

## Near Real-Time Room Monitoring System for Occupancy Detection in the College of Computing and Information Sciences at Caraga State University-Main

Bryan Ian C. Ramos

College of Computing and Information Sciences, Caraga State University

Patrick James D. Dionen

College of Computing and Information Sciences, Caraga State University

R.Z A. Tabaranza

College of Computing and Information Sciences, Caraga State University

Giovanni T. Esma

College of Computing and Information Sciences, Caraga State University

<https://doi.org/10.5109/7395692>

---

出版情報 : Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES). 11, pp.1395-1401, 2025-10-30. International Exchange and Innovation Conference on Engineering & Sciences

バージョン :

権利関係 : Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International



## Near Real-Time Room Monitoring System for Occupancy Detection in the College of Computing and Information Sciences at Caraga State University-Main

Bryan Ian C. Ramos<sup>1</sup>, Patrick James D. Dionen<sup>1</sup>, R.Z A. Tabaranza<sup>1</sup>, Giovanni T. Esma<sup>1</sup>

<sup>1</sup>Caraga State University - Main, College of Computing and Information Sciences,

[gtesma@carsu.edu.ph](mailto:gtesma@carsu.edu.ph)

**Abstract:** *A laboratory occupancy monitoring system is essential to promote safety, efficiency, and resource management in school operation. This study aims to develop an automated, near-real-time monitoring system that processes real-time CCTV streams to track occupancy, enhancing monitoring efficiency. The system utilizes YOLOv5 for object detection and DeepSORT for tracking. A web-based application, built using FastAPI and running on Uvicorn, delivers asynchronous data updates, displaying current occupancy levels and logging user stay durations against scheduled room usage. Compared to manual tracking, the automated system significantly reduces staff workload and enhances accuracy in monitoring and resource allocation. The results highlight the system's potential to improve operational efficiency and support informed decision-making in laboratory management.*

**Keywords:** Computer Vision, Near-real time, DeepSORT, Object Detection, YOLOv5

### 1. INTRODUCTION

The study developed a near-real-time laboratory occupancy monitoring system using computer vision, powered by the lightweight YOLOv5 model and deepSORT for accurate object detection. This configuration avoids repeated detection of the same individual across frames, achieving an average occupancy accuracy of 90% to 95%. Leveraging modern tools such as React.js, Vite, FastAPI, and Uvicorn, the system supports fast, asynchronous processing, with an average data retrieval time of 39.34 seconds across eight laboratories. While minor false positives occurred in early frames, detection accuracy improved as more frames were processed, enhancing the overall reliability of occupancy logs.

The system addresses staffing challenges in the College of Computing and Information Sciences by integrating with existing Hikvision RTSP streams to track lab occupancy automatically. It logs activity with minimal delays under 60 seconds and supports usage insights based on occupancy duration. Usability testing yielded a System Usability Scale (SUS) score of 75.75%, above the industry average. Users highlighted ease of use, integration, and confidence in system operation. While most roles reported high satisfaction, slightly lower ratings from some staff reflected the system's development constraints. Overall, the system enhances lab monitoring efficiency and supports smarter room usage.

#### 1.1 Background of the study

According to Rukmani et al. [7] and Yang et al. [8], as technology advances, smart monitoring systems have been adopted to enhance occupant comfort, energy efficiency, and resource management, with fine-grained occupancy data playing a key role in optimizing building operations. Real-time monitoring through computer vision and image processing enables automation and dynamic adaptation [7][8], with vision-based approaches reducing energy demand by analyzing surveillance data [5]. Deep learning models like YOLO combined with

DeepSORT enable accurate, fast object tracking for live applications such as social distancing monitoring [1][6]. In education, smart classrooms use IoT and AI-based monitoring systems to manage electricity, air quality, and safety, while also aiding emergency response and supporting planning and research [3][2].

#### 1.2 Statement of the Problem

The monitoring of the laboratory occupancy in the College of Computing Information Sciences is performed manually. This method is prone to human error and does not provide any real-time update, which is important in optimizing the laboratory usage and scheduling. Although CCTV cameras are installed, the current approach requires staff to monitor CCTV streams on the screen the whole day and review recordings which is time consuming. Moreover, the laboratory manager's need to remain focused on monitoring screens significantly limits their ability to perform other administrative or operational tasks. These limitations highlight the need for an automated monitoring solution that can enhance efficiency, reduce staff workload, and provide timely insights for better resource management.

#### 1.3 Objectives of the study

The general objective of the study is to develop a laboratory near-real-time room occupancy monitoring system with the integration of a Real-Time Streaming Protocol for near-real-time feedback of laboratory CCTV cameras. The specific objectives of this study are the following: utilize YOLOv5 for person detection using the Deep SORT algorithm to handle occlusions and re-identify objects for laboratory occupancy detection; capture near-real-time laboratory occupancy via Real-Time Streaming Protocol (RTSP); develop a user-friendly monitoring dashboard that displays near-real-time occupancy data, including number of people detected within the room, time-in and time-out, logs, and duration of occupancy to enhance the monitoring and management capabilities of laboratory staff within a single interface; and evaluate the system's usability and effectiveness by conducting a System Usability Scale (SUS) assessment.

## 2. RELATED WORKS

### 2.1 Integration of CCTV Streams Through RTSP

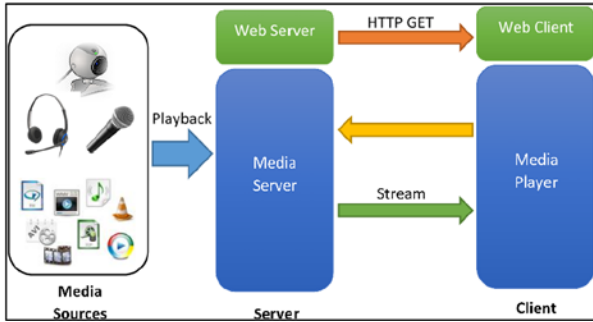


Fig. 1. Interaction Between Client and Server Using RTSP

Real-Time Streaming Protocol (RTSP) is a widely used standard for transmitting live video streams from CCTV cameras, enabling real-time monitoring. In educational settings, the integration of RTSP with existing CCTV infrastructure could provide a cost-effective solution for continuous surveillance and analysis. According to Syaifudin et al. [9], RTSP is often used for streaming digital media, including audio and video in real time allowing clients control real-time media streaming on the server. Furthermore, the use of RTSP enables viewing live video from direct source (such as encoder or camera) through the use of standard media player that is compatible with the protocol as shown in Figure 1. By creating RTSP text-based link protocol, users can access multimedia remotely [10].

### 2.2. Application of YOLO and DeepSort in Monitoring

You Only Look Once (YOLO) is a modern computer vision model used to predict bounding boxes and class probability of identified objects. It was introduced by Joseph Redmon and Ali Farhadi in 2015, and now it is a viral and widely used algorithm in Computer Vision. Due to its detection characteristic. Jiang, P et al. [11] stated that YOLO has a straight forward structure which directly outputs the position and category of bounding boxes through the neural network. According to Xie and Yao [12], YOLO in 2016 was designed with a new network increasing the speed of operation of the entire network by casting the object detection entirely into a regression problem, spanning input image segmentation through the classification of each grid target.

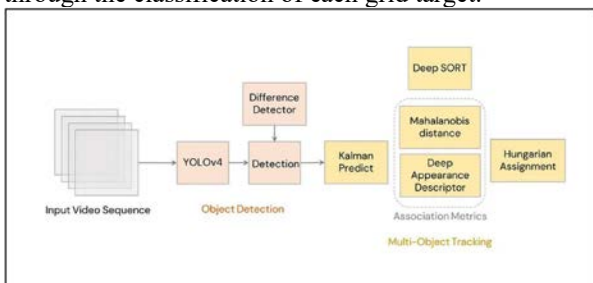


Fig 2. Multi-Object tracking with the use of YOLO and DeepSort algorithm [13].

DeepSORT algorithm uses a simple Kalman filter for dealing correlations of frame by frame data, and uses Hungarian algorithm to measure its correlation. Figure 2 highlights a sample of using YOLO with DeepSORT algorithm. Additionally, DeepSORT multiple object tracking pipeline processes a video sequence, with the use of YOLOv4 for detecting objects in each frame. The

detection of objects is then tracked using a Kalman Filter for predicting and combining Deep Appearance Descriptors and Mahalanobis distance for association metrics. Lastly, Hungarian Assignment algorithm is used to assign unique identities tracking objects across the frames for a robust multi-object tracking. Overall, YOLO and DeepSORT combination is a creative approach in real-time monitoring, 18 object detection and tracking assigning tracked objects with IDs in frame by frame maintaining the detection stability of the model.

### 2.3 YOLOv5 as a Lightweight Model

YOLOv5 remains widely recognized for its reliability and efficiency in various applications due to its real-time performance and high accuracy, making it a popular choice for real-world applications such as object detection and tracking for small and simple systems. According to Yu et al. [14] small object detection algorithms basically suffer from low accuracy of detecting small objects. In addition, Ren et al. [15] stated that the traditional object detection algorithm on mobile, embedded and limited server devices is a challenge due to limitation of memory and computation resources. A study of Zhou et al. [16] successfully implemented YOLOv5 in a RFID defect detection system with 98.9% recognition rate. Ming et al. [17] integrate YOLOv5 with the production cameras to detect defects on towels during production line.

## 3. METHODOLOGY

### 3.1 Conceptual Framework

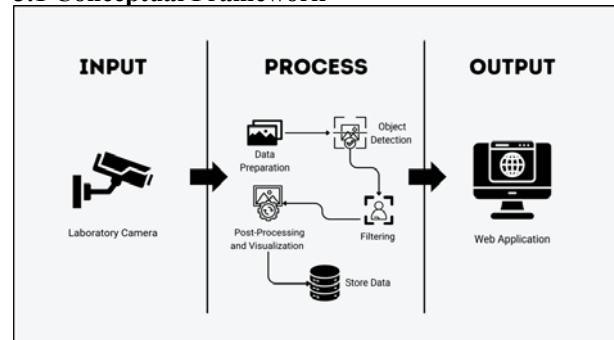


Fig. 3. Conceptual Framework

The presented diagram shown in figure 3, illustrates the operational framework of the Image processing monitoring system. It outlines the implementation of monitoring by integrating multiple cameras, used to detect persons within a single pretrained YOLO model. As illustrated, the input frames captured from the real-time streaming protocol (RTSP) link of the camera will be prepared and used by the model to detect number person and which can later be stored in the database and used for client interface assumption based on tracking number of people, date and time of room duration occupancy.

With this type of system architecture, monitoring laboratories usage will become more efficient as it provides near-real-time data of room occupancy, supporting laboratories incharge to make data-driven decision making for identifying rooms that are most likely occupied and investigate patterns that might be helpful for future maintenance, security and allocation of lab available resources.

#### 3.2.1 Requirement Analysis

The researcher aims to determine the specific needs for a Near-Real-Time Room Monitoring System by analyzing

the laboratory environment, activities, and monitoring requirements of the laboratory in-charge. The study will focus on:

- **CCTV Integration:** Assessing current camera types, placements, and how footage is accessed and integrated into lab operations.
- **Challenges and Limitations:** Identifying issues like technical constraints, system compatibility, and difficulties in accessing or reviewing footage.
- **Functional Requirements:** Defining features such as real-time alerts, object detection, a user-friendly dashboard, and analytics tools.
- **Non-Functional Requirements:** Documenting performance, security, scalability, and reliability needs to ensure system effectiveness and stability.

### 3.2.2 System Architecture and Implementation Design

The Near-Real-Time Room Monitoring System for Occupancy Detection is designed to minimize delay and ensure efficient processing, with a target update cycle of under 60 seconds. It uses a client-server architecture, with a dynamic React.js and Vite dashboard on the client side, providing real-time occupancy updates. On the server side, FastAPI and Uvicorn handle asynchronous video stream processing and API requests, reducing latency. The system integrates Hikvision CCTV cameras through RTSP connections to stream live footage, which is processed using YOLOv5mu for object detection and deepSORT for tracking, ensuring accurate identification without repeated detections. Updates to the dashboard occur within a 60-second cycle. A relational database stores occupancy logs, motion detection events, and historical data, optimized for fast queries, enabling quick access to occupancy trends and analytics while maintaining scalability and performance.

### 3.2.3 Implementation

The implementation of the Near-Real-Time Room Monitoring System focuses on seamless integration for real-time occupancy detection, with a strict delay limit of under 60 seconds. Hikvision CCTV cameras connect via RTSP over LAN for reliable streaming, supporting both wired and wireless connections. The YOLOv5mu model detects individuals, and deepSORT tracks them across frames. The system uses React.js and Vite for the client dashboard, while FastAPI and Uvicorn handle server-side processing and real-time updates. Occupancy logs and data are stored in a PostgreSQL database, with image files stored externally in Supabase to maintain efficiency. This setup ensures fast data retrieval, accurate tracking, and effective reporting of occupancy activities.

### 3.2.4 Testing

The testing phase focused on validating the core functionalities and performance metrics of the Near-Real-Time Room Monitoring System. The primary objective was to ensure that the YOLOv5mu model accurately detects the presence of a person in the room and registers occupancy logs in the database when a person enters or leaves the room. However, while the

model successfully detects the person, it does not accurately count the number of people in the room.

To evaluate the system's performance, five test tasks were conducted to assess the delay in occupancy detection. These tests specifically measured the time taken to detect occupancy within each room, considering the system processes video streams from eight CCTV cameras simultaneously. The goal was to ensure that the delay for processing the entire cycle of all eight CCTV cameras remained under 60 seconds, in line with the system's near-real-time requirements. Testing confirmed that the system met this performance target, with all tests showing that the occupancy detection delay was consistently maintained below the 60-second threshold.

### 3.3 Camera Configuration and Field of View

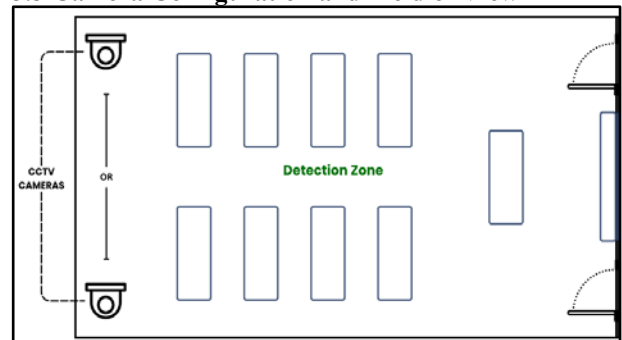


Fig. 4. Camera Field Zone

Considering the camera HikVision angle, and position. Researchers will utilize the existing setup of laboratories CCTV point of view as detection zones. During the detection process, the frames from real-time streams of CCTV's will be analyzed by the pretrained model identifying class-0 object (person) summing all the detected classes before measuring the weight of laboratory occupancy.



Fig. 5. Multimedia Laboratory Object Detection Camera Set-up

Moreover, the set-up of the CCTV is wide enough to recognize a person even at the longest distance. As long as the entity is visible to the camera, the detection of tracking time and date of room occupancy will be effective, despite the existence of minor limitations such as misidentifying the entity due to obscuration at the point of view of the camera. Overall the feed from the camera stream is still reliable and will be used for capturing occupancy of laboratories in CCIS.

### 3.3 Occupancy Detection Process

The development of the system is designed to monitor laboratory occupancy using near-real time image processing technique, leveraging the capabilities of a pretrained YOLO model and deepSORT algorithm.

Automating the detection and identification of individuals within the monitor space, streaming real time feedback from RTSP to capture visual outputs.

The pretrain YOLOv5 model is already known for its higher mean average precision (mAP) and compatibility through the DeepSORT algorithm. The system of this study will utilize YOLOv5mu.pt for detecting objects that have class id of 0 (person) together with DeepSORT algorithm to achieve versatile monitoring.

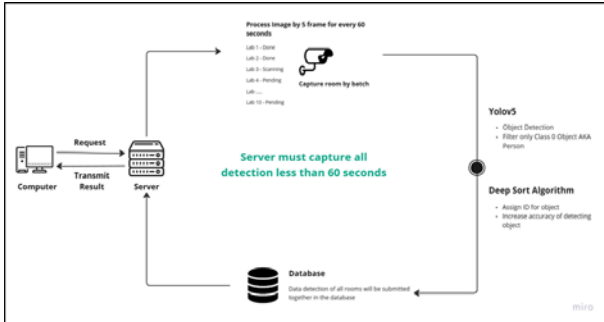


Fig. 6. System Occupancy and Detection flow

As depicted in figure 6, input from the camera stream that has been accessed through RTSP link will be collected and used to detect the person occupying the space. With 5 frame input per 1-minute detection on each laboratory, the system had enough real time data to transmit for web application input for laboratory monitoring.

### 3.3.1 Image Processing Error Filtering Object Detection

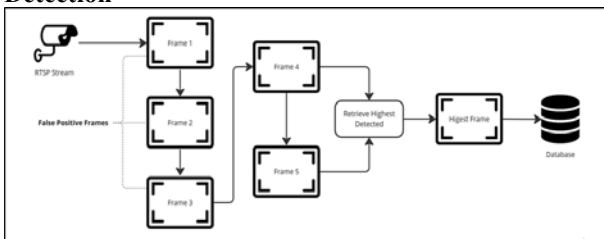


Fig. 7. Frame Detection Capture Algorithm

The detection process of frames captured in laboratories CCTV will be filtered based on the highest frame detection. Although detection is not fully accurate due to the limitation of computer vision in the field, the number of class-0 (person) in detection will be used as density of room occupancy. Moreover, to avoid false positive frame detection resulting from stream and internet interruption, the figure 7 illustrates the alternative of filtering frames to capture better and reliable results in capturing the highest number of people during the room detection.

### 3.3.2 Real-Time Streaming Protocol Integration

To effectively integrate existing CCTV installed in laboratories of the College of Computing and Information Science of Caraga State University. This study will use Real-Time Streaming Protocol method to integrate CCTV cameras effectively even if it's on LAN or Internet based connection.



fig. 8. Real-Time Streaming Protocol Link Format

As illustrated, the RTSP link is composed of 4 components. The protocol identifier (rtsp://) that indicates the URL of RTSP protocol contains username,

password, IP address, port and path. The account section such as username and password are based on the account register within the CCTV and the second section of the link is the IP address of the server hosting the RTSP stream the port number on the server where the RTSP stream is listening with path that navigate pacific stream or resource on the server together separated by (@) symbol navigate camera stream.

### 3.3.3 HIKvision RTSP Format

The system integration will leverage CCTV cameras installed in the Hiraya Hall Building of the College of Computing and Information Science. The integration will require the usage of RJ45 to achieve high quality and live feedback stream during the detection and monitoring stream of the server.

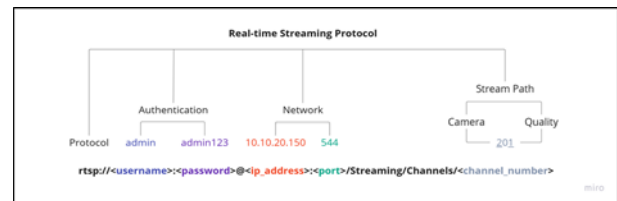


Fig 9. HIKvision RTSP Link Format

Similar to the format shown at figure 8, HIKvision format is different when it comes to path format except for authentication and network. Since four cameras are dependent on a single hub, stream integration requires more path information as four CCTVs are within the IP address of a single hub. As illustrated in Figure 9, the RTSP path format for HIKvision requires the channel number of the camera as it would be needed to indicate and retrieve specific camera streams and quality. With this alternative, the integration of the stream for the server will be effective while the object detection is expected to be reliable.

### 3.4 Requirement Analysis

The Near Real-Time Room Monitoring for Occupancy Detection System enables users to monitor rooms using image processing powered by YOLOv5mu and DeepSORT. To ensure successful implementation, the system's functional and non-functional requirements are identified throughout development.

Functional Requirements: The system must achieve at least 50% accuracy in detecting persons (class-0), provide near-real-time monitoring with object detection overlays, record image feeds and detection events, and offer an intuitive, user-friendly interface for accessing live feeds and logs.

Non-Functional Requirements: The system must ensure low-latency performance, secure data transmission and storage, accessible usability, broad camera compatibility, scalability for multiple monitoring points, and ease of maintenance for future upgrades.

### 3.4.1 Software Requirements

Table 1. Software Requirements

CLASSIFICATION	MINIMUM VALUE REQUIREMENT
----------------	---------------------------

<b>Front-end</b>	Html, CSS, JavaScript
<b>Back-end</b>	Python 3.9
<b>Database</b>	PostgreSQL, Supabase
<b>Integrated Development Environment</b>	Visual Studio Code
<b>Framework</b>	ReactJS, Tailwind CSS, FastAPI

Table 1, presents the classification of the software requirements into distinct categories: front-end, back-end, database, integrated development environment (IDE), and framework. These classifications mentioned have a specific minimum value requirement set by the research.

### 2.4.2 Hardware Requirements

**Table 2.** Hardware Requirement

CLASSIFICATION	MINIMUM VALUE REQUIREMENT
<b>OS Name</b>	Windows 11
<b>CPU</b>	AMD Ryzen 5 5625U
<b>GPU</b>	AMD Radeon Graphics
<b>Installed Physical Memory (RAM)</b>	8 GB
<b>System Storage</b>	459 GB SSD

Table 2. shows the computer hardware components alongside their respective minimum value requirement. These components are the operating system (OS), processor, graphics processing unit (GPU), installed physical memory (RAM), and system storage are based on researcher analysis and current specs of computers used during the development.

### 2.5 System Architecture

The diagram in Figure 10 shows the data flow and relationships in the research context. It highlights two external entities: Facility Management staff and the administrator. The staff monitors laboratory usage in real time, generating reports to identify unauthorized use. The admin, who oversees the system, can view and monitor rooms based on staff data and has the authority to add or modify room information and create user accounts.

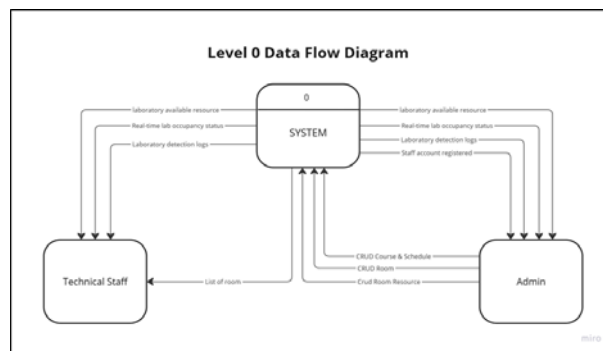


Fig. 10. Level 0 System Logical Data Flow Diagram

## 3. RESULTS AND DISCUSSION

### 3.1 Pretrained Model Detection Accuracy

To retrieve multiple streams in one cycle at the same time, the RTSP link of the camera has been processed and managed in batch to achieve optimized performance of the occupancy detection cycle. As the server is capable of handling asynchronous execution of script, multiple requests from data retrieval while cycle detection processes are executed will be done without having any conflict or stack pending request from the client to server. Moreover, the objective of retrieving near-real time data has been effective due to the distribution of stream frame feedback retrieval from streams.

Attempts	Environment status	Total	False Positive	False Negative	Precision	Recall	F1-score
1	Moderate Occupancy	7	1	0	0.87 (87.5%)	1 (100%)	0.93 (93.3%)
2	High Occupancy	23	1	2	0.95 (95.8%)	0.91 (91.3%)	0.93 (93.5%)
3	Active Session	23	0	2	1 (100%)	0.91 (91.3%)	0.95 (95.5%)
4	Low lighting	22	0	4	1 (100%)	0.81 (81.8%)	0.89 (89.9%)
5	Unoccupied	0	0	0	1 (100%)	1 (100%)	1 (100%)

Fig. 11. YOLOv5 Accuracy of Environment Status Classification

Figure 11 presents the evaluation of the pre-trained YOLOv5 model with DeepSORT for laboratory occupancy detection. The model's performance, measured by precision, recall, and F1-score, varies across different scenarios. In moderate occupancy (less than 10 people), the model achieves a 93.3% F1-score. In high occupancy scenarios, despite one false positive caused by partial obstructions or reflections, the model maintains a 92.5% F1-score. The model performs even better in active sessions, with a 95.5% F1-score, showing its ability to detect individuals in crowded, moving environments. In low-light and densely packed scenarios, the model achieves 89.9% F1-score, with only 4 false positives due to obstructions. Finally, in unoccupied rooms, the model performs flawlessly with a 100% F1-score, indicating its effectiveness for detecting unoccupied laboratory spaces. The model consistently surpasses the recommended 75% performance threshold in all conditions.

Batch	Second Duration
1st	39.36 seconds
2nd	38.11 seconds
3rd	40.54 seconds
4th	38.44 seconds
5th	40.26 seconds
Total Average	39.34 seconds.

Fig. 12. Room Distribution Stream Cycle

As shown in figure 12, the stream detection cycle, which involves monitoring data from eight cameras with five frame feedbacks per stream, achieves an average data retrieval time of 39.34 seconds across five batch attempt cycles. This performance aligns well with the project's objective of achieving near-real-time data processing. Given that the College of Computing and Information Sciences has eight laboratories to monitor and maintain, the system efficiently supports the simultaneous monitoring of all eight rooms, meeting the institution's monitoring requirements effectively.

### 3.2 Stream Image Feedback for Detection

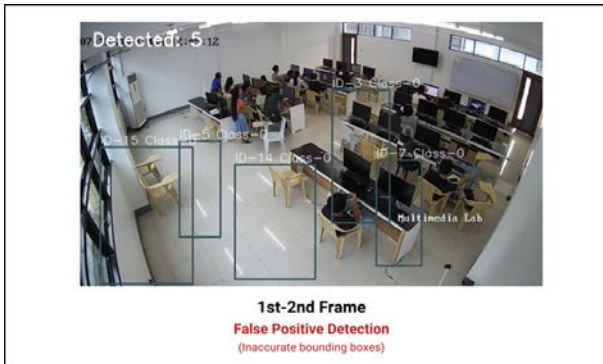


Fig. 13. False Positive Frames Detection Comparison

Figure 13 illustrates that initially, the first and second frames of the detection may produce false positives, resulting in random bounding boxes and repeated detections across different CCTV streams. However, as the frame count increases to 9, the detection improves, with bounding boxes accurately identifying the correct objects. This indicates enhanced detection accuracy, with positive results consistently captured in 3 to 5 out of the 5 stream frames.

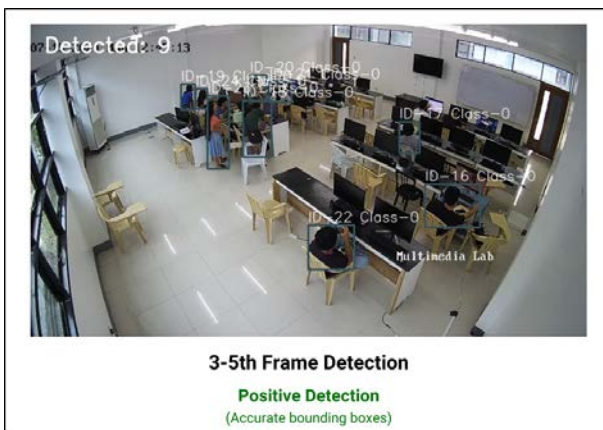


Fig. 14. False Positive Frames Detection Comparison

In figure 14, the feedback detection from 5 frames requires time to capture the most accurate detection count result. The distribution of 5 frames in a single batch of camera is designed to assist the model to recognize the object through frames leading to more accurate detection based on the number recognized object class-0 during the process.

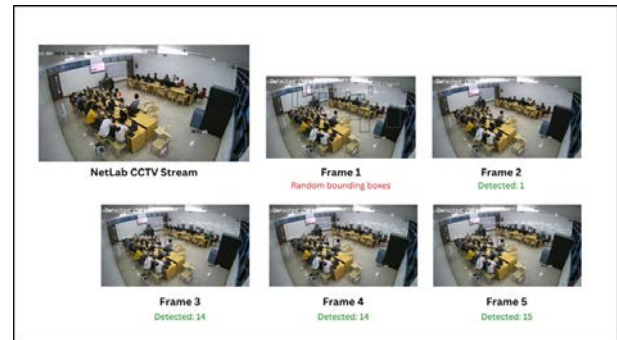


Fig. 15. Frame Detection Result

## 4. SUMMARY

The study developed a near-real-time laboratory occupancy monitoring system using YOLOv5mu and DeepSORT for accurate, non-redundant person detection across five frames per stream. The system achieved a detection accuracy of 90–95%.

It uses React.js and Vite for the front end, and FastAPI with Uvicorn for efficient stream processing and API handling. The average data retrieval time was recorded at 39.34 seconds across five batches, enabling reliable real-time monitoring of eight laboratories.

Initial frames occasionally produced false positives, but accuracy improved as more frames were processed, resulting in consistent and accurate occupancy logs. The system addresses previous challenges in manual monitoring due to limited staff by integrating with existing Hikvision RTSP streams and generating occupancy logs with delays under 60 seconds.

A System Usability Scale (SUS) test showed an average score of 75.75%, above the 68% benchmark. Positive feedback highlighted ease of use, integration, and user confidence. The highest satisfaction came from student assistants (92.5%), while some lower scores reflected scope limitations within the one-semester timeline.

Overall, the system meets monitoring needs, enhances operational efficiency, and provides valuable insights into room usage while reducing the burden on laboratory staff.

## 5. CONCLUSION

The researchers successfully developed a web-based room occupancy detection system for the College of Computing and Information Sciences, utilizing React.js for the front end and FastAPI for the back end, integrated with a pre-trained YOLOv5 lightweight model (yolov5mu) and the DeepSORT algorithm for accurate object detection and tracking. The system seamlessly connects to the CCIS computer laboratory cameras through Real-Time Streaming Protocol (RTSP), enabling real-time processing of video streams for effective monitoring. Results demonstrated the system's accuracy and reliability in detecting and tracking individuals while

minimizing redundant detections, thereby identifying discrepancies between actual room usage and scheduled allocations. This capability provides actionable insights for optimizing laboratory resource utilization. Overall, the study met its objective by delivering a practical and efficient near-real-time monitoring system that showcases the potential of computer vision technologies in addressing operational challenges in academic environments.

## 6. REFERENCES

Reference to a journal publication:

- [1] M. I. Azhar, F.H.K. Zaman, N.M. Tahir, & H. People tracking system using DeepSORT. In 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE) (pp. 137–141), 2022 <https://doi.org/10.1109/ICCSCE50387.2020.9204>
- [2] H. Elkhouchi, Y. NaitMalek, A. Berouine, M. Bakhouya, D. Elouadghiri, & M. Essaaidi. Towards a real-time occupancy detection approach for smart buildings. *Procedia Computer Science*, 134, 114-120. 2018. <https://doi.org/10.1016/j.procs.2018.07.151>
- [3] Y. D. S. Fidarliyan, A. B. Prasetijo, A. B., & D. Eridani. Classroom occupancy monitoring system using IoT device and the k-nearest neighbors algorithm. *Jurnal Teknik Elektro*, 14(2). 2022. <https://doi.org/10.15294/jte.v14i2.37141>
- [4] R. Hassan, & S. Salaudin. Implementation of IoT based smart classroom: Room occupancy estimation, automatic seat allocation, and environment monitoring. 2020. <http://hdl.handle.net/10361/14362>
- [5] P. W. Tien, S. Wei, J. K. Calautit, J. Darkwa, & C. Wood. Real-time monitoring of occupancy activities and window opening within buildings using an integrated deep learning-based approach for reducing energy demand. *Applied Energy*, 308, 118336. 2022. <https://doi.org/10.1016/j.apenergy.2021.118336>
- [6] N. S. Pun, S. K. Sonbhadra, S. Agarwal, & G. Rai. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. arXiv. 2021. <https://doi.org/10.48550/arXiv.2005.01385>
- [7] P. Rukmani, G. K. Teja, M. S. Vinay, & B. P. K. Reddy. Industrial monitoring using image processing, IoT and analyzing the sensor values using big data. *Procedia Computer Science*, 133, 991-997. 2018. <https://doi.org/10.1016/j.procs.2018.07.077>
- [8] B. Yang, S. Yang, X. Zhu, M. Qi, H. Li, X. Zhihan, X., Cheng, & F. Wang. Computer vision technology for monitoring of indoor and outdoor environments and HVAC equipment: A review. *Sensors*, 23(13), 6186. 2023. <https://doi.org/10.3390/s23136186>
- [9] Y. W. Syaifudin, I. F. Rozi, R. Ariyanto, & R. Erfan. Study of Performance of Real Time Streaming Protocol (RTSP) in Learning Systems. *International Journal of Engineering & Technology*, 7(4), 216-221. 2018. <https://DOI:10.14419/ijet.v7i4.44.26994>
- [10] J. W. Li, F. S. Yang, Y. C. Chang, & Y. L. Chiu. Integrated visual-perception real time monitoring system. Proceedings of the 2015 International Conference on Control, Simulation and Computing (CSSCC), Vienna, Austria. 2015. <https://www.inase.org/library/2015/vienna/bypaper/CSSCC/CSSCC-22.pdf>
- [11] P. Jiang, D. Ergu, F. Liu, Y. Cai, & B. Ma. A review of YOLO algorithm developments. *Procedia Computer Science*, 199, 1066–1073. 2022. <https://doi.org/10.1016/j.procs.2022.01>
- [12] T. Xie & X. Yao. Smart logistics warehouse moving-object tracking based on YOLOv5 and DeepSORT. *Applied Sciences*, 13(17), Article 9895. 2023. <https://doi.org/10.3390/app13179895>
- [13] M. Moin, “What is DeepSORT and how to implement YOLOv7 object tracking using DeepSORT”, Medium, Website: <https://medium.com/@m.moinfaisal/what-is-object-tracking-and-why-deepsort-and-how-to-implement-yolov7-object-tracking-using-deepsort-f0c952f89b06> (Accessed: 01-5-2025)
- [14] Q. Yu, M. Cheng, W. Zhang, Y. Tan, & T. Huang. Tiny object detection with lightweight YOLOv5 based on attention mechanism. 2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (ACAIT), 10138007. 2022 <https://doi.org/10.1109/ACAIT56212.2022.10138007>
- [15] J. Ren, Z. Wang, Y. Zhang & L. Liao. YOLOv5-R: Lightweight real-time detection based on improved YOLOv5. *Journal of Electronic Imaging*, 31(3), 033033. 2022 <https://arxiv.org/pdf/1506.02640v5>
- [16] J. Zhou, Z. Ming, G. Ding, S. Zhao, T. Furukawa, and K. Lu, “The RFID Electronic Tag Defect Detection System Based On YOLOv5 For Industrial Scenarios,” *Proceedings of international exchange and innovation conference on engineering & sciences (IEICES)*, vol. 8, pp. 44–51, Oct. 2022, doi: 10.5109/5909061.
- [17] M. Zhao, J. Ding, G. Zhou, S. Du, T. Furukawa, and K. Lu, “A Two-model Automatic Towel Defect Detection Method Based On YOLOv5,” *Proceedings of international exchange and innovation conference on engineering & sciences (IEICES)*, vol. 8, pp. 386–393, Oct. 2022, doi: 10.5109/5909122.