International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Powered Video Surveillance for Object Detection and Tracking

Guide - Prof. Omkar Kolate[1]
Saquib Nasir Sayyed[2]
Suraj Sudhakar Patil[3]
Shree Shekhar Deore[4]
Krish Ashok Dalvi[5]
Department of Artificial Intelligence and Machine Learning,
Zeal Polytechnic, Pune, Maharashtra, India

ABSTRACT

Closed-circuit television (CCTV) systems capture a large amount of video data, which makes manual review of footage a time-consuming process. This paper introduces a proposed design for identifying specific objects in CCTV recordings and tracking their movement across different frames. The system framework integrates an object detection module, a multi-object tracker, trajectory recording functionality, and configurable event rules. The paper provides an overview of existing research on intelligent video surveillance, object detection, multi-object tracking, and video anomaly detection. It outlines the proposed workflow, possible benefits, constraints, privacy protections, and a plan for evaluation. The intended output includes annotated video clips, track identifiers, approximate movement paths on the image plane, timestamps, and a searchable event log. This system has not been tested in this paper, so no claims regarding accuracy, speed, or other performance metrics are made. Its effectiveness can only be assessed using official representative videos, documented ground truth annotations, detailed configurations, and appropriate measures related to detection, tracking, runtime, and event-level performance.

Index Terms— CCTV, object detection, multi-object tracking, trajectory analysis, video surveillance.

I. INTRODUCTION

CCTV systems are widely used in various locations such as shops, campuses, offices, parking areas, factories, and public spaces. Traditional setups typically involve recording video for later review by a human operator. Examining long recordings or footage from multiple cameras can be very time-intensive. Computer vision can help by identifying objects within individual frames and linking these identifications over time to form continuous tracks.

Object detection involves identifying the type of an object and placing a bounding box around it in a frame. Object tracking connects these identifications across different frames to estimate the movement of an object and assign a unique track ID for as long as the connection remains accurate. These actions can assist an operator in locating important time intervals, reviewing the path of an object, and examining specific events. A track ID is a temporary label assigned by the system; it does not reveal someone's real-world identity or confirm their responsibility for an incident.

This paper outlines a proposed design for CCTV movement monitoring and presents a plan for evaluating its functionality. The goal is to organize object detections, movement details, and event records with timestamps to aid human review. The design is not presented as a completed or experimentally validated system.

II. LITERATURE REVIEW

A. Intelligent video surveillance

Karbalaie et al. examine event detection in surveillance settings and highlight the difficulties of identifying events from complex video footage [1]. Ben Mabrouk and Zagrouba discuss the recognition of abnormal behaviors in intelligent video surveillance, detailing how scene context and feature representation influence the process [2]. A review of edge-based video anomaly detection explores the potential advantages and limitations of processing data close to the source [3]. These studies encourage the separation of low-level detection and tracking tasks from higher-level interpretation of events.

B. Detection and tracking methods

Simple Online and Realtime Tracking (SORT) merges detection with motion prediction and data association [4]. Its straightforward design makes it a useful starting point, although missed detections and object occlusion can disrupt tracking. DeepSORT enhances association between frames by incorporating appearance descriptors [5]. ByteTrack uses lower-confidence detections in a secondary association step to improve the recovery of objects that might be missed [6]. These techniques involve different trade-offs in terms of computational efficiency, appearance modeling, and track continuity, and their performance should be compared under uniform test conditions.

C. Deep-learning-based surveillance

A study by Yadav et al. describes an approach to object detection and tracking using YOLOv8 and DeepSORT [7]. PASS-CCTV addresses proactive anomaly detection in challenging environmental conditions [8]. These studies demonstrate how object detection and tracking can support application-specific event rules. However, a detection and tracking pipeline alone is insufficient as a complete anomaly detection system. Unusual behavior requires a clearly defined event, context-specific rules, or a separately trained model, along with validation using representative footage.

D. Recent reviews and project motivation

Park et al. survey deep-learning methods for multiple-object tracking [9], while Guan et al. examine multi-object tracking methods and emerging trends [10]. A 2025 review explores recent developments in multi-object tracking, evaluation measures, datasets, and technical challenges [11]. A systematic review of video anomaly detection addresses issues related to datasets, evaluation, and generalization [12].

The existing literature shows that detection, association, and event interpretation remain active areas of research. Based on this, this paper does not claim that object tracking or event logging constitutes a novel research contribution.

The project aims to introduce a clear and organized process that brings together object detection, tracking identifiers, estimated movement paths, adjustable event rules, and time-stamped records all within a single tool designed for reviewing video footage. Whether this approach is cost-effective, precise, or beneficial should be confirmed through actual development and testing.

Table I. Summary of selected research

Study	Main focus	Relevance to this design
[1], [2]	Surveillance events and abnormal behaviour	Context for event interpretation
[3]	Edge-based video anomaly detection	Motivates local processing; resource limits remain
[4], [5]	SORT and DeepSORT	Tracking baseline and appearance-based association
[6]	ByteTrack data association	Uses lower-confidence detections in association
[7], [8]	YOLOv8 + DeepSORT; proactive CCTV	Examples of detection/tracking and event applications
[9]–[12]	Tracking and anomaly-detection reviews	Context for method trade-offs and evaluation challenges

III. PROBLEM STATEMENT AND OBJECTIVES

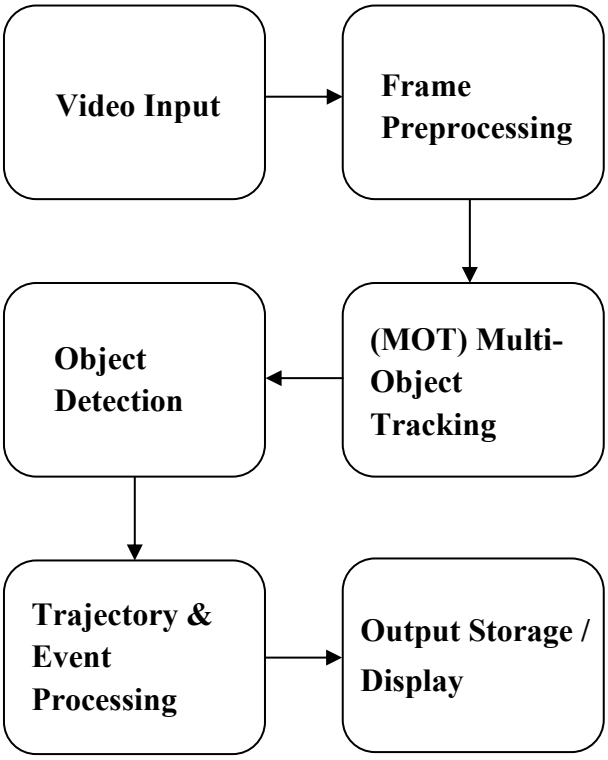
This project seeks to reduce the time and effort needed to review CCTV footage and identify meaningful object movements. Manual reviews can be slow and might miss short or subtle events. A tracking system must also handle lost detections, partial object blockage, changes in lighting, busy scenes, and variations in object appearance.

The objectives of the project are: (1) to identify specific object types in each video frame; (2) to link these object detections across frames and assign tracking IDs; (3) to record estimated paths within the image and corresponding times; (4) to produce an annotated video and a searchable log of events; (5) to assess the quality of detection, the continuity of tracking, the speed of processing, and the performance of event rules; and (6) to handle data in a way that respects privacy and allows for human review. The specific object categories, confidence thresholds, tracker settings, and testing procedures are decisions to be made during implementation and must be documented before evaluation.

IV. PROPOSED SYSTEM

A. System architecture

The proposed system consists of six main components: video input, frame preprocessing, object detection, multi-object tracking, trajectory and event processing, and output storage or display. Input can be a recorded video or an authorized live CCTV feed. Each frame is processed by a selected pre-trained detector, such as a YOLO-family model. A detection includes a class label, a bounding box, and a confidence score. A tracker links these detections across frames and establishes a continuous track. Subsequently, the system logs the time of occurrence, object category, tracking identifier, center coordinates of the bounding box, and the results of any triggered event rules. Specific details such as the versions of the detector and tracker, model weights, confidence limits, frame dimensions, policies for skipping frames, and the hardware used are not detailed in this document. These parameters must be carefully chosen and documented to ensure the implementation can be reproduced.



B. Workflow for Processing

Step 1—Input Phase: Import a video file and determine its resolution and frame rate.

Step 2—Detection Phase: Execute the detection model on every analyzed frame, applying a specified confidence limit.

Step 3—Tracking Phase: Link new detections to established paths utilizing an appropriate tracking algorithm like ByteTrack or DeepSORT.

Step 4—Movement Analysis: Compute the center point of each bounding box and add these coordinates to the track's positional history.

Step 5—Event Processing: Optionally trigger alerts for specific activities, such as entering a restricted area, traversing a digital boundary, or loitering in a specific sector for longer than allowed.

Step 6—Output Generation: Visually overlay class labels, bounding boxes, trajectories, and tracking IDs onto the video, while simultaneously exporting a time-stamped log to a database or CSV file.

C. Representation of Movement

Given a bounding box defined by its top-left corner (x, y), along with its height h and width w, the center point can be calculated as c_x = x + w/2 and c_y = y + h/2. The spatial shift between a pair of consecutive observations (labeled 1 and 2) is computed using the distance formula:

$$d = \sqrt{[(c_{x2} - c_{x1})^2 + (c_{y2} - c_{y1})^2]}.$$

This value indicates the pixel-based shift within the image array. By aggregating these shifts over consecutive frames, one can estimate the overall trajectory length within the

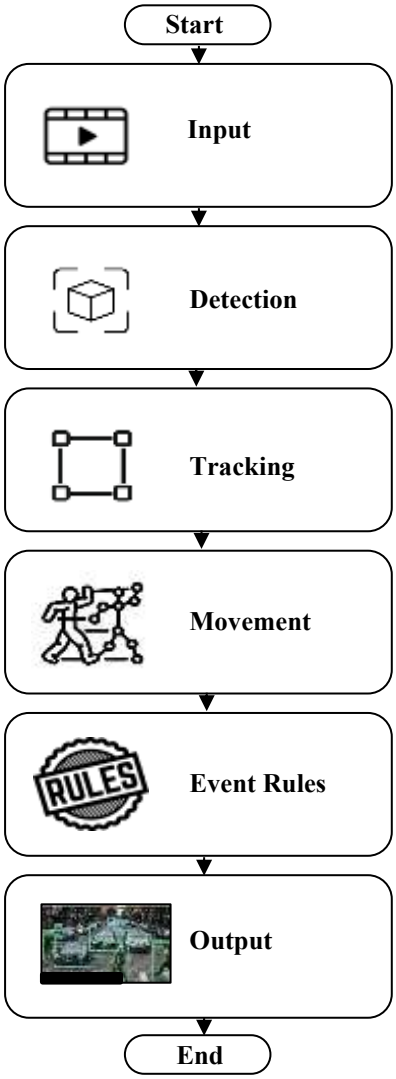


image plane. It is important to note that these pixel measurements do not equate to real-world distances unless the scene geometry is mapped and the camera is properly calibrated. Furthermore, time logs should be calculated based on the frame rate and the specific frame index, rather than relying on the assumption of a uniform processing rate.

D. User Interface and Event Documentation

The graphical interface is designed to present the augmented video feed alongside a concise log that details the track ID, timestamp, event category, and the specific region involved. The system should allow operators to sort this log by area, object class, or time, facilitating quick navigation to pertinent video clips. Notifications must focus on describing objective occurrences—such as "Object 12 crossed Boundary A at 14:03:12"—rather than making assumptions about a subject's intentions or classifying them as suspicious. Parameters like time tolerances for events, maximum dwell durations, and the spatial dimensions of regions must be clearly defined prior to conducting any event-focused evaluations.

V. PROSPECTIVE BENEFITS

This proposed framework has the potential to accelerate the review of video archives by enabling operators to easily search through tracking IDs and timestamps. It offers visual representations of movement, monitors multiple objects simultaneously, archives organized data for subsequent analysis, and allows for the customization of event triggers like loitering times, unauthorized zone access, and digital line crossings. The modular nature of the detection and tracking components facilitates seamless upgrades as technical needs shift. Furthermore, executing these processes locally could minimize the bandwidth required to send raw video data to cloud servers, contingent on the chosen deployment strategy and hardware capabilities. It is crucial to emphasize that these are anticipated benefits rather than empirically validated results within this study.

VI. STRATEGY FOR IMPLEMENTATION AND ASSESSMENT

An initial prototype could be developed using the Python programming language, leveraging OpenCV to process video streams, a pre-trained model from the YOLO series for identifying objects, and a tracking algorithm like ByteTrack or DeepSORT. Recorded tracking data and event timestamps could be preserved in an SQLite database or a simple CSV format. To ensure the research can be reproduced, all technical specifications—including software and model versions, specific weights, confidence cutoffs, processing frame rates, input resolutions, hardware specifications, and tracker settings—must be meticulously documented.

Testing procedures should rely on legally sourced, highly representative video datasets equipped with accurate manual annotations. This testing environment must encompass a variety of real-world challenges, including different camera angles, partial blockages, fluctuating crowd sizes, and variable lighting conditions. At present, this document does not define a specific baseline, hardware setup, selected model configuration, annotation guidelines, or dataset volume; these elements must be thoroughly defined prior to publishing any empirical findings.

Table II. Suggested evaluation checklist

Aspect	Possible measures	Details to report
Object detection	Precision, recall, mAP	Per class and test condition
Tracking	IDF1, HOTA and/or MOTA; ID switches; fragmentation	Track continuity and missed tracks
Runtime	Frames per second (FPS), processing latency	Hardware, input resolution, model version
Event rules	Event-level precision/recall, false alarms, detection delay	Rule definitions, ground truth, time tolerance

The accuracy of object detection can be quantified via mean average precision (mAP), recall, and precision metrics, provided that the dataset contains appropriate bounding box labels and classifications. The effectiveness of the tracking mechanism can be gauged utilizing metrics such as MOTA, HOTA, and IDF1, alongside measurements of track breaks and identity swaps. Processing efficiency must be recorded in terms of latency and frames per second on the specified computing hardware. The reliability of event rules should be tested against metrics like detection delays, false alarms, and event-focused precision and recall. It is imperative to transparently report all aggregation techniques, matching criteria, and metric formulas. Since this system remains in the design phase, no statistical or numerical outcomes are presented in this document.

VII. CONSTRAINTS, PRIVACY, AND ETHICAL IMPLICATIONS

Automated tracking systems are inherently flawed. Objects with similar appearances might accidentally swap identifiers, while visual obstructions can easily fragment established tracks. Furthermore, the reliability of detection can be severely compromised by dense crowds, camera shaking, heavy shadows, adverse weather (like rain or glare), and poor video resolution. It is crucial to remember that a system-assigned track ID is merely a transient marker, not a confirmation of a person's actual identity. Movement tracked on a two-dimensional screen does not confirm intent or exact physical traversal distances. Additionally, automated event rules are prone to generating both false negatives and false positives, necessitating rigorous testing before real-world deployment.

Implementing such a system must strictly adhere to relevant legal frameworks and organizational guidelines. Suggested privacy protections involve establishing a firm data retention schedule, utilizing robust encryption, securing physical and digital storage, limiting data access, and explicitly stating the surveillance purpose. Technologies like facial recognition or advanced identity estimation must never be integrated without explicit legal approval and independent ethical justification. Whenever feasible, video processing should occur locally, archiving only the absolute minimum amount of metadata required. Finally, human supervisors must evaluate any system-generated alerts before any significant actions are taken. These statements serve as best-practice guidelines; this paper makes no assertion that a fully audited, secure system has been successfully deployed.

ACKNOWLEDGEMENT

We wish to convey our profound gratitude to our project guide, Prof. Omkar Kolate, for his continuous encouragement, expert insights, and essential support throughout the course of this research. His constructive feedback played a critical role in refining the proposed architecture. We are also deeply appreciative of the faculty and administration at the Department of Artificial

Intelligence and Machine Learning at Zeal Polytechnic, Pune, for supplying the academic resources and atmosphere needed to develop this study. Lastly, we offer our heartfelt thanks to our peers and families for their unwavering motivation during the creation of this manuscript.

REFERENCES

[1] A. Karbalaie, F. Abtahi, and M. Sjöström, "Event detection in surveillance videos: a review," Multimedia Tools and Applications, vol. 81, pp. 35463–35501, 2022, doi: 10.1007/s11042-021-11864-2.

[2] H. Ben Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," Expert Systems with Applications, vol. 91, pp. 480–491, 2018, doi: 10.1016/j.eswa.2017.09.029.

[3] D. R. Patrikar and M. R. Parate, "Anomaly detection using edge computing in video surveillance system: review," International Journal of Multimedia Information Retrieval, vol. 11, no. 2, pp. 85–110, 2022, doi: 10.1007/s13735-022-00227-8.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468, doi: 10.1109/ICIP.2016.7533003.

[5] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645–3649, doi: 10.1109/ICIP.2017.8296962.

[6] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in Proceedings of the European Conference on Computer Vision (ECCV), 2022, doi: 10.1007/978-3-031-20047-2_1. [Verify final proceedings page range if required by the journal.]

[7] A. Yadav, A. Chaturvedi, and S. Rani, "Object detection and tracking using YOLOv8 and DeepSORT," in Advancements in Communication and Systems, 2024, pp. 81–90, doi: 10.56155/978-81-955020-7-3-7. [Verify metadata against the publisher record.]

[8] H. Jeon, H. Kim, D. Kim, and J. Kim, "PASS-CCTV: Proactive anomaly surveillance system for CCTV footage analysis in adverse environmental conditions," Expert Systems with Applications, vol. 254, art. 124391, 2024, doi: 10.1016/j.eswa.2024.124391.

[9] Y. Park, L. M. Dang, S. Lee, D. Han, and H. Moon, "Multiple object tracking in deep learning approaches: A survey," Electronics, vol. 10, no. 19, art. 2406, 2021, doi: 10.3390/electronics10192406.

[10] Z. Guan et al., "Multi-object tracking review: retrospective and emerging trend," Artificial Intelligence Review, vol. 58, art. 235, 2025, doi: 10.1007/s10462-025-11212-y.

[11] S. Li, H. Ren, X. Xie, and Y. Cao, "A review of multi-object tracking in recent times," IET Computer Vision, vol. 19, no. 1, art. e70010, 2025, doi: 10.1049/cvi2.70010.

[12] Y. A. Samaila, P. Sebastian, N. S. S. Singh, A. N. Shuaibu, S. S. A. Ali, T. I. Amosa, G. E. Mustafa Abro, and I. Shuaibu, "Video anomaly detection: A systematic review of issues and prospects," Neurocomputing, vol. 591, art. 127726, 2024, doi: 10.1016/j.neucom.2024.127726.