receive immediate notification of compliance issues while maintaining a visual record within the application itself.

5.2.2 Cheating Detection Alert Mechanism

For the Cheating Detection System, the alert process is designed to capture comprehensive evidence of academic dishonesty:

1. When a cheating behavior is detected with sufficient confidence:
   - The system immediately begins recording a video clip of the incident
   - A red indicator appears in the interface highlighting the detected behavior
   - The recording continues for a configurable duration (default: 15 seconds)
   - An Excel log entry is generated with detailed information about the incident
2. The alert panel updates to show:
   - The type of cheating behavior detected
   - The confidence score of the detection
   - A preview of the recording
   - The status of the email notification process

This multi-faceted approach ensures that exam proctors receive both immediate notification and sufficient evidence to take appropriate action.

5.3 Email Notification System

The email notification system was designed to provide timely alerts to relevant personnel while including all necessary evidence and context. Different email templates and content were designed for each system to address their specific monitoring priorities.

5.3.1 Safety Violation Email Alerts

Email alerts for safety violations were designed with the following components:

1. **Subject Line**: Clear indication of the type of violation (e.g., "ALERT: Safety Equipment Violation - MainFrame Area")

2. **Body Content**:

   - Timestamp and location of the violation
   - Specific missing safety equipment (e.g., "Helmet: No, Safety Jacket: Yes")
   - Brief description of the violation context
   - Link to access the monitoring system directly
3. **Attachments**:

   - Screenshot of the violation showing the individual without proper equipment
   - Excel log export with recent violation history

The email design prioritizes clarity and immediate understanding of the safety issue, allowing security personnel to respond quickly and appropriately.

5.3.2 Cheating Detection Email Alerts

Email alerts for detected cheating incidents contain more comprehensive evidence packages:

1. **Subject Line**: Clear indication of the type of cheating detected (e.g., "ALERT: Potential Cheating Incident - Phone Usage")

2. **Body Content**:
   - Timestamp and exam/course information
   - Type of cheating behavior detected
   - Confidence score of the detection

  - Brief description of the incident
  - Instructions for reviewing the attached evidence

3. **Attachments**:
   - Screen recording of the detected incident
   - Excel log with detailed detection information
   - Additional context screenshots if available

This comprehensive package provides exam proctors and academic integrity officers with all necessary evidence in a single communication, facilitating appropriate follow-up.

5.3.3 Email Delivery Confirmation

Both systems include email delivery confirmation to ensure that alerts are successfully transmitted:

1. The interface displays an "Email Sent" confirmation when alerts are successfully delivered
2. If email delivery fails, a warning is displayed with options to retry
3. A log of all sent emails is maintained within the system for verification

This feedback loop ensures that users are aware of the notification status and can take alternative action if email delivery is compromised.

5.4 Design Evolution and User Feedback

5.4.1 User Feedback

The initial interface concept featured a single, comprehensive system with multiple tabs for different monitoring functions. However, early user testing revealed several limitations:

1. Users found it difficult to switch contexts between behavior monitoring and cheating detection
2. The combined interface created unnecessary complexity for users who only needed one function
3. Alert settings appropriate for one context were not always suitable for the other

Based on this feedback, we evolved the design to the current two-system approach with a selection hub. This separation allowed us to:

1. Optimize each interface for its specific use case
2. Simplify the user experience by removing irrelevant controls
3. Implement targeted alert mechanisms appropriate to each context
4. Develop specialized email templates for different violation types

User testing of the revised design showed significant improvements in task completion time and user satisfaction, confirming the effectiveness of the approach the team took, which is described in the section below.

5.4.2 Design Evolution

The interface design evolved significantly across the project lifecycle, guided by both user testing and Human-Computer Interactions (HCI) principles.

Phase 1 (SDP I)

In the initial phase of the project, the system began as a single, basic Tkinter window with minimal UI elements. In this interface, there were no email alerts or customizable options.

Figure 25 Showcasing the Simplicity of the Original Interface

Phase 2 (Early SDP II):

As the team progressed into the early phases of SDP II, the design of the user interface began demonstration improvements. In this version, multiple bounding boxes were drawn on-screen to indicate detected classes. However, due to one individual having multiple detections, multiple bounding boxes can be seen as shown in the figure below.

The design lacked color coding and visual hierarchy which is essential. The team acknowledged that users could potentially face confusion when interpreting the scene. Additionally, the side panel on the right displayed raw detection data, however urgency indicators were not clearly provided. These limitations could significantly increase cognitive load and introduce the risk of overlooked violations in real-time monitoring scenarios.

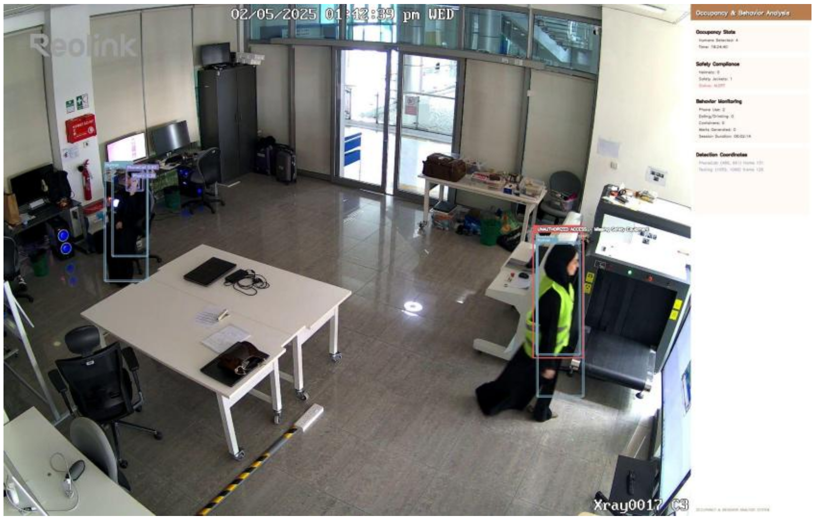


Figure 26 Initial Interface Update

Phase 3 (Late SDP II):

The figure below represents the third major version of the system interface, developed during the later part of SDP II. This version marked a significant shift in both usability and functionality compared to earlier designs.

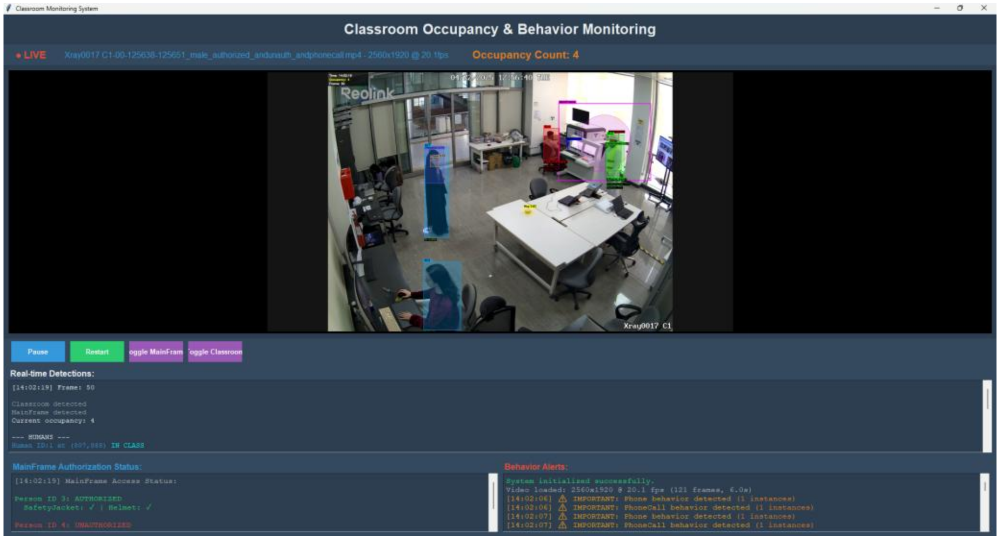


Figure 27 Later Interface Update

Several critical enhancements were introduced, and key improvements in this phase include:

- Color-Coded Bounding Boxes: Each detected class was assigned a unique color - for example, phone in orange, human in blue, and so on. This allows users to distinguish between overlapping classes with a single glance. This reduced visual ambiguity and reduced perceptual grouping.
- Live Occupancy and Detection Counters: A real-time occupancy count was added to the top, and detailed logs showing real-time detections were moved to the bottom instead of the top right. These indicators helped users maintain situational awareness without needing to scan individual bounding boxes.
- Alert Panel with Urgency Tags: Behavior alerts was added to the lower-right corner, featuring time-stamp notifications for key behaviors. Alerts were provided with urgency tags, guiding user attention toward critical incidents.
- MainFrame Authorization Status: Located on the lower-left corner, this logic-driven access control summary checked whether each individual entered a restricted zone without the required safety equipment.
- Task-Centric Layout: Buttons for toggling classroom zones and pausing video streams were added.

Phase 4 (Final):

The final iteration of the interface maintained the core functional improvements introduced in Phase 3, while incorporating a more polished and professional visual representation. The key focus in this phase was on visual refinement, institutional branding, and improved alignment with the user environment.

As shown in the figure below, the background was updated to a clean, light-gray color scheme, replacing the darker theme from earlier phases. This change was based on considerations regarding visual strain. The team agreed that a darker palette could be visually straining, especially over long monitoring sessions. These finishing touches created a visually coherent, professionally branded, and user-friendly system, ready for deployment.

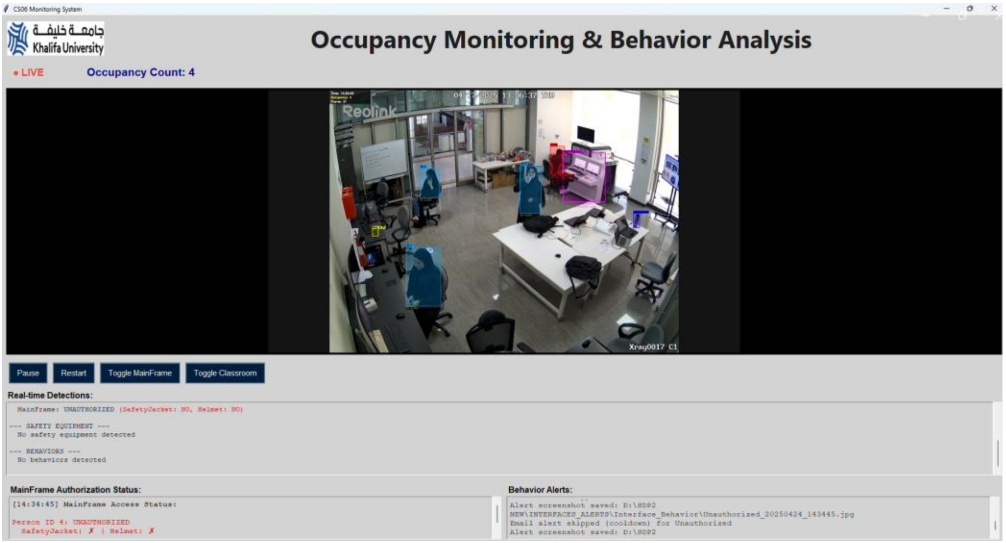


Figure 28 Final Interface

5.5 Alert Threshold Configuration

A critical aspect of the design was determining appropriate thresholds for triggering alerts. Setting these thresholds required balancing sensitivity (catching all violations) against specificity (avoiding false alarms).

5.5.1 Safety Violation Alert Thresholds

For the Behavior Monitoring System, the team implemented the following threshold configurations:

1. **Restricted Area Detection**: An alert is triggered when a person's bounding box overlaps with the MainFrame restricted area by at least 50% (configurable in settings)

2. **Safety Equipment Detection**:

    - Helmet detection requires a confidence score of at least 70%
    - Safety jacket detection requires a confidence score of at least 75%
    - Equipment is considered missing when confidence falls below these thresholds
3. **Confirmation Requirement**: A violation must be detected in at least 3 consecutive frames before triggering an alert to avoid false positives due to momentary detection errors

These thresholds were established through extensive testing in laboratory conditions and refined based on real-world performance.

5.5.2 Cheating Detection Alert Thresholds

For the Cheating Detection System, the team implemented more stringent thresholds due to the serious nature of academic integrity violations:

1. **Cheating Behavior Detection**:
   - Phone usage requires a confidence score of at least 85%
   - Looking at another student's paper requires a confidence score of at least 80%
   - Accessing unauthorized materials requires a confidence score of at least 85%
2. **Duration Requirement**: Behaviors must be detected for at least 2 seconds before triggering recording and alerts
3. **Confirmation Frames**: At least 5 consecutive frames must show the detected behavior before an alert is issued

These higher thresholds reduce the likelihood of false accusations while still capturing genuine violations effectively.

Chapter 6 Validation

6.1 Evaluation Metrics

Evaluation metrics provide a comprehensive overview of a model's performance across various object detection and behavior recognition tasks. These metrics assess the system's ability to accurately detect and classify both static objects and dynamic human behaviors, while minimizing false positives and false negatives. Key indicators such as precision, recall, and mean Average Precision (mAP) offer a quantitative evaluation of the model's effectiveness. Confusion matrices further break down performance by class, identifying specific areas for improvement. Additionally, inference speed and real-time responsiveness validate the system's ability to operate efficiently under live classroom conditions, ensuring it meets the project’s performance objectives [16].

In SDP I, the system was evaluated primarily for object detection using a single YOLOv8 model, whereas in SDP II, two specialized YOLOv8 models were trained and integrated, one focused on object detection, and the other on recognizing safety gear, behaviors such as phone use, eating, and cheating. This expanded evaluation framework required refined metric tracking across more complex classes.

1. **Precision and Recall:** Precision and recall are fundamental metrics used to measure the system’s detection accuracy. Precision evaluates the proportion of true positive detections among all positive predictions, while recall assesses the proportion of true positive

detections among all actual positives. These metrics are particularly critical in reducing false alarms and ensuring that all relevant detections such as humans in restricted zones or students using mobile phones are correctly flagged [17]. In SDP II, despite the greater challenge posed by behavior detection, the system achieved high precision for classes like 'Helmet' and 'Cheating' at high confidence thresholds, while maintaining acceptable recall across varied behavior classes.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Equation 1 Precision & Recall

2. **Mean Average Precision (mAP):** This metric evaluates the model's performance across all object or behavior categories by averaging the precision-recall curve for each class. mAP remains a key metric in multi-class detection tasks, offering a holistic view of detection effectiveness [18]. For SDP II, mAP@0.5 was recorded at 0.49, while mAP@0.5:0.95 reached 0.30 highlighting solid performance across a broader and more complex set of 15 classes. High mAP values were observed for safety such as 'Helmet' (0.975) and 'SafetyJacket' (0.992), while behavior classes such as 'Eating' and 'Texting' demonstrated more variation due to their ambiguous and visually subtle nature.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Equation 2 Mean Average Precision

Where N is the number of categories and APi is the average precision for category i. The difference between the mAP@0.5 and mAP@0.5:0.95 lies in the Intersection over Union (IoU) thresholds, with the former calculated at a fixed 50% overlap and the latter averaged across thresholds from 0.50 to 0.95. This distinction reveals the model’s robustness in handling detection under varying levels of object overlap.

3. **Confusion Matrix:** Confusion matrices offer a detailed breakdown of classification performance for each class, clearly visualizing true positives, false positives, false negatives, and true negatives. These matrices help identify underperforming or overlapping classes such as “Cup” and “Mug” in SDP I, or “PhoneCall” and “Drinking” in SDP II and guide refinements like class weighting, data augmentation, or threshold adjustments [19].

Metric	Description
True Positive TP	An instance for which both predicted and actual values are positive.
False Positive FP	An instance for which predicted value is positive but actual value is negative.
False Negative FN	An instance for which predicted value is negative but actual value is positive.
True Negative TN	An instance for which both predicted and actual values are negative.

Table 2 Confusion Matrix

4. **Detection Visualization:** In SDP I, object detection visualizations showcased accurate identification of entities such as “Human,” “Cup,” “Mug,” and the “MainFrame” restricted area, each labeled with confidence scores. These visualizations supported system interpretability and alert validation.

   In SDP II, detection expanded to include both objects and behaviors, with visual overlays used to highlight detected actions (e.g., “Texting,” “Cheating”) and items (e.g., “Helmet,” “Bottle”). Real-time detection displays provided user-friendly monitoring capabilities for proctors. Screenshots were automatically saved when violations occurred, offering clear and organized visual evidence to support post-incident analysis and reporting. The system also generated frame-by-frame CSV logs detailing object locations, detected class names, and confidence levels enhancing post-event review and system accountability.

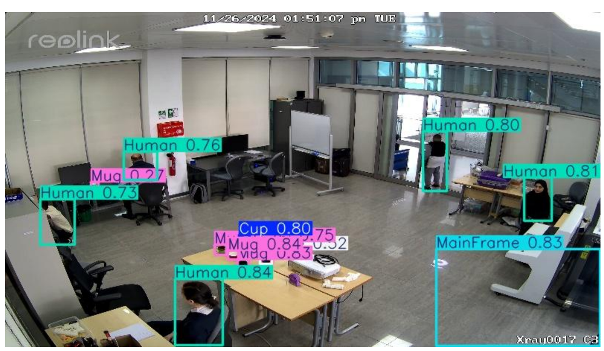


Figure 29 Object Detection Visualization SDP I

6.1.1 Verification

Verification is the process of ensuring that the system has been implemented in strict accordance with the predefined design specifications and functional requirements outlined during the planning phase. In this project, verification was essential to confirm that each system component—ranging from object detection and behavior recognition to real-time alerting and structured data logging—was correctly constructed and operated as intended.

SDP I Verification:

The primary objective in SDP I was to develop a reliable system capable of detecting physical objects within classroom environments. Verification involves testing the YOLOv8-based model across multiple scenarios using pre-recorded video footage. The system was expected to accurately detect key entities such as 'Human', 'Cup', 'Mug', 'Bottle', and to monitor activity in the restricted 'MainFrame' zone. Through systematic trials, the model successfully displayed labeled bounding boxes for these objects, with detection outcomes closely aligned with expectations. The alert mechanism reliably triggered when a person entered the restricted zone, and the confidence scores for object detections were consistent with training-phase performance. These results validated that the implemented features met the original project requirements and confirmed the correct construction of the object detection system.

SDP II Verification:

In the second development phase, the system architecture was enhanced by integrating two independently trained YOLOv8 models, one dedicated to object detection and another to behavior recognition. Verification in SDP II extended beyond evaluating object detection accuracy to include the system's ability to perform robustly under both live and recorded classroom conditions. Core functional requirements involved detecting behaviors such as phone usage, eating, drinking, and cheating, in addition to safety-related items like helmets and safety jackets. The system was subjected to a diverse set of test scenarios to ensure that each detected violation appropriately triggered alerts, screenshots were automatically saved, and detection logs were accurately recorded in structured CSV files. The graphical user interface and mode-switching features were also verified to confirm seamless operation in both 'Normal' and 'Exam' environments. Collectively, these verifications demonstrated that all functional components including behavioral alerts, screenshot captures, and detection logging operated reliably, thereby validating that the SDP II system satisfied its expanded design requirements for real-time classroom behavior monitoring.

6.1.2 Validation

Validation is the process of determining whether the constructed system effectively achieves its intended purpose in realistic, real-world conditions. This phase goes beyond ensuring the system was built correctly (verification) and assesses its ability to perform successfully in practical environments. In this project, validation focused on evaluating the system's capacity to accurately detect objects and recognize behaviors in dynamic classroom settings. The system was tested using live video streams and diverse pre-recorded scenarios to ensure it could generalize well across different conditions. By analyzing detection accuracy, consistency, and the system's ability to respond to real-time events, the validation process confirmed that the system met its goals of providing reliable, real-time monitoring and alerts, making it suitable for deployment in real educational environments.

SDP I Validation:

During the SDP I phase, validation efforts focused on assessing the system's reliability in detecting objects commonly encountered in classroom environments. The model was tested on both training data and previously unseen test data to evaluate its generalization capabilities. Visual inspections, along with key detection metrics such as mean Average Precision (mAP) and precision-recall curves, were employed to confirm that the system accurately identified objects like 'Human', 'Cup', 'Mug', and the restricted 'MainFrame'. The system consistently detected these objects with high confidence, and confusion matrices revealed minimal misclassification between classes. Performance was particularly strong for larger, more defined objects, such as 'MainFrame' and 'Human'. By successfully identifying these critical objects and triggering alerts for restricted zone violations, the system met its validation objectives.

SDP II Validation:

The SDP II phase introduced a more complex validation process, as the system was now required to recognize not only physical objects but also human behaviors in real-time. This included behaviors such as cheating, phone usage, eating, drinking, and wearing safety gear. Validation tests were conducted using both live video streams and a variety of pre-recorded classroom scenarios. Metrics like mAP@0.5 (which reached 0.49) and class-specific F1 scores were used to assess detection accuracy. The system demonstrated reliable performance in detecting and distinguishing key behaviors such as 'Cheating' and 'Helmet' usage, both of which achieved an F1 score of 1.00. However, more challenging behaviors, such as 'Eating' and 'PhoneCall', resulted in slightly lower performance scores. The system's overall behavior recognition capabilities were validated through real-time alert generation, accurate class labeling, and correct screenshot and CSV logging.

The validation process confirmed that SDP II effectively met its intended monitoring goals. The system demonstrated reliable, responsive, behavior-aware classroom monitoring, producing consistent outputs across different real-world conditions.

6.1.3 Evaluation

Evaluation is a critical step in assessing how well the trained model performs in both training and testing phases. It ensures that the model has not only learned effectively but also that it can generalize well to new, unseen data. This section outlines the evaluation processes carried out to measure the model's performance and its ability to accurately detect and classify various classes.

6.1.3.1 Training Evaluation

The evaluation of the model's performance during the training phase is crucial for assessing its ability to generalize to unseen data and its proficiency in accurately classifying classes. To comprehensively measure performance, several metrics were utilized, including precision, recall, and F1-score, which offer a balanced view of classification accuracy and error rates. Additionally, confusion matrices were employed to visualize class-wise performance, highlighting areas of strength and identifying any potential weaknesses in the model. These evaluation methods provided valuable insights into the model's learning process and its effectiveness in detecting and classifying objects in various scenarios.

SDP I Confusion Matrix:

Figure 30 Confusion Matrix After Training YOLOv8s Model

Confusion Matrix Analysis

The confusion matrix, as shown in figure 17 above, offers a class-wise breakdown of predictions, highlighting True Positives (correct predictions) on the diagonal and misclassifications in the off-diagonal values. Below are the observations based on the raw and normalized confusion matrices:

1. **Cup Class**:
   - Correct Predictions: 490 True Positives.
   - Misclassifications: 175 instances were labeled as "Background," indicating the model occasionally struggles to distinguish cups in cluttered or ambiguous scenes.

2. **MainFrame Class**:
   - Excellent performance with **100% accuracy**. All 23 instances were correctly identified, demonstrating the model's ability to detect restricted areas reliably.
3. **Bottle Class**:
   - Correct Predictions: 254 instances.
   - Misclassifications: 80 instances were categorized as "Background," suggesting the need for further augmentation to distinguish bottles from background objects in complex settings.
4. **Human Class**:
   - Strong performance with minimal misclassifications. This indicates that the model is well-trained to detect humans even in challenging environments.
5. **Mug Class**:
   - Correct Predictions: 246 instances.
   - Misclassifications: 57 labeled as "Background," highlighting the need for additional training or augmentation to handle variations in mug appearances or settings effectively.

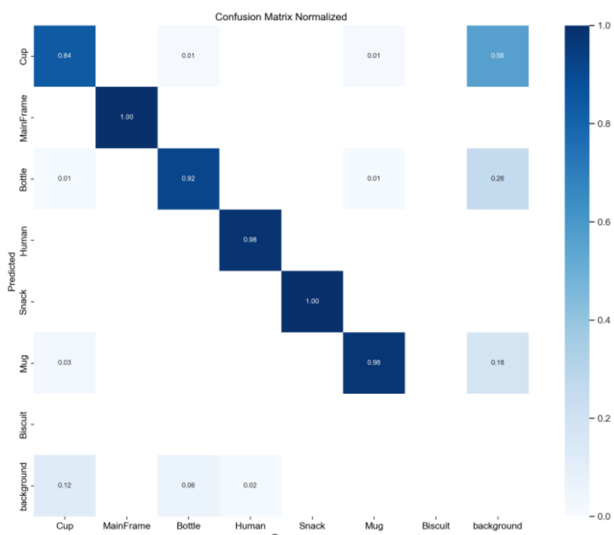


Figure 31 Normalized Confusion Matrix

Normalized Confusion Matrix

The normalized confusion matrix in Figure 18, which displays the proportion of predictions rather than raw counts, provides a more intuitive understanding of the model’s reliability for each class. For instance:

- **MainFrame Class**: A perfect score of **1.00**, confirming the model's consistency for restricted area detection.
- **Cup Class**: Despite achieving a precision of 0.84, the model showed some confusion with the background (0.56 misclassification rate), necessitating further refinement.
- **Mug Class**: Precision of 0.98, although occasional overlap with the background exists.

SDP II Confusion Matrix:

For SDP II, two distinct YOLOv8 models were trained, one to detect objects such as Cup and safety such as SafetyJacket, Helmet, and another tailored to behavioral actions including PhoneCall, Texting, and Cheating. The confusion matrices captured for SDP II illustrate the model's classification performance across both object and behavior-based classes.

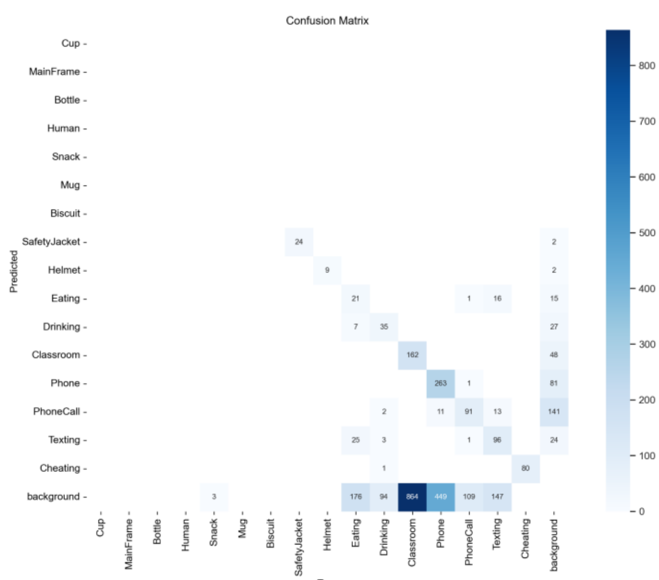


Figure 32 Confusion Matrix After Training YOLOv8 Models

Confusion Matrix Analysis:

- **SafetyJacket and Helmet:** Achieved perfect classification accuracy: All 24 SafetyJacket and 9 Helmet instances were correctly identified with no misclassification.
- **Cheating:** High precision with 80 true positive detections and no false positives or negatives, resulting in a normalized accuracy of 1.00.
- **Texting:** Identified 96 instances correctly but misclassified 13 as PhoneCall and 24 as Background.
- **PhoneCall:** Detected 91 true positives; however, 13 were predicted as Texting and 11 as Phone.
- **Cup, Mug, Eating, and Drinking:** Detection was modest: Mug (48 TP), Eating (21 TP), Drinking (35 TP), with frequent confusion mainly involving the Background class.

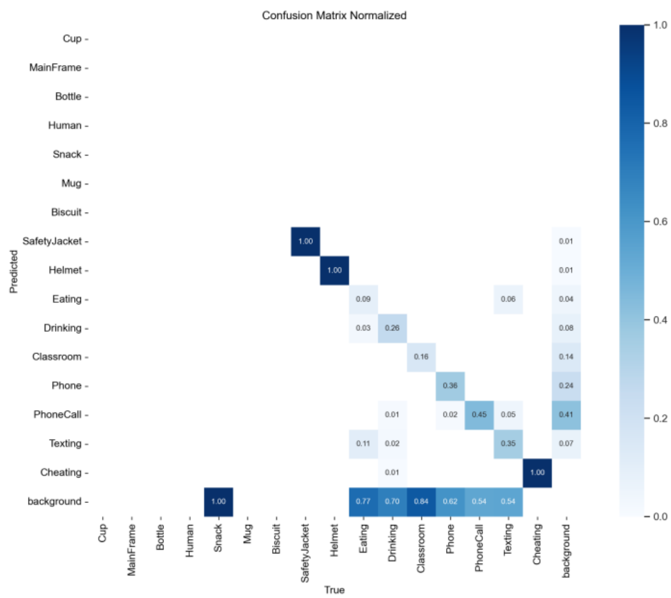


Figure 33 Normalized Confusion Matrix

Normalized Confusion Matrix Insights:

- **Helmet and SafetyJacket:** Achieved perfect scores (1.00), confirming complete prediction accuracy.
- **Cheating:** Also scored a normalized accuracy of 1.00, showcasing robust behavioral detection.
- **Texting and PhoneCall:** Displayed normalized accuracies of 0.35 and 0.45, respectively, with noticeable class overlap.

- **Eating and Drinking:** Encountered classification challenges due to visual ambiguity with the Background category, reflected in lower normalized precision: Eating (0.09), Drinking (0.26).

Observations and Recommendations:

- **Overall Performance:** Across both SDP I and SDP II, the models demonstrated strong detection capabilities for clear and easily recognizable classes. In SDP I, objects like Human and MainFrame consistently achieved high precision and recall. Similarly, in SDP II, classes such as SafetyJacket, Helmet, and Cheating showed excellent accuracy with minimal misclassifications, reflecting the effectiveness of the specialized YOLOv8 models.
- **Challenging Classes:** Smaller or visually similar objects like Cup and Mug in SDP I and behaviorally nuanced actions such as PhoneCall and Texting in SDP II exhibited overlaps and confusion. These challenges underline the need for further data diversification and more refined visual cues in the training datasets to improve class separability.
- **Confusion with Background:** A common trend observed in both stages was the misclassification of subtle objects and behaviors as background. In SDP I, Cup, Bottle, and Mug occasionally blended into cluttered scenes, while in SDP II, behavioral classes like Eating and Drinking were often mistaken for background elements. This indicates a strong need for improved annotation practices and targeted augmentation strategies such as varying lighting, angles, and occlusion scenarios to enhance detection reliability in realistic environments.

Precision-Confidence Curve Analysis

This curve illustrates how the model's precision—its ability to make accurate detections—varies with different confidence thresholds. It helps us evaluate how trustworthy the system’s predictions are as confidence increases.

SDP I P_curve :

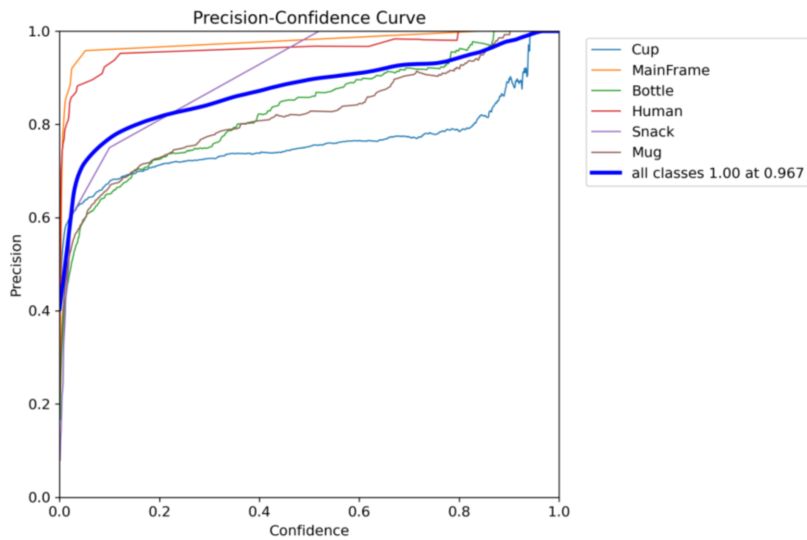


Figure 34 Precision-Confidence Curve

Key Observations:

1. **High Precision at Elevated Confidence Thresholds**:
   - At confidence thresholds approaching 1.0, the precision for all classes converges to nearly 100%, indicating the model's reliability in its high-confidence predictions.
   - This demonstrates that when the model is confident in its detections, its predictions are almost always accurate.
2. **Class-Specific Trends**:
   - **MainFrame and Snack**:
     - These classes exhibit consistently high precision across all confidence levels, suggesting the model is robust and less prone to false positives for these categories.
   - **Cup and Mug**:
     - Precision is lower at lower thresholds, reflecting challenges in distinguishing these classes due to similarities with other objects or the presence of false positives.
3. **Overall Model Performance**:
   - For all classes combined, precision reaches 100% at a confidence threshold of 0.967. This indicates that the model is effective in minimizing errors when predictions are made with high certainty.

SDP II P_curve :

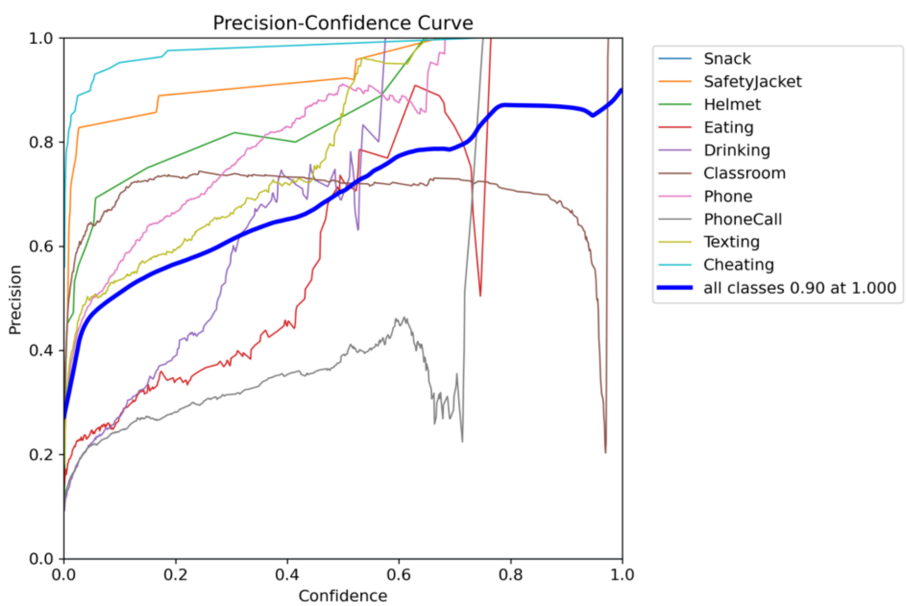


Figure 35 Precision-Confidence Curve SDP II

Key Observations:

1. **High Precision at Elevated Confidence Thresholds**:
   - As the confidence threshold approaches 1.0, the precision for most classes improves significantly.
   - At a threshold of 1.000, the combined precision across all classes reaches a peak of 90%, confirming that the system makes highly reliable predictions when it is most confident.
2. **Class-Specific Trends**:
   - Cheating, SafetyJacket, and Helmet classes achieve near-perfect precision even at lower thresholds, indicating that the system recognizes these classes with high confidence and low false positives.
   - Phone, Texting, and Drinking show moderate precision that gradually improves, meaning these classes are more difficult to distinguish at low confidence but benefit from stricter thresholds.

- PhoneCall and Eating show fluctuating precision across the range, suggesting that these behaviors are more prone to confusion and require clearer distinctions in training data.

3. **Overall Model Performance**:
    - The model's precision improves steadily across nearly all classes as the threshold increases, which demonstrates its reliability in producing accurate results when stricter decision boundaries are applied. However, the analysis also reveals certain classes that are sensitive to overlap or visual similarity, especially behavioral ones like PhoneCall and Eating.

Across both SDP I and SDP II, the precision-confidence analysis confirms that higher thresholds enhance prediction reliability by reducing false positives. However, some classes, particularly smaller objects in SDP I or overlapping behaviors in SDP II, continue to require dataset expansion, labeling improvements, or additional augmentation strategies to further elevate class-specific accuracy.

These findings underscore the importance of tailoring confidence thresholds and refining training inputs to maintain high-precision predictions in real-world scenarios involving diverse object types and human actions.

Precision-Recall Curve Analysis

The Precision-Recall (PR) curve evaluates the balance between the model's precision (how many of its predictions are correct) and recall (its ability to detect all relevant objects) across various confidence thresholds. It provides valuable insights into the model's performance for each class.

SDP I PR_curve :

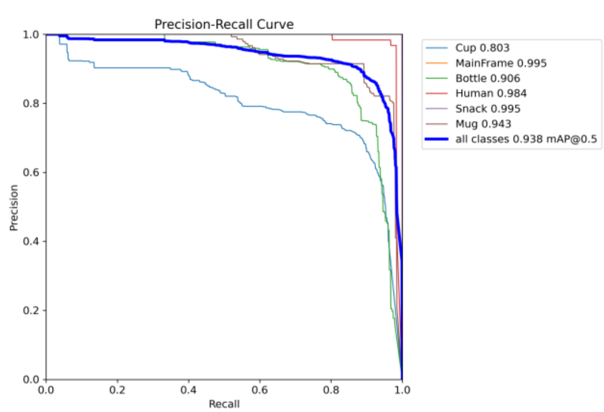


Figure 36 Precision-Recall Curve

Key Observations:

1. **Overall Performance**:
    - The model achieves a high **mean Average Precision (mAP@0.5)** of **93.83%**, indicating strong detection performance across all classes.
    - This result reflects the robustness of the model in balancing precision and recall while maintaining high detection accuracy.
2. **Class-Specific Analysis**:
    - **Bottle Class**:
        - A precision score of **0.906** demonstrates reliable detection for this class.
        - However, some decline in recall at higher confidence thresholds suggests a trade-off between missed detections and reduced false positives.
    - **Snack and MainFrame Classes**:
        - These classes exhibit near-perfect performance, with precision and recall curves tightly packed in the upper-right corner of the graph.
        - The model shows exceptional reliability in detecting these objects, even at higher confidence thresholds.
3. **Cup and Mug Classes**:
    - These classes show noticeable drops in recall at higher confidence thresholds, reflecting challenges in detecting smaller or more visually ambiguous objects.
    - The decline highlights areas where additional data augmentation or class-specific fine-tuning may improve performance.
4. **Recall Trends Across Confidence Thresholds**:
    - At lower confidence thresholds, recall for all classes approaches **1.0**, indicating the model's ability to detect most objects.
    - As the threshold increases, recall decreases for specific classes (e.g., **Cup**, **Bottle**) as the model prioritizes reducing false positives, becoming more conservative in its predictions.

SDP II PR_curve :

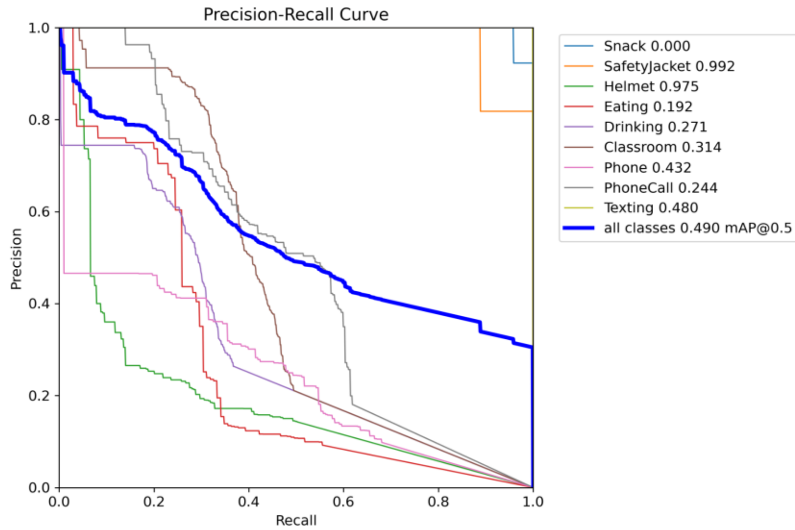


Figure 37 SDP II P-R Curve

Key Observations:

1. **Overall Performance:**

   The model achieved a mean Average Precision (mAP@0.5) of **49.0%**, a respectable result considering the complexity of classes like "PhoneCall," "Texting," and "Cheating," which involve behavior recognition rather than static object detection. This value reflects moderate overall performance but highlights the challenge of detecting subtle and overlapping behaviors in live classroom environments.

2. **Class-Specific Analysis:**

- **SafetyJacket and Helmet:** These classes demonstrated near-perfect performance, with precision-recall curves close to the upper-right corner. SafetyJacket reached a precision of **0.992**, and Helmet achieved **0.975**, indicating the system's strength in recognizing well-defined physical objects.
- **Texting and Cheating:** These behavior-based classes yielded relatively strong results for this category type. Texting reached a precision of **0.480**, and Cheating maintained stable detection across thresholds, suggesting the model effectively distinguished these activities from normal background movement.
- **PhoneCall and Eating:** Both classes showed noticeable challenges. PhoneCall achieved a precision of **0.244**, while Eating dropped to **0.192**, revealing difficulty in distinguishing

these actions from visually similar movements or background noise. These results emphasize the need for improved training samples or better-defined visual cues.

- **Drinking:** This class experienced lowest performance metrics, drinking achieved **0.271**, suggesting some improvement, though the recall curve still indicates room for refinement.

3. **Recall Trends Across Confidence Thresholds:**

   At lower thresholds, the system achieves high recall across most classes—meaning it can detect the majority of instances when being less strict. However, as confidence thresholds increase, precision improves, but recall drops, particularly for complex behavior classes like PhoneCall and Eating. This trend shows the system becomes more cautious, reducing false positives at the cost of missing some true positives.

The PR curves across both SDP I and SDP II confirm the models' strong generalization capabilities, particularly for clearly defined classes such as MainFrame, Human, SafetyJacket, and Helmet. These categories consistently exhibited high precision and recall, even at stricter confidence thresholds. However, challenges remain in detecting smaller or behavior-based classes, such as Cup, Mug, PhoneCall, and Eating, where performance declined due to misclassifications or ambiguous visual cues. These results emphasize the need for targeted improvements, including better annotation strategies, expanded behavioral datasets, and advanced augmentation techniques. Collectively, these findings offer actionable guidance for optimizing the detection system to perform more accurately across both static object recognition and dynamic behavior analysis.

Recall-Confidence Curve Analysis

The Recall-Confidence curve provides insights into the model's ability to detect all relevant instances (recall) at varying confidence thresholds. It demonstrates how the model balances between capturing as many ground truth instances as possible and avoiding false negatives.

SDP I R_curve :

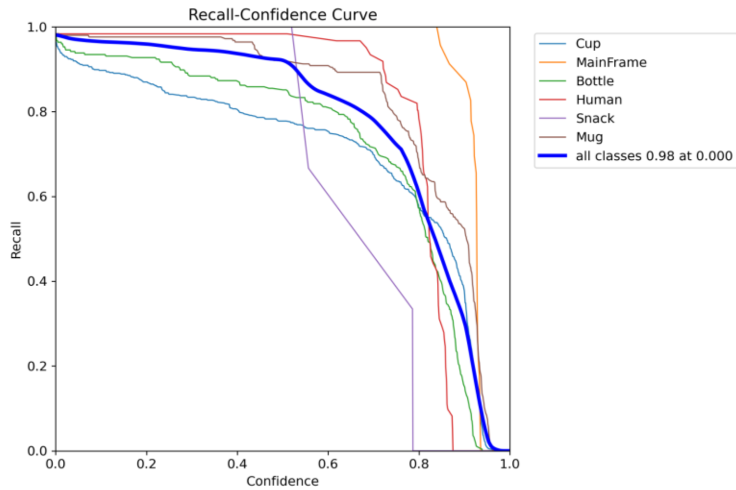


Figure 38 Recall-Confidence Curve SDP I

Key Observations:

1. **Overall Recall Performance**:
    - At lower confidence thresholds, recall for all classes approaches 1.0, indicating that the model is highly effective at capturing most objects when it is less conservative.
    - This behavior is expected, as lower thresholds allow the model to detect a larger number of instances, even at the cost of including some false positives.
2. **Class-Specific Trends**:
    - **MainFrame and Snack Classes**:
        - These classes maintain high recall across a wide range of thresholds, reflecting the model's consistent ability to detect these objects, even under stricter conditions.
    - **Cup and Bottle Classes**:
        - Recall for these classes shows a noticeable drop at higher confidence thresholds. This indicates that the model becomes more selective, which leads to some true positives being missed.
    - **Mug and Human Classes**:
        - These classes demonstrate a balanced trend, with a gradual decline in recall as the confidence threshold increases.
3. **Trade-Off with Precision**:
    - The curve illustrates the trade-off between achieving high recall and avoiding false negatives. While recall is high at lower thresholds, it may be accompanied by a reduction in precision, as the model captures more irrelevant detections.

    - Conversely, higher thresholds improve precision but reduce recall, particularly for more visually ambiguous or smaller objects like Cup and Bottle.
4. **Implications for Real-World Applications**:
    - The high recall at lower thresholds indicates the model's suitability for scenarios where missing an object is critical, such as detecting restricted areas (MainFrame) or safety-critical objects.
    - For applications requiring higher confidence, thresholds can be adjusted to optimize recall for key classes while balancing precision.

SDP II R_curve :

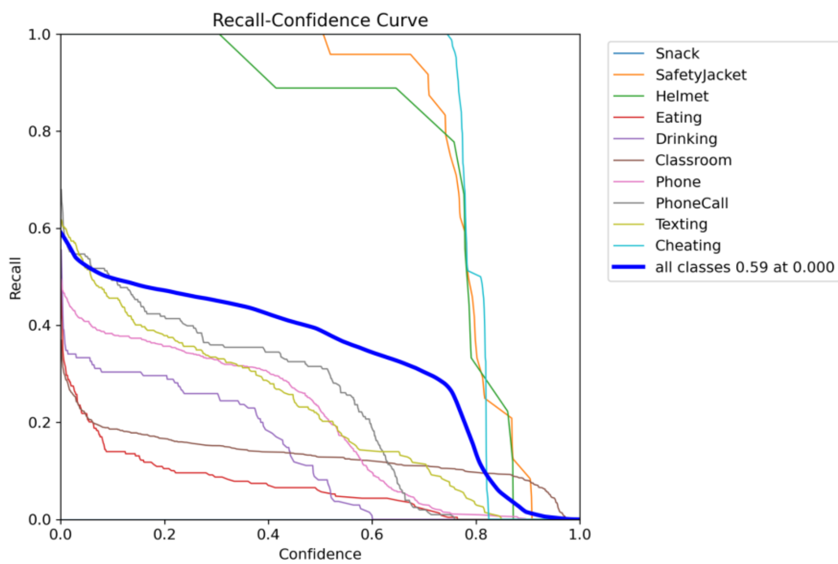


Figure 39 Recall-Confidence Curve SDP II

Key Observations:

1. **Overall Recall Performance:** At lower confidence levels, recall for most classes approaches 1.0, indicating the model's capability to identify nearly all instances when it is less strict about detection certainty. However, this comes at the cost of including more false positives.
2. **High-Recall Classes – Cheating, SafetyJacket, Helmet:** These classes achieved exceptional recall rates across most thresholds. Specifically, Cheating consistently maintains high recall, ensuring critical behavioral violations are detected promptly. SafetyJacket and Helmet also perform well, underlining the system's strength in recognizing safety equipment.

3. **Mid-Level Performance – Phone, Texting, PhoneCall:** The model demonstrates moderate recall for behavioral classes like **Phone**, **Texting**, and **PhoneCall**, especially at lower confidence values. However, their curves drop faster as confidence increases, reflecting difficulty in distinguishing between visually similar actions as the model becomes more selective.
4. **Low-Recall Classes – Eating and Drinking:** These classes show lower and more volatile recall values across the curve. This suggests that the model struggles to consistently detect these behaviors, especially when the objects involved (like food or beverages) are partially occluded or small in size.
5. **Trade-Off Insights:**

- Like in SDP I, the curve highlights the classic precision-recall trade-off: as confidence thresholds increase, false positives decrease, but some true positives are missed.
- This trade-off must be managed depending on context—for example, when detecting **Cheating**, it may be acceptable to allow a few false positives if it ensures no case goes undetected.

6. **Implications for Real-World Use:** The recall trends in SDP II show that the system is especially effective for critical actions like **Cheating** and **Safety gear detection**, making it highly suitable for monitoring controlled environments like exam rooms or labs. However, behavioral classes that rely on subtle visual cues may benefit from more focused dataset enhancement, including better annotations, lighting variation, and pose diversity.

The Recall-Confidence curve confirms that both SDP I and SDP II models demonstrate excellent detection capabilities at lower confidence thresholds. In SDP I, classes like MainFrame, and Human consistently achieved high recall, making the system highly effective in capturing key objects even under relaxed conditions. Similarly, in SDP II, the model excelled in detecting behaviors and safety-related objects such as Cheating, Helmet, and SafetyJacket, maintaining strong recall across wide thresholds. However, both systems showed challenges at higher thresholds, where classes like Cup, Bottle, Eating, and Drinking experienced noticeable recall drops. These drops highlight a need for targeted improvements, such as more diverse training data and refined augmentation techniques. Overall, this analysis emphasizes the critical role of threshold tuning based on real-world application needs whether the priority is minimizing missed detections or reducing false positives.

6.1.3.2 Testing Evaluation

Testing evaluation focuses on assessing the system's real-time performance, its ability to detect objects and behaviors accurately, and its overall effectiveness in operational environments. In both SDP I and SDP II, the system's performance was evaluated through comprehensive testing processes to ensure it met real-world demands and accurately processed live video feeds.

1. **SDP I Testing Evaluation:**

The final step involved integrating the trained YOLOv8 model with a live video feed to evaluate its real-time detection and tracking capabilities. During this phase, the model continuously detected and tracked classroom objects such as 'Cup', 'Bottle', 'Mug', and 'Human' across video frames, displaying the results in real time. This allowed the team to evaluate how accurately the system could identify objects and how efficiently it processed video frames in real time. Performance was monitored to confirm that frames were analyzed efficiently without compromising precision. The system demonstrated strong object detection results while maintaining the responsiveness necessary for real-time deployment.

A critical alert feature was included, capturing and saving screenshots whenever unauthorized access to the restricted 'MainFrame' zone was detected. When the model identified a person entering the restricted area based on the configured Intersection over Union (IoU) threshold, it automatically saved the relevant frame in the alert directory. This provided visual documentation of violations for post-event review and security audits.

Additionally, the system generated structured CSV files for each processed video. These files included frame-by-frame summaries containing detected class names, confidence scores, and bounding box coordinates, helping the team to trace detection trends and verify system accuracy. These capabilities, combined with a high mAP@0.5 of 93.83% and an overall accuracy of 76%, confirmed the model's readiness for real-world classroom environments.

Figure 40 Successfully Detected and Alerted Human in Restricted Area

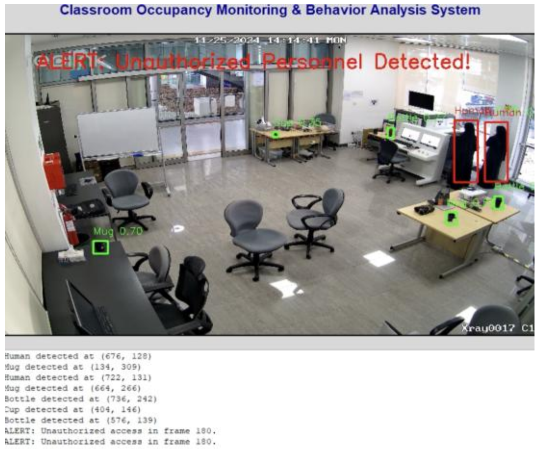


Figure 41 Successfully Identified Illegal Objects and 2 Humans in Restricted Area

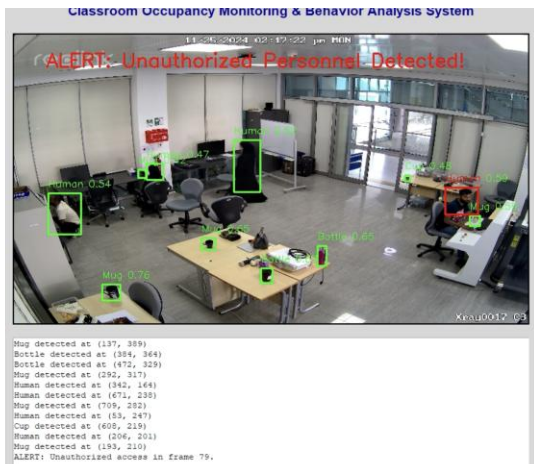


Figure 42 Showcasing the Detection Capabilities of the System

The system incorporates a critical alert feature that captures and saves screenshots whenever unauthorized access is detected in the restricted "MainFrame" area. When the model identifies a person entering the restricted zone based on Intersection over Union (IoU) thresholds, it automatically saves a frame of the event as an image in the respective Alerts/VideoX directory. This feature enhances post-event evaluation by providing visual evidence of unauthorized activities, which can be used for security audits and further investigation. The systematic organization of these alert images ensures efficient access and review of critical events, bolstering the system's monitoring capabilities.

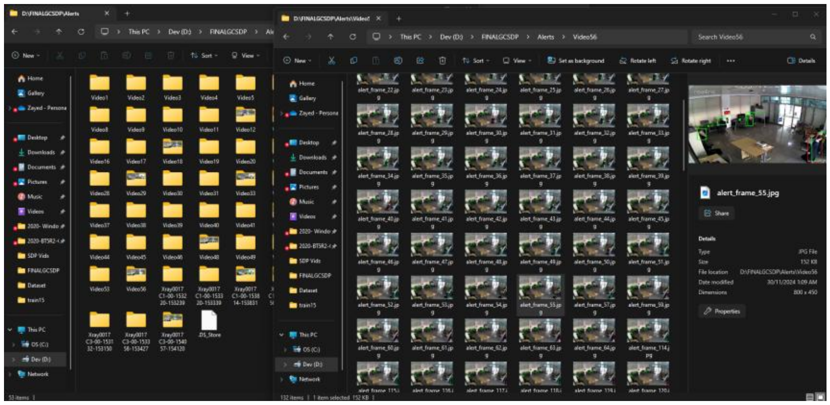
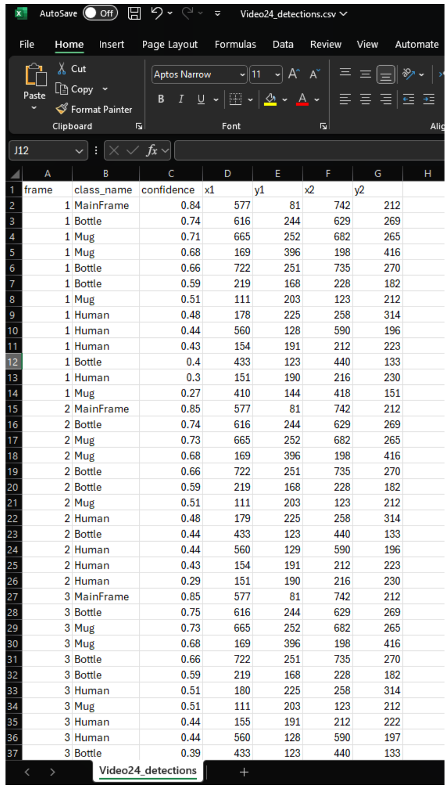


Figure 43 Alert Directory

For each processed video, the system generates a detailed CSV, as shown in the figure below, file that is automatically saved in the directory structure under runs/detect/videoX. This file provides a comprehensive, frame-by-frame summary of the detections made during real-time video analysis. The CSV contains essential information, such as the frame number, detected class names (e.g., "MainFrame," "Bottle," "Mug," etc.), confidence scores, and bounding box coordinates (x1, y1, x2, y2). These bounding box values represent the spatial location of the detected objects in the video frames. This feature allows for precise monitoring of object presence and positioning, enabling the team to analyze detection trends, verify alerts, and ensure accuracy. It serves as a valuable tool for post-analysis, debugging, and validation, offering insights into the model's performance across different classes and scenarios.

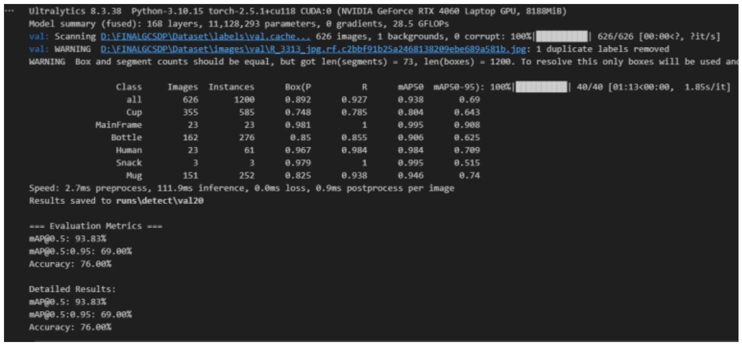


frame	class_name	confidence	x1	y1	x2	y2
1	MainFrame	0.84	577	81	742	212
1	Bottle	0.74	616	244	629	269
1	Mug	0.71	665	252	682	265
1	Mug	0.68	169	396	198	416
1	Bottle	0.66	722	251	735	270
1	Bottle	0.59	219	168	228	182
1	Mug	0.51	111	203	123	212
1	Human	0.48	178	225	258	314
1	Human	0.44	560	128	590	196
1	Human	0.43	154	191	212	223
1	Bottle	0.4	433	123	440	133
1	Human	0.3	151	190	216	230
1	Mug	0.27	410	144	418	151
2	MainFrame	0.85	577	81	742	212
2	Bottle	0.74	616	244	629	269
2	Mug	0.73	665	252	682	265
2	Mug	0.68	169	396	198	416
2	Bottle	0.66	722	251	735	270
2	Bottle	0.59	219	168	228	182
2	Mug	0.51	111	203	123	212
2	Human	0.48	179	225	258	314
2	Bottle	0.44	433	123	440	133
2	Human	0.44	560	129	590	196
2	Human	0.43	154	191	212	223
2	Human	0.29	151	190	216	230
3	MainFrame	0.85	577	81	742	212
3	Bottle	0.75	616	244	629	269
3	Mug	0.73	665	252	682	265
3	Mug	0.68	169	396	198	416
3	Bottle	0.66	722	251	735	270
3	Bottle	0.59	219	168	228	182
3	Human	0.51	180	225	258	314
3	Mug	0.51	111	203	123	212
3	Human	0.44	155	191	212	222
3	Human	0.44	560	128	590	197
3	Bottle	0.39	433	123	440	133

Figure 44 Output CSV File that Shows Confidence Rate and Coordinates of Identified Objects

Accuracy Evaluation for SDP I

The model achieved an impressive performance in its evaluation metrics, with an mAP@0.5 of 93.83% and an mAP@0.5:0.95 of 69.00%. These results highlight the model's capability to accurately detect and classify objects across various scenarios, with a strong emphasis on precision at higher thresholds. The overall accuracy was reported at 76.00%, indicating reliable predictions in real-world conditions. Each class demonstrated robust detection performance, with particularly high results for "MainFrame" and "Mug," achieving near-perfect mAP scores. The evaluation also underscored areas for improvement, such as further refining detection for smaller objects like "Cup" and "Bottle," to enhance the model's adaptability across diverse settings. The balance of precision, recall, and inference speed highlights the model's suitability for deployment in real-time object detection tasks.



```
Ultralytics 8.3.38  Python-3.10.15 torch-2.5.1+cu118 CUDA:0 (NVIDIA GeForce RTX 4060 Laptop GPU, 8188MiB)
Model summary (fused): 168 layers, 11,128,293 parameters, 0 gradients, 28.5 GFLOPs
val: Scanning D:\FINALGCSDP\Dataset\labels\val.cache... 626 images, 1 backgrounds, 0 corrupt: 100%|██████████| 626/626 [00:00<?, ?it/s]
val: WARNING  D:\FINALGCSDP\Dataset\images\val\R_3313_jpg.rf.c2bbf91b25a2468138209ebe689a581b.jpg: 1 duplicate labels removed
WARNING  Box and segment counts should be equal, but got len(segments) = 73, len(boxes) = 1200. To resolve this only boxes will be used and

                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 40/40 [01:13<00:00,  1.85s/it]
                   all        626       1200      0.892      0.927      0.938       0.69
                   Cup        355        585      0.748      0.785      0.804      0.643
             MainFrame         23         23      0.981          1      0.995      0.908
                Bottle        162        276       0.85      0.855      0.906      0.625
                 Human         23         61      0.967      0.984      0.984      0.709
                 Snack          3          3      0.979          1      0.995      0.515
                   Mug        151        252      0.825      0.938      0.946       0.74
Speed: 2.7ms preprocess, 111.9ms inference, 0.0ms loss, 0.9ms postprocess per image
Results saved to runs\detect\val20

=== Evaluation Metrics ===
mAP@0.5: 93.83%
mAP@0.5:0.95: 69.00%
Accuracy: 76.00%

Detailed Results:
mAP@0.5: 93.83%
mAP@0.5:0.95: 69.00%
Accuracy: 76.00%
```

Figure 45 Model Accuracy Evaluation - SDP I

2. **SDP II Testing Evaluation:**

In SDP II, the testing process expanded to include not only object detection but also behavior recognition in real time. The upgraded system utilized two YOLOv8 models to monitor classroom environments and identify actions such as 'Cheating', 'Phone Usage', 'Eating' ,'Drinking', and safety like helmets and safety jackets.

The system was tested on both recorded videos and live camera streams, analyzing each frame in real time. Detected violations triggered automatic alerts, with screenshots saved to organized alert directories and CSV logs capturing essential information, including class labels, confidence scores, and bounding box positions.

Testing confirmed that the system reliably identified critical behavior categories such as 'Cheating', 'Helmet', and 'SafetyJacket', all of which achieved F1-scores near or equal to 1.00. Evaluation metrics showed an mAP@0.5 of 0.49 and mAP@0.5:0.95 of 0.30, reflecting solid performance across 15 diverse classes. While behaviors like 'PhoneCall' and 'Eating' showed slightly reduced detection scores due to visual complexity, the system remained consistent and responsive under varying conditions.

This phase also validated the smooth functioning of the user interface and switching between Normal and Exam modes, ensuring that all real-time features—behavior alerts, screenshot captures, CSV logging, and automated email notifications functioned reliably during all testing scenarios.. The email system sends alerts in real time, attaching both the screenshot and the corresponding log, enabling remote monitoring and documentation of behavioral alerts. The results demonstrated that the system was capable of supporting behavior sensitive monitoring in real-world educational settings.

Figure 46 Successfully Detected PhoneCall and Safety Jacket and Helmet

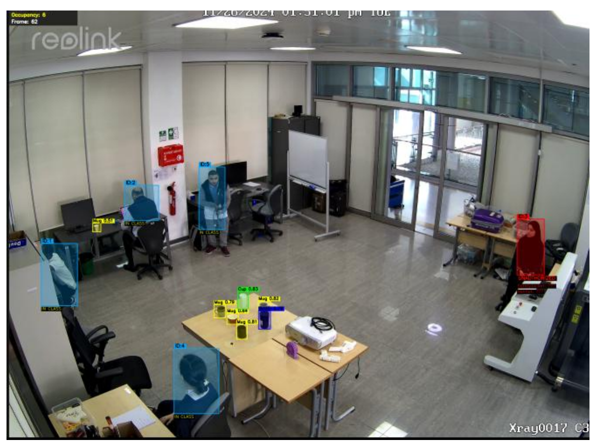


Figure 47 Successfully Detected Humans , Bottle , Mug , Cup , and Unauthorized

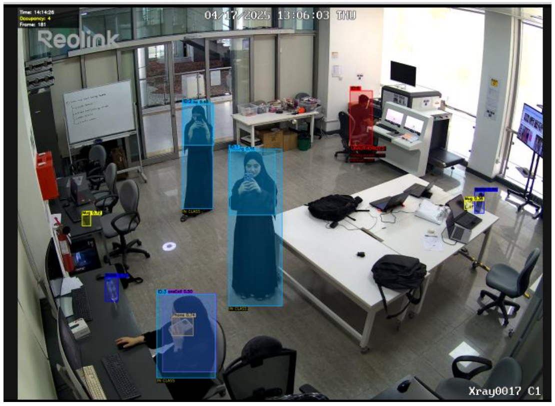


Figure 48 Successfully Detected PhoneCall and Texting

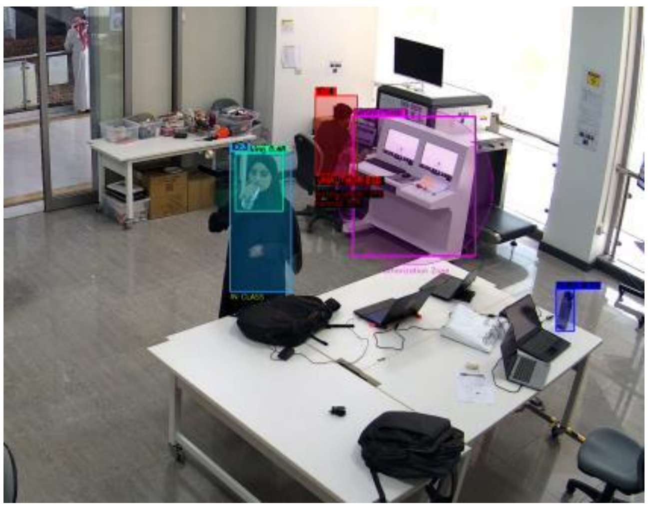


Figure 49 Successfully Detected Drinking

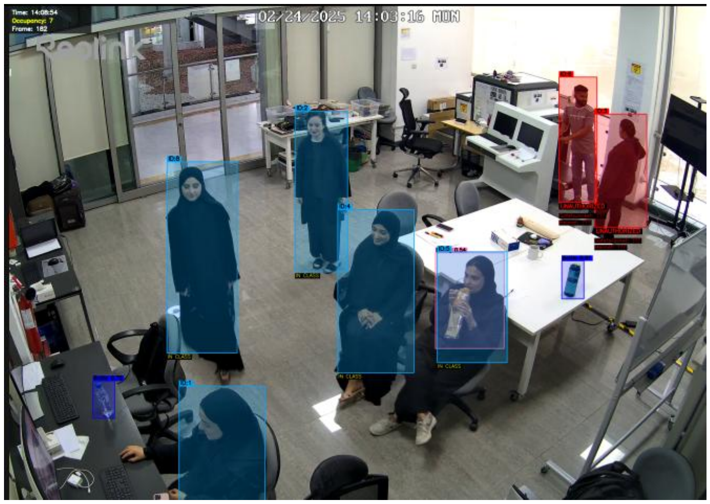


Figure 50 Successfully Detected Eating

Figure 51 Successfully Detected Cheating with Alert

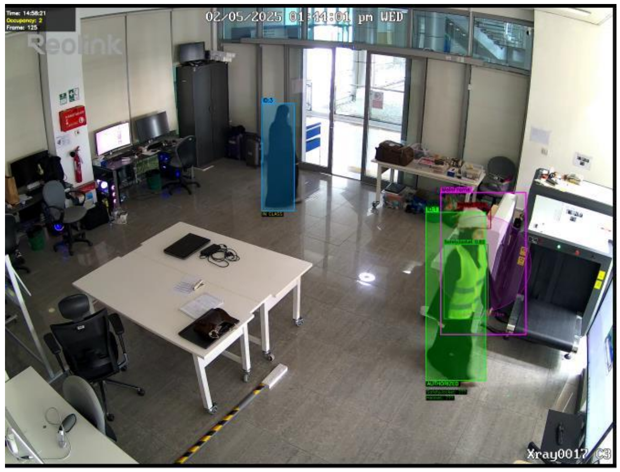
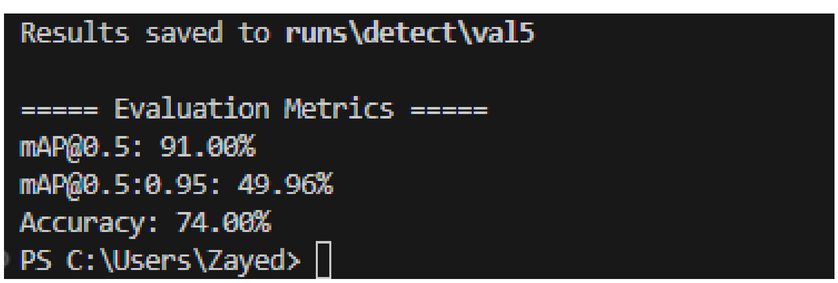


Figure 52 Successfully Detected Authorized Access

Accuracy Evaluation for SDP II

As shown in the figure below, the model achieved strong performances across the evaluation metrics, with an mAP@0.5 of 91.00% and an mAP@0.5:0.95 of 49.96%. These models reflect the models ability to effectively detect and classify a diverse set of objects and behaviors, with notable precision over varying Intersection over Union (IoU) thresholds. The overall accuracy was reported at 74.00%, suggesting consistent and reliable predictions under practical classroom monitoring scenarios. Class-wise analysis revealed particularly strong performance for categories such as safety jacket and helmet, while some action-based classes like eating compared lower detection rates. Although this is to be expected as behaviors such as eating present greater variability in how individuals perform eating actions, leading to more diverse visual patterns, there is still room for potential improvement. The balance between precision, recall, and generalization highlights the model's suitability for real-time deployment, with future work aimed at enhancing the detection of highly variable behaviors to further optimize system robustness and adaptability.



```
Results saved to runs\detect\val5

===== Evaluation Metrics =====
mAP@0.5: 91.00%
mAP@0.5:0.95: 49.96%
Accuracy: 74.00%
PS C:\Users\Zayed>
```

Figure 53 Model Accuracy Evaluation - SDP II

6.1.4 Training Curve Analysis

The figure below demonstrates the training and validation metrics recorded during the YOLOv8s model training phase for both behavior and object detection across the YOLO models. These plots offer insight to learn the models progression and generalization capabilities.

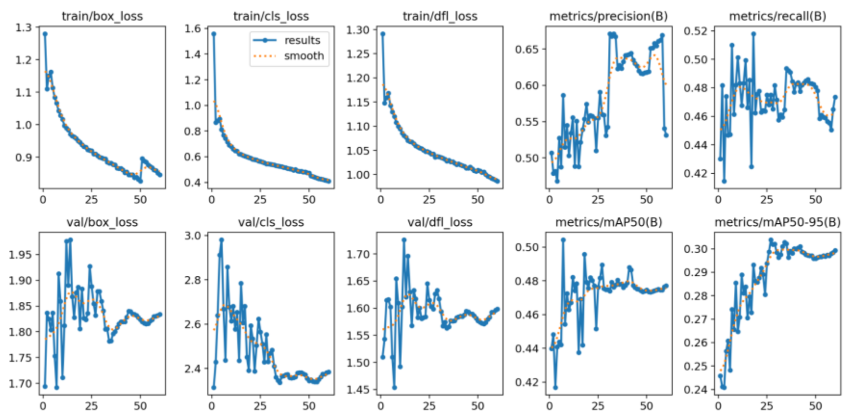


Figure 54 Training Curves

Loss Curves:

- Box Loss: The box loss curve shows a decrease from around 1.3 to 0.9
- Classification Loss: A steady decline was demonstrated from 1.5 to approximately 0.4
- Distribution Focal Loss: The team noticed a smooth drop from around 1.30 to 1.05. This reflects that the model is able to improve its ability to accurately localize and classify bounding boxes over time.

The results of the loss curves display a consistent downward trend, which indicates successful convergence during training.

Validation Losses:

- Validation Box Loss: As shown in the figure below it remained around 1.75-1.85
- Validation Classification Loss: Demonstrated a mild downwards trend
- Validation Distribution Focal Loss: After initial oscillations, it stabilized at around 1.50. This means that the model is able to generalize to unseen data, however the team took note of the fact that there is room for improvement.

It is evident the validation metrics showed more fluctuation than training losses, but this is to be expected due to batch variance and scene diversity.

Evaluation Metrics:

- Precision improved and peaked at around 0.66, which demonstrates a significant increase in correct positive predictions as the model became more confident.
- Recall stabilized at around 0.47 which shows consistent performance in retrieving relevant detections.

- mAP@0.5 gradually increased, then plateaued at 0.49, this showcases the strong detection performance at 0.5 IoU threshold.
- mAP@0.5:0.95 reached around 0.30, which reflects good generalization across different ranges of IoU thresholds. This is a reliable indicator that the model is robust across varying degrees of detection strictness.

Overall, these training curves validate the effectiveness of the YOLOv8 training pipeline. The clear decline in losses and the corresponding rise in mAP and precision demonstrate effective learning. It is evident that the model reached stable and reliable performance.

6.2 Results

This section presents a detailed analysis of the system's performance during training and testing stages for both SDP I and SDP II. For SDP I, the evaluation focused primarily on object classification accuracy and confusion matrix interpretation. In contrast, SDP II expanded the evaluation to include behavior recognition alongside object detection.

The training evaluation covered metrics such as precision, recall, and mean Average Precision (mAP), helping assess how well the models learned from the training datasets. Testing performance was evaluated using unseen video footage and analyzed through PR curves, F1 confidence curves, and confusion matrices. In addition, the evaluation in SDP II included real-time alert generation and system responsiveness, validating the practicality and robustness of the behavior-monitoring setup. While SDP I emphasized detecting physical entities in restricted areas, SDP II highlighted the system’s ability to handle more detailed behavior scenarios with greater variability and complexity—marking a significant advancement in the project’s capability and scope.

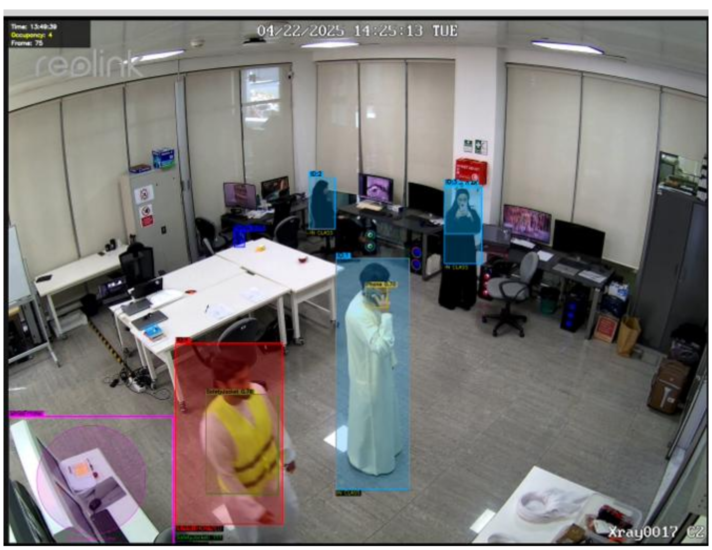


Figure 55 Successfully Detected and Works for Both Genders

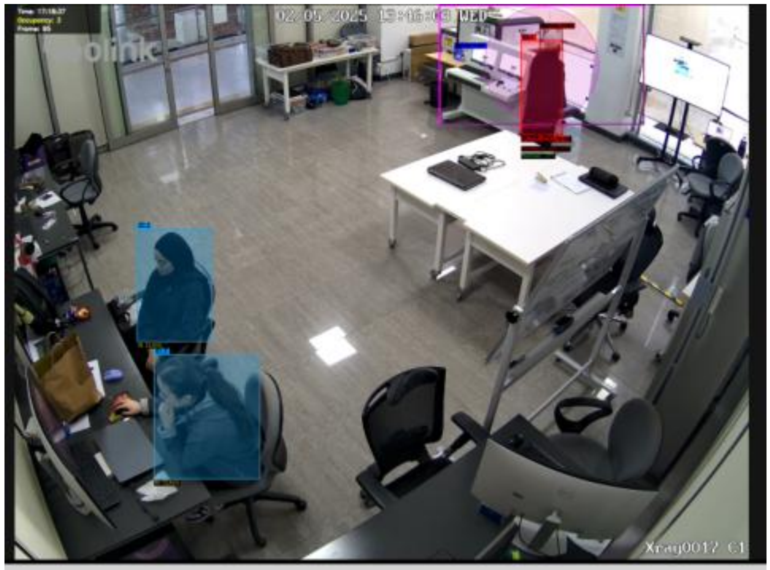


Figure 56 Successfully Differentiate Between Safety Jacket and Helmet

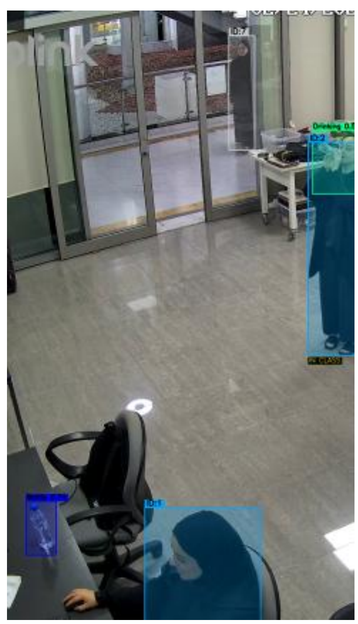


Figure 57 Successfully Differentiate Between Individuals In vs. Out the Classroom

6.3 Failure Cases Analysis

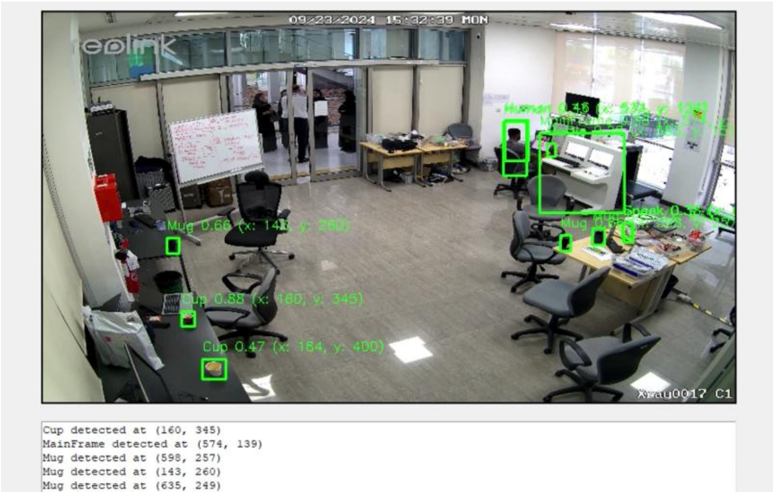


Figure 58 Failure to Alert Human in Restricted Area

The system successfully detects the "MainFrame" within the designated area, which is defined as an illegal area. However, the system fails to trigger an alert regarding the unauthorized person entering this restricted zone. Additionally, the detection system should not display the "MainFrame" boundaries, as the system is designed for keeping the restricted area hidden from view.

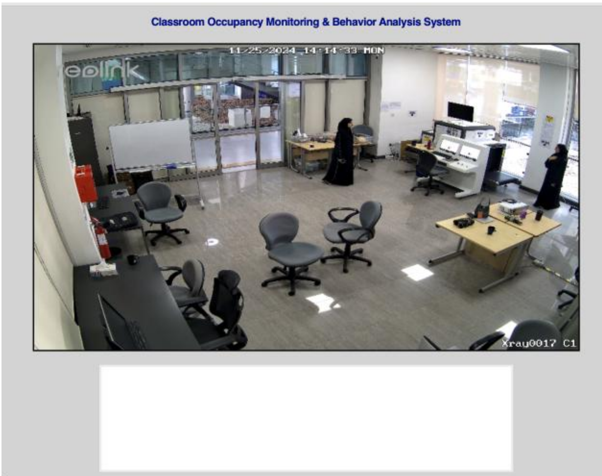


Figure 59 Failure to Detect Anything

The system has not detected any objects or persons in the illegal area. This shows that the space is currently empty, with no noticeable activities or events taking place.

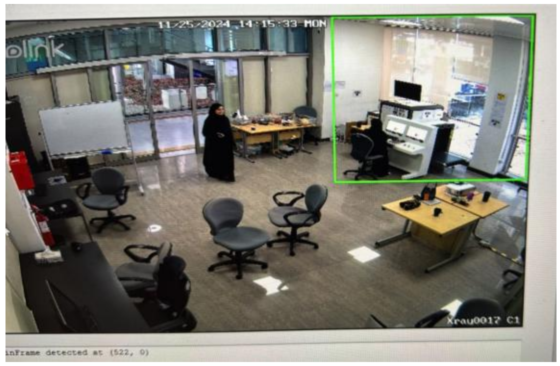


Figure 60 Only Detecting MainFrame

The system incorrectly detects only the "MainFrame," even though there is no person entering the illegal area to detect in the scene. This is a failure, as the system should not flag the "MainFrame" in the absence of any relevant activity or interaction. The detection is unnecessary and misleading.

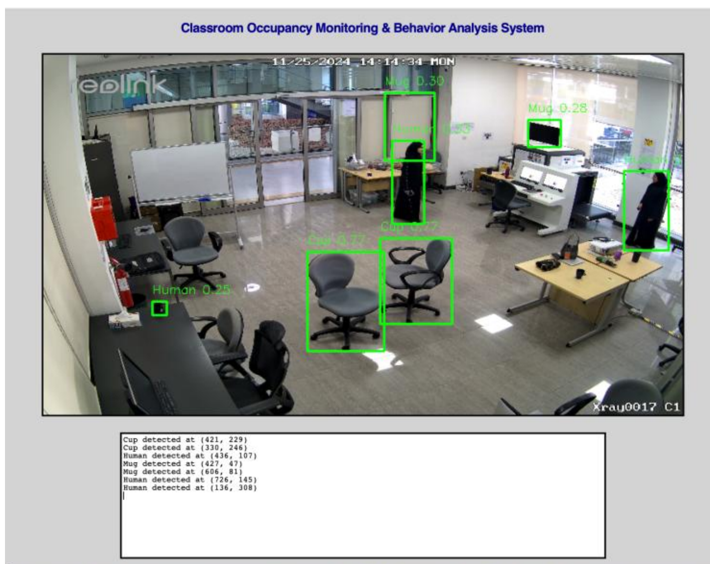


Figure 61 Incorrect Object Classification

The system shows incorrect labels. For example, a chair is mistakenly labeled as a "Cup," which is clearly wrong. Additionally, some "Human" labels appear in areas where there are no people, and other objects, like "Mug," are either mislabeled or detected with low confidence.

Figure 62 Drinking Behavior Not Detected

In this frame, the system successfully detects all the people present in the room and updates the occupancy count to six. However, it completely misses the drinking behavior of the person that is standing. No alert is triggered, and the system only labels the person as “Human” without recognizing the action taking place. This highlights a limitation in the behavior detection model, where clear actions like drinking can be overlooked, potentially affecting the accuracy and effectiveness of the monitoring system.

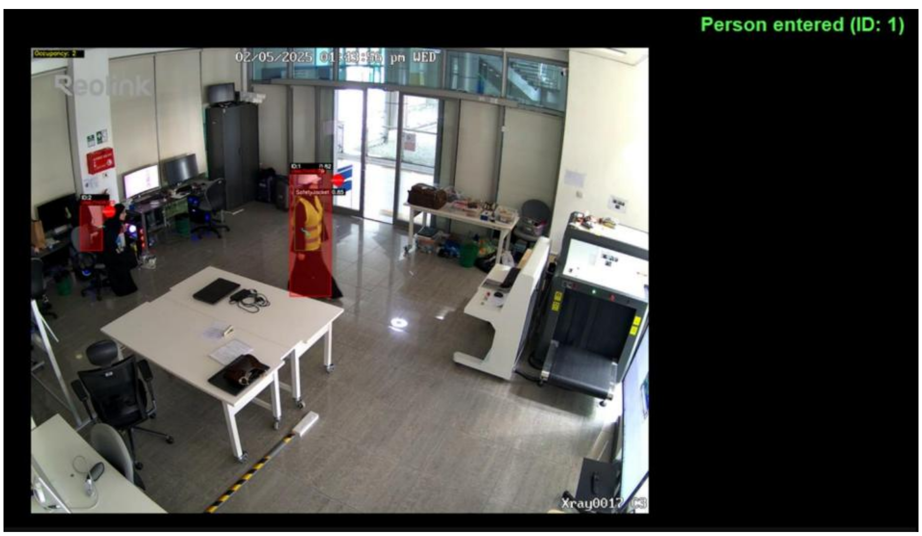


Figure 63 Inaccurate Bounding Box Placement

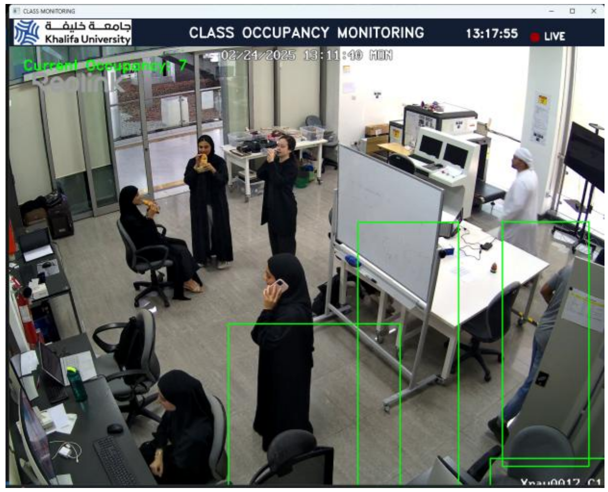


Figure 64 Detection Failure Due to Incorrect Window Size

In this scene, the system correctly detects the presence of individuals and updates the occupancy count to two. However, the bounding box around the person wearing the safety jacket is poorly placed and does not properly align with their body. This misalignment affects the reliability of object detection, especially when behaviors or safety need to be tracked accurately. Even though the “Person entered” alert is triggered, the inaccurate bounding box can lead to misclassification or failure to identify specific actions or violations.

This image shows the live monitoring view with an occupancy count of seven. Although several people in the frame are clearly eating or drinking, none of these behaviors are detected or labeled by the system. The issue here is caused by the incorrect window size, which distorts the video feed and affects the accuracy of the detection model. Because of this display issue, the system fails to recognize visible actions, highlighting the importance of maintaining proper video dimensions for reliable behavior detection. These failure cases provide actionable insights for future iterations and helps the team prioritize improvements such as bounding box accuracy and UI dimension scaling.

Chapter 7 Conclusion

7.1 Summary of work done

The project progressed through two significant phases, SDP I and SDP II, with each phase building upon the previous one to enhance the capabilities of the classroom monitoring system using artificial intelligence. In SDP I, the primary focus was on object detection. The team curated a comprehensive dataset featuring classroom items such as cups, bottles, and humans. This dataset was annotated with bounding boxes and class labels, forming the foundation for training a YOLOv8 model capable of detecting these objects in real-time video feeds. The system was also designed to log detections into structured CSV files and capture screenshots, particularly when unauthorized access to restricted areas (e.g., the MainFrame) was detected.

As the project transitioned to SDP II, the system's scope expanded significantly. Two YOLOv8 models were integrated: one for object detection and another for behavior recognition. The system now had the capability to detect both physical objects and a range of human behaviors, including cheating, phone use, eating, and compliance with safety regulations such as wearing helmets and safety jackets. To support these new capabilities, a more comprehensive and diverse dataset was created, reflecting the varied nature of behaviors and classroom settings. The project also introduced a dual-mode user interface, offering Normal Mode and Exam Mode for different monitoring scenarios. In Normal Mode, the system focused on detecting violations related to safety, such as the absence of helmets and safety jackets in restricted areas. When such violations were detected, the system automatically captured a screenshot and sent it via email. In Exam Mode, the system specifically monitored academic integrity, identifying cheating behaviors such as phone usage, texting, or phone calls, and sending alerts when such activities were detected. In both modes, the system tracked and logged occupancy counts, providing additional context for the detected events.

Furthermore, SDP II introduced a comprehensive email notification system, enabling the system to send instant alerts whenever violations were detected, ensuring a prompt response. This phase also marked a significant advancement in real-time monitoring capabilities, with the system now capable of sending immediate alerts and recording evidence, either through screenshots or screen recordings, and sending these directly to the designated recipients. Together, these improvements turned the system into a fully functioning classroom and exam monitoring solution that could reliably operate in real-world conditions, helping to enhance safety, accountability, and academic integrity.

7.2 Conclusion Statement

This project successfully demonstrated the practical application of computer vision and deep learning to meet the increasing demand for intelligent classroom monitoring solutions. Through a structured and systematic development process, the team effectively trained and deployed both object and behavior recognition models, which performed consistently and reliably in real-world scenarios. In SDP I, the project established the foundational object detection pipeline, which was crucial for identifying various items in the classroom. SDP II then expanded on this by introducing more sophisticated capabilities, including behavior analysis, user interface design, and real-time alerting functionalities. These advancements enabled the system to detect a wide range of behaviors, from safety violations to academic misconduct. This project not only showcased the importance of collaborative teamwork but also emphasized the power of iterative development in enhancing system performance. As a result, the system is now both scalable and reliable, significantly improving classroom safety, accountability, and operational efficiency. With the integration of smart monitoring tools, the system can detect, alert, and respond to activities in real time, offering a comprehensive solution for modern educational environments.

7.3 Project Constraints, Standards and the Environment

7.3.1 Project Constraints

Throughout the development of the classroom monitoring system, the group encountered several constraints that shaped their approach and implementation decisions. Time constraints posed a significant challenge, as the group had to deliver both the object detection capabilities in SDP I and the more complex behavior recognition features in SDP II within a tight academic timeline. This limited their ability to test the system across more diverse environments and scenarios, particularly for behavior detection where variations are nearly endless.

Hardware limitations affected the model selection and training approaches. While more complex models might have yielded higher accuracy, the group strategically chose YOLOv8s to balance performance with the computational resources available. The real-time processing requirements further constrained the options, as the system needed to analyze video feeds without significant latency to provide timely alerts for safety violations and academic dishonesty. Dataset constraints significantly impacted model training. Despite efforts to capture diverse scenarios, the primary dataset was collected from a controlled laboratory environment with limited camera angles. Though supplemented with Roboflow datasets, the lack of extensive real-world classroom data across different lighting conditions, room layouts, and student demographics potentially limits the system's generalizability.

Privacy and ethical considerations constrained the data collection approach. The group needed to ensure that all participants were aware of the recording and that data was handled responsibly. This limited their ability to gather extensive data from actual exam scenarios, where academic dishonesty would naturally occur. Resource constraints affected implementation choices. With limited access to high-performance computing resources, the group had to optimize the training process, carefully selecting batch sizes and number of epochs to achieve reasonable results without requiring excessive training time or computational power.

Finally, deployment constraints shaped the interface design and alert mechanisms. The system needed to function on typical hardware found in educational institutions, with user-friendly interfaces that didn't require specialized technical knowledge to operate, limiting some advanced features that might otherwise have been implemented.

These constraints, while challenging, ultimately led to practical design decisions that ensured the system remained implementable within the available resources while still meeting the core objectives of classroom monitoring and behavior analysis.

7.3.2 Standards

The development of the classroom monitoring system involved adherence to several important standards and ethical guidelines:

IEEE 2089-2021: Standard for Age Appropriate Digital Services Framework This standard provided guidance on developing systems that interact with users of different age groups, including students. The group considered these principles when designing the monitoring system to ensure it maintained appropriate boundaries while observing classroom activities.

ISO/IEC 27001: Information Security Management This standard helped shape the data security protocols for the system, particularly important since the monitoring system captures and stores potentially sensitive classroom footage. The group implemented secure storage practices and access controls to protect recorded data.

IEEE P7000: Model Process for Addressing Ethical Concerns During System Design This standard influenced the ethical considerations throughout the design process, particularly regarding privacy and transparency. The system incorporates visible indicators when monitoring is active to maintain transparency with students and faculty.

IEEE 7010-2020: Recommended Practice for Assessing the Impact of Autonomous and Intelligent Systems on Human Well-Being This standard helped the group evaluate potential psychological impacts of persistent monitoring on students and teachers, guiding the development of non-intrusive alert systems that minimize classroom disruption.

IEEE/ACM Code of Ethics The development team adhered to core principles from these codes, including:

- Acting in the public interest and considering the welfare of students and teachers
- Avoiding conflicts of interest in system design and evaluation
- Maintaining honest reporting of system capabilities and limitations
- Respecting privacy while achieving legitimate monitoring objectives

General Data Protection Regulation (GDPR) Principles Though not formally required in all contexts, GDPR principles guided data collection, processing, and retention policies, emphasizing data minimization and purpose limitation for all captured footage.

NIST Special Publication 800-53: Security and Privacy Controls This publication provided a framework for implementing appropriate technical security controls to protect the data collected by the monitoring system.

ISO 14001: Environmental Management Systems This standard influenced decisions about system design that would minimize environmental impact, including optimizing computational efficiency to reduce energy consumption during continuous monitoring operations.

These standards collectively guided the ethical, technical, and environmental aspects of system development, helping the group navigate complex considerations around surveillance, privacy, security, and sustainability while creating an effective classroom monitoring solution.

7.3.3 Environmental Impacts and Sustainability

Environmental considerations played a subtle but important role in the development of the classroom monitoring system. Though not explicitly stated as primary objectives, the group made several design choices that positively impacted the system's environmental footprint. The selection of YOLOv8s as the core detection model represented a deliberate balance between computational requirements and detection accuracy. By choosing this model over more resource-intensive alternatives, the group reduced the overall energy consumption needed for real-time processing. This decision was particularly significant given that classroom monitoring systems typically operate for extended periods during academic hours. The streamlined architecture of YOLOv8s requires less computational power, resulting in lower electricity usage when compared to larger, more complex models that might offer marginally better detection at the cost of substantially higher energy consumption.

Software optimization efforts further enhanced the system's sustainability profile. The processing pipeline was designed to minimize redundant operations, particularly when running dual models for both object and behavior detection. Frame skipping was implemented during periods of low activity, allowing the system to reduce computational load during quieter classroom moments. These algorithmic efficiencies directly translate to lower power requirements, especially important when considering that such systems might be deployed across multiple classrooms in an educational institution. The group also embraced resource efficiency in their implementation approach. Rather than developing a system requiring specialized hardware, they designed the software to function on typical desktop computers commonly found in educational settings. This hardware-agnostic approach extends the useful life of existing equipment, reducing electronic waste by eliminating the need for dedicated monitoring hardware. Additionally, the flexible configuration options allow administrators to adjust processing priorities, balancing detection accuracy against power consumption according to specific monitoring needs.

Data storage considerations reflected environmental awareness as well. The system was designed to save screenshots and video clips only when violations were detected, rather than maintaining continuous recording. This selective storage approach significantly reduces digital storage requirements and the associated energy costs of maintaining large data repositories. Alert emails were also designed to be concise, with optimized image attachments that reduce data transfer volumes and the corresponding network infrastructure load. While these environmental considerations were secondary to the primary functional objectives, they demonstrate how thoughtful design choices can reduce resource consumption even in specialized AI applications. Future iterations of the system could further emphasize sustainability by incorporating more explicit power management features, such as scheduled monitoring periods or adaptive processing based on classroom occupancy levels.

7.3 Critical Appraisal of Work Done

The project made significant advancements in developing AI-driven object and behavior detection for classroom monitoring. In SDP I, the team successfully established a robust object detection pipeline, achieving impressive accuracy in detecting classroom items such as cups, mugs, and humans using a well-structured YOLOv8 model. This laid the foundation for SDP II, where the system's functionality was expanded with the integration of dual models—one for object detection and another for behavior recognition. Real-time alerting, structured logging, and an automated email notification system were also implemented, significantly enhancing the system's responsiveness and usability. Collaborative effort was key throughout, with the team refining the models, developing the user interface, and adding features across both phases.

However, despite the progress, the project encountered several challenges that revealed areas for further refinement. One recurring issue was ensuring strong generalization between visually similar behavior classes, such as 'PhoneCall' and 'Eating'. These behaviors required more robust annotations and a broader set of training samples to reduce confusion and false positives. Another challenge arose from maintaining real-time responsiveness while running dual-model detection, which demanded substantial processing resources. To address this, the system was optimized to efficiently manage frame processing and alert generation, minimizing any lag in performance.

Despite these challenges, the system demonstrated reliable performance in real-world testing, producing meaningful outputs through detection logs, screenshots, and email alerts. These results lay a strong foundation for future iterations, where improvements in detection granularity, system efficiency, and user experience can be made.

7.3.1 Challenges and Solutions

1. **Video Feed Quality:** Poor lighting conditions and low-resolution video posed initial challenges during detection, particularly for smaller or darker objects. To address this, image enhancement techniques and preprocessing filters were introduced to improve frame clarity and support consistent model performance.
2. **Model Accuracy for Ambiguous Behaviors:** Behavioral classes such as 'PhoneCall', 'Eating', and 'Drinking' were more difficult for the model to detect accurately due to their

subtle and overlapping visual features. This was resolved by augmenting the dataset with more varied examples and refining the labeling process to improve clarity and training consistency.

3. **System Responsiveness and Scalability:** Integrating two models in parallel and processing continuous video streams in real time initially led to lag and delayed alerting. This was resolved by optimizing the model inference loop and balancing system resource allocation. Additional logic was also implemented to manage alert timing and reduce repeated triggers, ensuring the system remained responsive and scalable.
4. **Alert Communication:** A new feature introduced in SDP II was email alerting. Ensuring timely and reliable email delivery under network constraints required configuring SMTP services and testing across different email platforms. Once implemented, this allowed users to receive instant notifications during rule violations, extending the system's reach beyond local monitoring.

These solutions reflect the team's ability to adapt to technical challenges and improve the system through iterative problem-solving and teamwork.

This project demonstrated the practical application of computer vision and deep learning to address the growing need for intelligent classroom monitoring solutions. Through systematic development, the team successfully trained and deployed object and behavior recognition models that performed reliably in real-world scenarios. SDP I laid the groundwork by establishing the object detection pipeline, while SDP II introduced advanced capabilities for behavioral analysis, interface design, and real-time alerting. The project not only highlighted the value of collaborative teamwork but also underscored the effectiveness of iterative development in improving system performance. As a result, the outcome was a scalable and reliable system that enhances safety, accountability, and efficiency in classrooms using smart monitoring tools that can detect, alert, and respond to activities in real time.

7.4 Next Steps

7.4.1 Model Training Enhancement

While our current detection models provide a strong foundation, there are several areas for improvement to ensure more accurate and robust performance in both safety compliance and academic integrity monitoring:

- Expand the safety compliance detection model by incorporating additional scenarios from different environments, such as construction sites, to account for a wider range of safety conditions.

- Improve detection accuracy for various safety equipment, including hard hats, high-visibility vests, safety goggles, gloves, and steel-toed boots, ensuring reliable identification across diverse settings.
- Incorporate various lighting conditions and partial occlusion scenarios to improve detection reliability in real-world settings.
- Enhance the cheating detection model by including a more diverse set of academic dishonesty behaviors, such as the use of smartwatches, hidden earbuds, and note passing.
- Refine the detection of safety equipment usage, not just presence, to ensure proper usage, such as correctly fastened helmets or properly worn safety gear.
- Utilize more advanced training techniques to address edge cases and challenging detection scenarios, including scenarios involving partial occlusion or subtle violations.

These steps will help further enhance the system's reliability and expand its applicability to a broader range of real-world environments.

7.4.2 Feature Expansion

To further enhance the capabilities of both the safety compliance and academic integrity systems, the following feature expansions are proposed:

- Implement multi-camera support for comprehensive classroom and facility coverage
- Add audio analysis capabilities to detect verbal cues in both contexts
- Develop proximity detection to identify unsafe distances between personnel and hazardous equipment, as well as suspicious proximity between students
- Create configurable monitoring zones with different sensitivity levels for both academic and safety contexts
- Implement predictive analytics to identify potential risks before violations occur
- Develop capabilities for detecting improper tool usage and unsafe work practices in industrial settings
- Add environmental monitoring for industrial workplaces
- Create advanced behavioral analysis for exam environments to detect subtle cheating patterns

7.4.3 Scalability and Deployment

To ensure the system can support larger organizations and adapt to growing needs, several scalability improvements are recommended:

- Develop multi-site monitoring capabilities with centralized management for both academic institutions and industrial facilities
- Create hierarchical reporting structures for complex organizational needs

- Implement cross-platform compatibility for diverse IT ecosystems
- Develop cloud-based deployment options with appropriate security controls
- Create enterprise-grade authentication and authorization systems
- Implement resource-efficient monitoring for high-camera-count environments
- Develop automated scaling capabilities for fluctuating monitoring needs
- Create comprehensive system health monitoring and alerting

References

[1] T. R. Ruggles and J. M. LeBlanc, "Behavior Analysis Procedures in Classroom Teaching," in International Handbook of Behavior Modification and Therapy, Springer, Boston, MA, 1985. [Online]. Available: https://doi.org/10.1007/978-1-4615-7278-7_11

[2] J. H. Lim, E. Y. Teh, M. H. Geh, and C. H. Lim, "Automated classroom monitoring with connected visioning system," in 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, Malaysia, 2017, pp. 386–393. doi: 10.1109/APSIPA.2017.8282063.

[3] Y. Nureni and O. Asiribo, "Networking CCTV cameras & Passive Infra-Red sensors for E-classroom monitoring system: Proactive approach to quality assurance in education system," ResearchGate, Jul. 2017. [Online]. Available: https://www.researchgate.net/publication/318636912

[4] X. Zhao, G. Wang, Z. He, and H. Jiang, "A survey of moving object detection methods: A practical perspective," Neurocomputing, vol. 503, pp. 28–48, 2022. [Online]. Available: https://doi.org/10.1016/j.neucom.2022.06.104

[5] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Machine Learning and Knowledge Extraction, vol. 5, no. 4, pp. 1680–1716, 2023. doi: https://doi.org/10.3390/make5040083

[6] J. Pedro, "Detailed Explanation of YOLOv8 Architecture: Part 1," Medium, Dec. 4, 2023. [Online]. Available: https://medium.com/@juanpedro.bc22/detailed-explanation-of-yolov8-architecture-part-1-6da9296b954e.[Accessed: Nov. 1, 2024].

[7] V. Shehu and A. Dika, "Using real time computer vision algorithms in automatic attendance management systems," in Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces, Cavtat, Croatia, 2010, pp. 397–402.

[8] A. S. Pillai, "Student Engagement Detection in Classrooms through Computer Vision and Deep Learning: A Novel Approach Using YOLOv4," Sage Science Review of Educational Technology, vol. 5, no. 1, pp. 87–97, Aug. 2022. [Online]. Available: https://journals.sagescience.org/index.php/ssret/article/view/144

[9] W. Zhang, J. Calautit, P. W. Tien, Y. Wu, and S. Wei, "Deep Learning Models for Vision-Based Occupancy Detection in High Occupancy Buildings," Journal of Building Engineering, 2024, Article ID 111355. [Online]. Available: https://doi.org/10.1016/j.jobe.2024.111355

[10] "GPU Acceleration with PyTorch," Toxigon. [Online]. Available: https://toxigon.com/gpu-acceleration-pytorch [Accessed: Nov. 14, 2024].

[11] Python Software Foundation, "smtplib — SMTP protocol client," Python Documentation. [Online]. Available: https://docs.python.org/3/library/smtplib.html [Accessed: April. 18, 2025].

[12] J. Fincher, "An Intro to Threading in Python," Real Python. [Online]. Available: https://realpython.com/intro-to-python-threading/ [Accessed: April. 18, 2025].

[13] "Queue in Python," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/queue-in-python/ [Accessed: April. 18, 2025].

[14] N. Ndung'u, "Image Processing With the Python Pillow Library," Real Python. [Online]. Available: https://realpython.com/image-processing-with-the-python-pillow-library/ [Accessed: April. 18, 2025].

[15] S. Dewi, Y. Mahendra, and I. R. Kusuma, "Implementation of PyTorch Framework in Building Deep Learning Models," Journal of Physics: Conference Series, vol. 1693, no. 1, p. 012100, 2020. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1693/1/012100/meta [Accessed: Nov. 14, 2024].

[16] G. Nitschke and L. Taylor, "Improving Deep Learning with Generic Data Augmentation," in Proceedings of IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2018), Bengaluru, India, 2018, pp. 1–8.

[17] "An Introduction to YOLOv8," RevComm Tech Blog. [Online]. Available: https://tech.revcomm.co.jp/yolov8-introduction-en [Accessed: Nov. 16, 2024].

[18] "How IEEE Standards Protect the Planet and Keep Us Safe," IEEE Transmitter. [Online]. Available: https://transmitter.ieee.org/how-ieee-standards-protect-the-planet-and-keep-us-safe/. [Accessed: April. 25, 2025].

Appendices

Appendix A: Logbook

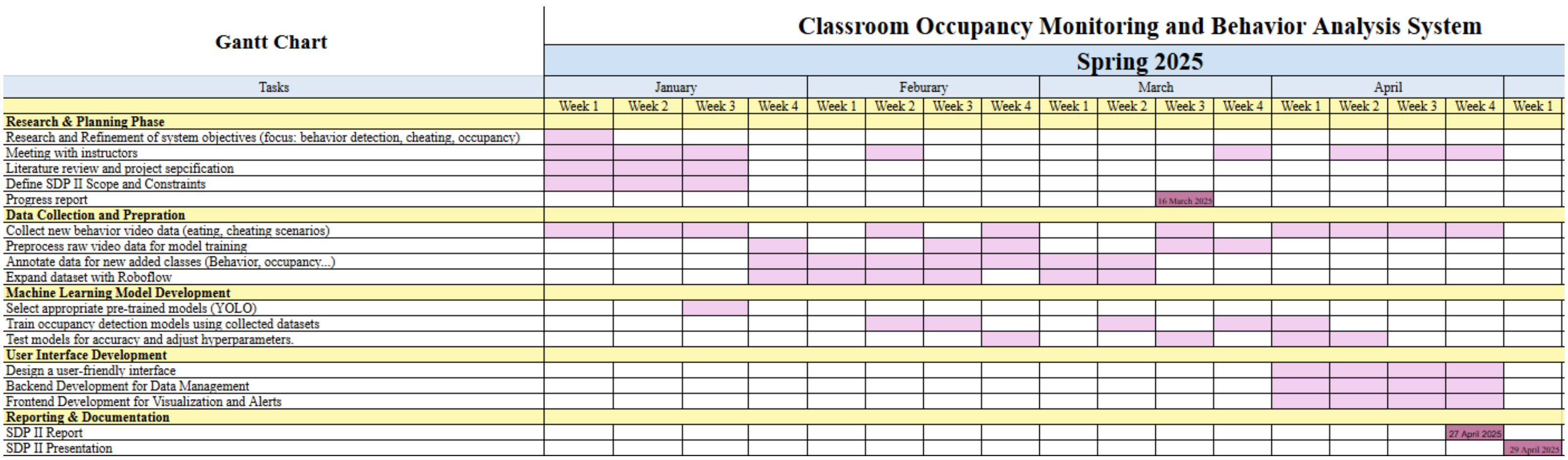
Date	Event/Meeting	Details
13/1/2025	Meeting with Dr. Naoufel	Discussed the next steps for *Senior Design Project II* and evaluated the best approach to proceed with the project.
14/1/2025	Group Meeting	Held a team meeting to plan the upcoming tasks. Discussed work distribution and identified additional classes to integrate into the project.
14/1/2025	Meeting with Dr. Naoufel	Held a brief meeting with Dr. Naoufel to review and obtain approval for the newly proposed behavior classes.
15/1/2025	Collecting Data	Collaborated with Dr. Maregu to gather additional data from the CCTV system in the R Building lab to support the project.
22/1/2025	Collecting Data	Collected additional data in a new setting, focusing on a classroom environment to enhance the dataset.
23/1/2025	Collecting Data	Gathered additional data specifically focused on safety equipment classes to expand and refine the dataset.
24/1/2025	Splitting Data	Divided the collected data among group members to begin the labeling process for each class.
25/1/2025-8/2/2025	Labeling the Data	Over this period, the team started labeling the collected data. After dividing the data frames among group members, each member carefully annotated the data according to the classes. This process was essential for ensuring consistency and accuracy in the dataset, laying the groundwork for subsequent analysis and model training.
5/2/2025	Collecting Testing videos	Gather testing videos with all the different classes from the CCTV to be tested once the training process is finished.
13/2/2025	Training Our Model SDP II model	Trained the initial YOLOv8s model using the labeled dataset

20/2/2025	Collecting more training videos	After our initial training, we observed that our model was trained with an imbalance; therefore, we needed to collect more data on the behaviors.
20/2/2025	Split new data	We split the newly collected data among group members to start labeling the classes.
24/2/2025	Collecting data on new class	Gathered data for a new behavior class “cheating” from the lab CCTV.
27/2/2025	Split the cheating data	Splitting the new class frames among group members to start labeling the class.
3/3/2025	Meeting with dr. Naoufel	Discussed the integration of the “Phone” and “Cheating” behavior classes and reviewed dataset adjustments.
14/3/2025	Re-trained model	Re-trained the YOLOv8s model with the updated dataset including the new classes.
15/3/2025	Model testing	Evaluated the updated model using the collected test videos.
16/3/2025	Progress report submission	Submitted the SDP II progress report.
20/3/2025	Test Video Recording	Recorded additional footage for testing purposes.
24/3/2025	Interface Development	Worked on building and refining the user interface for the system

30/3/2025	Email Alert System Integration	Implemented an email alert system that automatically sends notifications when specific events are detected, enhancing real-time response and communication.
7/4/2025	Meeting with Dr. Naoufel	Discussed the content and structure of the SDP II final report.
8/4/2025	Group meeting	Held a group meeting to review current progress and discuss the remaining tasks required for system integration and testing. The team also outlined a timeline for upcoming deliverables and coordinated next steps to ensure timely project completion
10/4/2025	Report Distribution	Assigned sections of the final report to group members.
17/4/2025	Collecting more videos	Collecting more videos for testing the system.
22/4/2025	Final Test Video Collection	Collected final set of videos for system testing and validation.
23/4/2025	SDP II Presentation Preparation	Work began by splitting and creating slides for the SDP presentation, highlighting major achievements, challenges, and future plans.
25/4/2025	SDP II Final Report Preparation	The group met on campus to finalize the SDP II Final Report.
27/4/2025	SDP II Final Report Submission	Submitted the SDP II Final Report.
29/4/2025	SDP II presentation	The team delivered the final presentation, showcasing the project to supervisors and examiners. The presentation highlighted key achievements,

		demonstrated the system's functionality, and addressed questions about potential improvements and future applications.

Appendix B: Gantt Chart



Appendix C: Git-Hub link

[GitHub - CS06-SDP/CS06]()