IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Based Lost And Found Monitoring System With CCTV Integration

YASIR QURESHI, ABDUL KAREEM, MOIN FAROOQUI, SAAD MANSOORI, AND PROF. AAKANSHA TIWARI

Abstract: With the rapid growth of CCTV surveillance systems in both public and private environments, a massive amount of video data is continuously generated. However, identifying lost or unattended objects from this footage remains a manual, time-consuming, and inefficient task. This paper presents an AI-based Lost and Found Monitoring System that automatically detects unattended objects using computer vision techniques. The system utilizes YOLOv8 for real-time object detection and ByteTrack for tracking objects across consecutive frames. A state-based logical model is implemented to analyze human–object interaction and determine whether an object is attended or unattended over time. Instead of focusing on identifying object ownership, the system emphasizes detecting loss situations based on spatial proximity and temporal behavior. The proposed system provides useful outputs such as snapshots, timestamps, and event logs, which assist security personnel in locating lost items efficiently. The system is lightweight, practical, and suitable for real-time deployment using existing CCTV infrastructure.

Keywords*: **CCTV*** surveillance, ***YOLOv8, ByteTrack,*** object detection, object tracking, lost and found monitoring system.

I. INTRODUCTION

CCTV SURVEILLANCE SYSTEMS ARE WIDELY DEPLOYED IN PUBLIC PLACES SUCH AS RAILWAY STATIONS, AIRPORTS, SHOPPING MALLS, EDUCATIONAL INSTITUTIONS, AND OFFICES. THESE SYSTEMS CONTINUOUSLY RECORD VIDEO DATA FOR SAFETY AND MONITORING PURPOSES. HOWEVER, IN MOST CASES, THIS RECORDED FOOTAGE IS ONLY REVIEWED AFTER AN INCIDENT OCCURS AND IS RARELY UTILIZED FOR IDENTIFYING LOST OR MISPLACED ITEMS.

IN REAL-WORLD SCENARIOS, PEOPLE FREQUENTLY FORGET PERSONAL BELONGINGS SUCH AS BAGS, WALLETS, OR MOBILE PHONES IN PUBLIC PLACES. THE CURRENT PROCESS OF RECOVERING THESE ITEMS INVOLVES MANUALLY REVIEWING CCTV FOOTAGE, WHICH IS TIME-CONSUMING, INEFFICIENT, AND PRONE TO HUMAN ERROR. SECURITY PERSONNEL OFTEN NEED TO ANALYZE HOURS OF VIDEO TO IDENTIFY THE LOCATION AND TIME AT WHICH AN ITEM WAS LEFT BEHIND.

With recent advancements in artificial intelligence and computer vision, it is now possible to automate the analysis of video data. Object detection and tracking techniques can identify and monitor objects across video frames. However, most existing systems are focused on security-related tasks such as intrusion detection or face recognition, while limited attention has been given to practical lost and found monitoring systems.

This paper proposes an AI-based Lost and Found Monitoring System that works with existing CCTV infrastructure. The system focuses on detecting unattended objects using a state-based logical approach, making it efficient, reliable, and suitable for real-world applications.

II. System Overview

The proposed system operates on CCTV video streams, either from live camera feeds or recorded footage, and follows a structured processing pipeline to detect and monitor objects.

The system begins by capturing video input from CCTV cameras. The video stream is divided into frames and processed using OpenCV to ensure consistent input quality. Each frame is then passed to the object detection module, where YOLOv8 identifies objects such as people and bags by generating bounding boxes and class labels.

After detection, ByteTrack is used for multi-object tracking. This module associates detected objects across consecutive frames and assigns temporary tracking IDs. However, since tracking IDs may change due to occlusion or detection noise, the system maintains a logical object memory that uses spatial distance and temporal consistency to preserve stable object identities.

A state-based decision model is implemented to determine the condition of each object. Based on human–object interaction, objects are classified into different states such as attended, potentially unattended, unattended, and resolved. When an object is confirmed as unattended, the system captures a snapshot and records event details including timestamp, camera ID, and object information in a database.

A Flask-based backend manages data processing and communicates with a web-based dashboard. The dashboard displays detected events, snapshots, and alerts, enabling efficient monitoring and analysis.

III. Methodology

The system follows a step-by-step methodology to detect unattended objects from CCTV footage in real time.

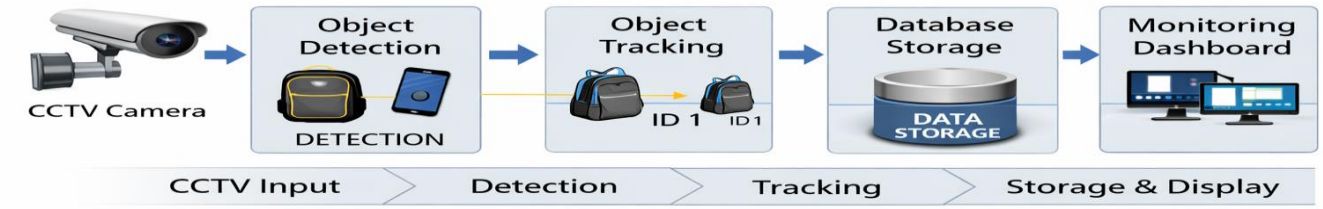
Initially, video input is captured either through a live RTSP CCTV stream or a recorded video file. OpenCV is used for frame extraction and preprocessing. The object detection module uses YOLOv8, which is initialized with pre-trained COCO weights and can be fine-tuned using a custom dataset to improve performance in CCTV environments.

Once objects are detected, ByteTrack is used to track them across frames using spatial and temporal association. Since tracking IDs are not permanent and may change, the system uses logical object memory to maintain consistent object identity based on distance and time.

The system applies a state-based model to detect unattended objects. Each object transitions through different states: attended, potentially unattended, unattended, and resolved. When a person is near the object, it is considered attended. If the person moves away, the system begins monitoring and marks the object as potentially unattended. If the object remains unattended beyond a defined time threshold, it is classified as unattended.

To improve reliability, the system incorporates temporal consistency, stationary verification, and person-presence memory. Alerts are triggered only when conditions are consistently satisfied, which helps reduce false positives. When an unattended object is detected, a snapshot is captured and the event is logged for further analysis.

Fig. 1 illustrates the overall system workflow, showing the flow of data from CCTV input to object detection, tracking, storage, and monitoring.



IV. EXPERIMENTAL SETUP AND RESULTS

This section presents a brief overview of the experimental setup and observed results for the implementation of the proposed system. The experiments were conducted using CCTV video footage to evaluate the system's ability to detect, track, and monitor objects in a surveillance environment. The system was evaluated qualitatively on CCTV video scenarios under different environmental conditions. The trained model demonstrated satisfactory performance under varying lighting and environmental conditions.

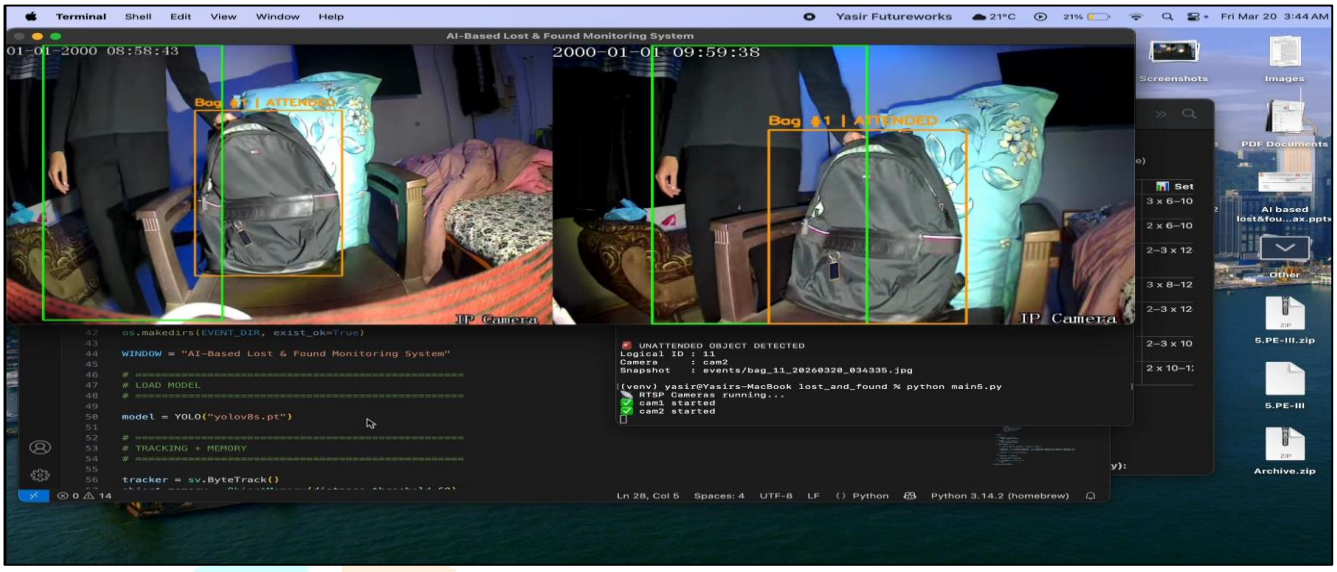
The system was tested on sample video streams captured from indoor and outdoor CCTV cameras. Object detection and tracking were performed in real time, and detected objects were successfully monitored across multiple frames. The system was able to maintain consistent object tracking behavior and record object appearance information effectively.

The experimental setup involved processing video streams in real time and observing the system's ability to detect and track objects consistently. The object detection module successfully identified multiple objects within the scene, while the tracking module maintained stable object identities across frames.

The system successfully identified unattended object scenarios by analyzing human–object interaction patterns. In test cases where a person left an object behind, the system correctly transitioned the object state and generated an unattended event with visual evidence.

The use of temporal thresholds and stationary verification reduced false positives and improved stability across different video conditions.

Fig. 2 shows sample output results of object detection and tracking on CCTV footage, highlighting the system's ability to monitor multiple objects simultaneously.



V. CONCLUSION

This paper presented an AI-based Lost and Found Monitoring System integrated with CCTV surveillance. The system combines object detection, tracking, and logical decision-making to automatically detect unattended objects.

By reducing the need for manual inspection of CCTV footage, the system improves operational efficiency and assists in the quick recovery of lost items. The use of a state-based model ensures stable and reliable performance in real-world scenarios.

Future work may include improving detection accuracy through dataset enhancement, optimizing system performance, and developing a more advanced dashboard for better visualization and user interaction

VI. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Prof. Ram Meghe College of Engineering and Management for providing the necessary academic support and resources. We would also like to thank our guide, Prof. Aakansha Tiwari, for her continuous guidance, valuable suggestions, and encouragement throughout the development of this project.

VII. REFERENCES

[I] J. Redmon and A. Farhadi, "YOLO: You Only Look Once: Unified, real-time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.
[II] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, " Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[III] G. Jocher, A. Chaurasia, and J. Qiu,
"YOLOv5 by Ultralytics," GitHub repository, 4

2020. [Online]. Available: https://github.com/ultralytics/yolov5
[IV] G. Jocher and Ultralytics Team, "YOLOv8 by Ultralytics," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[V] Y. Zhang, P. Sun, Y. Jiang, et al., "ByteTrack: Multi-object tracking by associating every detection box," Proceedings of the European Conference on Computer Vision (ECCV), pp. 1–17, 2022.

[VI] W. Luo, X. Zhao, and T. Kim, "Multiple object tracking: A review," arXiv preprint arXiv:1409.7618, 2014.

[VII] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[VIII] A. Ronacher, "Flask Web Development Framework," Pallets Projects, 2010. [Online]. Available: https://flask.palletsprojects.com
[IX] J. Dwyer, J. Nelson, and Roboflow Team, "Roboflow: Dataset Management and Annotation Tool," 2023. [Online]. Available: https://roboflow.com