WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

QUITE WATCH: REAL-TIME PUBLIC TRANSPORT SAFETY DETECTION

[1]Ms. S. H. Chaflekar, [2]Prachi Nichat, [2]Sakshi Channe, [2]Khushi Meshram, [2]Sejal Dantulwar

[1]Assistant Professor, Department of Information Technology, Priyadarshini Bhagwati College of Engineering, Nagpur, Maharashtra, India
[2]UG Student, Department of Information Technology, Priyadarshini Bhagwati College of Engineering, Nagpur, Maharashtra, India

ABSTRACT:

Public transportation systems carry large numbers of passengers, yet an emergency inside a vehicle may go unnoticed until a passenger raises an alarm or an operator reviews recorded footage. This paper presents Quiet Watch, a proposed intelligent passenger-safety monitoring system intended to support earlier identification and review of potentially unsafe events. The design accepts live or recorded video, samples frames, applies an AI-based event-analysis module, and forwards candidate incidents to an alert engine and a web dashboard. The planned software stack comprises React.js for the user interface, Node.js and Express.js for APIs, MongoDB for incident records, and Python with OpenCV and a suitable machine-learning model for video processing. Alerts are assigned a priority and presented to an authorised human operator, who can review evidence, confirm or dismiss an event, and maintain an audit trail. A citizen-reporting option can supplement camera-based detection where appropriate. The paper describes the problem, objectives, related technical foundations, architecture, evidence management, safeguards, limitations and future work. Quiet Watch is presented as a system design and prototype direction; no claim of field-proven accuracy or real-world emergency response performance is made. Evaluation using representative, consented data and human review is required before operational deployment.

Keywords: Passenger safety, public transportation, computer vision, anomaly detection, real-time alerting, incident management, privacy, human-in-the-loop AI

I. INTRODUCTION

Buses, trains and other public transportation services are essential to daily mobility. However, crowding, limited visibility, delayed reporting and the difficulty of continuously observing multiple camera feeds can make it challenging to recognise an incident quickly. Conventional CCTV is valuable for investigation, but footage may be reviewed only after an event has been reported. An intelligent monitoring workflow can help operators notice selected visual cues sooner, organise relevant evidence and coordinate a timely human response [8].

Quiet Watch proposes a software-based monitoring and incident-management platform that analyses video for predefined indicators of possible emergencies—for example, a fall, prolonged immobility, a physical altercation or an unusual crowd movement—subject to the capabilities and validation of the selected model. It does not determine guilt, intent or the existence of a crime. A model output is a signal for review, not a final judgment. The system is designed around three principles: early awareness, accountable human decision-making and privacy-conscious evidence handling [9].

The prototype combines a React.js dashboard, a Node.js/Express API, MongoDB for incident storage, and a Python/OpenCV processing service. A WebSocket channel may deliver live status updates while REST endpoints manage incidents and configuration [10]. The design supports recorded-video testing first; live CCTV integration depends on camera access, network conditions, authorisation and local policy.

II. LITERATURE REVIEW

Title of paper	Method/ Techniques used	Analysis and Observation
1. A Passenger Detection and Action Recognition System for Public Transport Vehicles (2024)	Uses convolutional neural networks (CNNs), multiple camera views, passenger detection, pose recognition, action classification, and region prediction.	The study demonstrates how computer vision can detect passengers and recognize their actions inside public transport vehicles. Multiple camera views help cover passenger areas and reduce duplicate counting.
2. Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey (2023)	Reviews deep learning approaches, feature extraction, video preprocessing, human activity recognition, and anomaly detection techniques.	The study explains how AI can identify unusual activities in surveillance videos. It also highlights challenges such as poor lighting, overlapping people, background changes, and limited training data.

3. Recent Advances in Video Analytics for Rail Network Surveillance for Security, Trespass and Suicide Prevention—A Survey (2022)	Reviews CCTV video analytics, object detection, human behaviour recognition, anomaly detection, tracking, and crowd analysis.	The paper discusses how automated surveillance can support public safety and help identify potentially dangerous events. It also highlights the importance of reliable video processing, privacy, and human oversight.
4. A Critical Analysis on Machine Learning Techniques for Video-Based Human Activity Recognition of Surveillance Systems: A Review (2024)	Reviews machine learning and deep learning techniques, including CNN, recurrent neural networks (RNN), hidden Markov models (HMM), and K-means clustering.	The study compares approaches for recognizing human activities from video. It highlights the importance of feature extraction, model selection, and performance evaluation. These techniques provide a foundation for exploring activity recognition in Quiet Watch.
5. Evidential Event Inference in Transport Video Surveillance (2016)	Uses a framework for combining visual observations, associating detected events with individual subjects, and analysing relationships between events over time.	The study addresses the difficulty of interpreting noisy or incomplete video detections.

III. METHODOLOGY

The methodology describes the systematic approach adopted to design and develop Quiet Watch, an intelligent passenger-safety monitoring system for public transportation [11]. The proposed system combines computer vision, artificial intelligence (AI), web development, and database management to support the identification and management of potential safety incidents.

The methodology is divided into multiple phases, beginning with video acquisition and preprocessing, followed by AI-based event detection, alert generation, incident storage, and dashboard-based monitoring [23]. The system is designed to provide authorized personnel with a centralized interface for reviewing potential incidents and maintaining incident records.

The main technologies include Python and OpenCV for video processing, an AI/ML model for event detection, Node.js and Express.js for backend services, MongoDB for data storage, and React.js for the user interface [22].

1. SYSTEM ARCHITECTURE AND WORKFLOW

The architecture is organized as a pipeline. Video may arrive from an authorised CCTV stream or a recorded file. A preprocessing service samples frames and normalises them for analysis [12]. The AI service produces a candidate event label and confidence score; thresholding and optional temporal rules reduce isolated noisy predictions. The alert engine assigns a configurable priority and sends an event to the backend. The backend stores metadata and controlled references to evidence in MongoDB [8]. The React dashboard receives updates and enables an authorised operator to review and change incident status.

Workflow stages:

1. **Input:** Receive a permitted video stream or upload a test recording; associate it with a vehicle/feed identifier and timestamp.
2. **Preprocessing:** Sample frames at a configurable interval, resize frames and handle missing or corrupt input.
3. **AI analysis:** Run a selected model for the supported event classes; apply confidence thresholds and, where feasible, temporal consistency checks.
4. **Alerting:** Create a candidate incident with event type, time, feed identifier, confidence, priority and evidence reference.
5. **Review:** An operator opens the incident, reviews the relevant clip/frame, and confirms, dismisses or escalates it [24].
6. **Storage and analytics:** Save status transitions and authorised actions; produce summaries only from recorded events and validated labels.

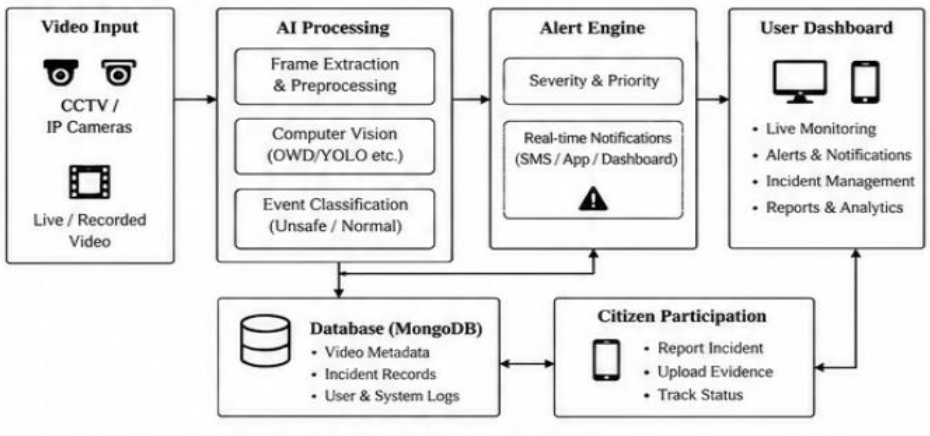


Fig.3. 1. Proposed System Architecture of Quite Watch.

2. PHASES AND MODULES OF THE PROPOSED SYSTEM

Module 1: Video Input and Acquisition-Collects live CCTV streams or recorded video from public transportation for safety monitoring [13].

Module 2: Video Preprocessing-Uses Python and OpenCV to extract video frames, resize images, and prepare data for AI analysis.

Express.js, assigns priority levels, and generates Module 3: AI-Based Event Detection Analyses video frames using an AI/ML model to identify predefined unsafe activities, such as fighting or falls [1].

Module 4: Alert Generation and Prioritization Processes detection results through Node.js and alerts for potential incidents.

Module 5: Incident Management and Database-Stores incident details, timestamps, event categories, priority levels, and resolution status in MongoDB for future reference.

Module 6: Real-Time Monitoring Dashboard-Uses React.js to display monitoring status, alerts, incident history, and reports through a centralized interface [14].

Module 7: Citizen Participation and Evidence Management-Allows passengers to report safety concerns and submit supporting evidence for review by authorized personnel.

Module 8: Testing and Evaluation-Evaluates the system using test videos and simulated incidents to measure detection accuracy, false alarms, processing time, and reliability [21].

IV. RESULT AND DISCUSSIONS

1.RESULT

The project "Quiet Watch: Intelligent Passenger Safety Monitoring and Emergency Detection for Public Transportation" aims to provide a software-based solution for monitoring passenger safety and managing potential incidents [15]. The system combines video processing, AI-based event detection, backend services, database management, and a web-based monitoring dashboard. The expected results of the proposed system include organized incident information, alert generation for supported safety-related events, and a centralized interface for authorized personnel to review and manage incidents [2]. The overall results depend on the successful integration and testing of the individual modules.

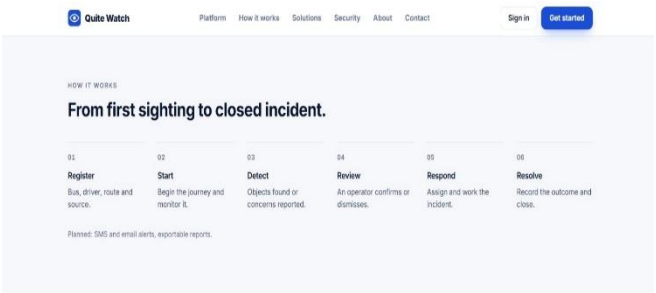


Fig. 4.1 Landing Page

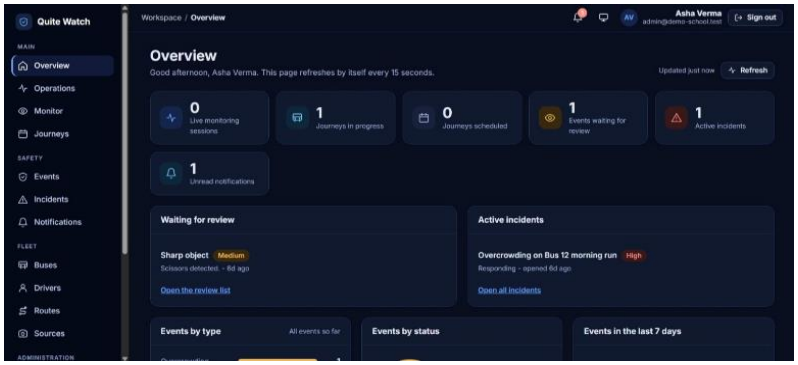


Fig. 4.2 Overview of the Dashboard

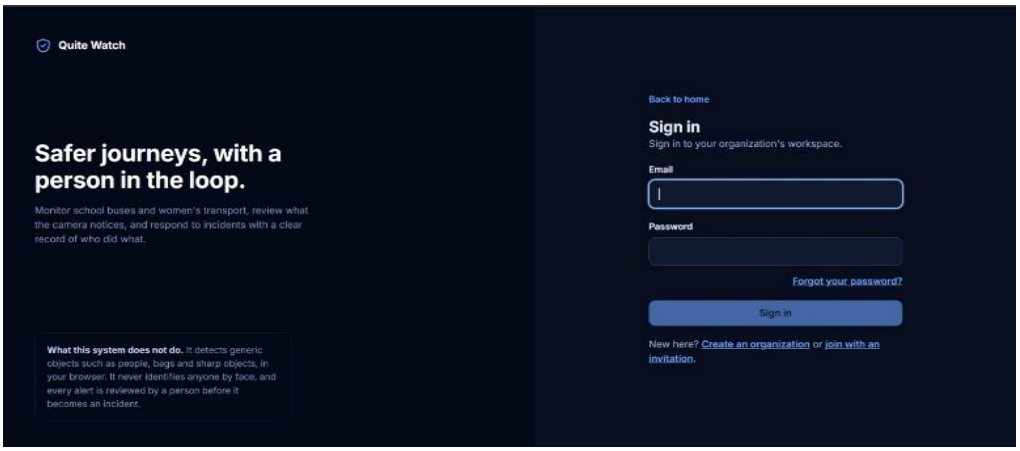


Fig. 4.3 Sign-In Window



Fig. 4.4 Progress Overview

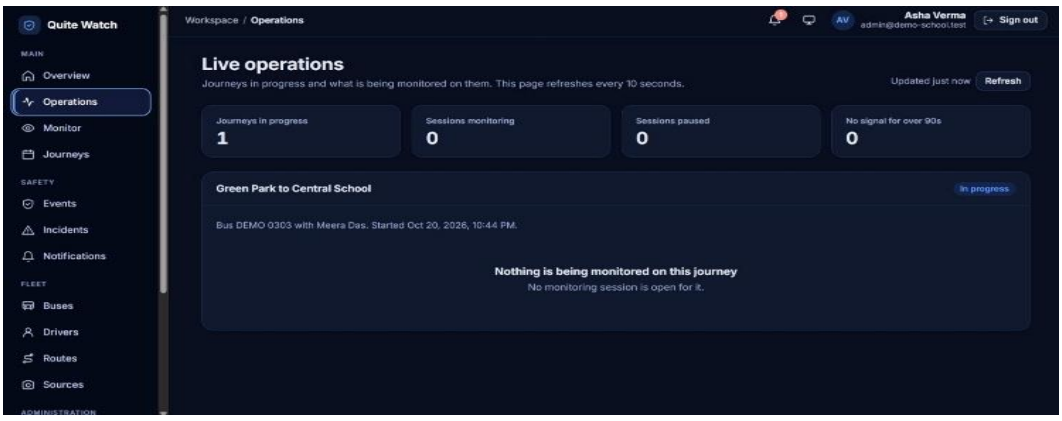


Fig. 4.5 Live Operations

2. DISCUSSION

The proposed Quiet Watch system demonstrates how AI, computer vision, and web technologies can be combined to support passenger safety monitoring [3]. The system is designed to organize video analysis, potential incident detection, alert management, and incident recordkeeping into a single workflow. The video preprocessing module prepares input data for analysis, while the AI detection module is intended to identify predefined safety-related events. The backend and database provide a structured method for managing incident information, and the dashboard offers an accessible interface for authorized personnel [18].

The system's effectiveness depends on several factors:

- Video Quality: Poor lighting, camera angles, obstruction, and low-resolution footage may affect event detection.
- AI Model Performance: The quality of predictions depends on the model, training data, supported event categories, and detection thresholds.
- Alert Reliability: Incorrect detections may produce false alerts, while some genuine incidents may be missed [6].
- Processing Speed: Hardware resources and video-processing methods influence the time required to analyses footage.
- Database and Dashboard Integration: Reliable communication between the backend, database, and frontend is important for displaying accurate incident information.
- Human Verification: Authorized personnel must review potential incidents before treating them as confirmed safety events or taking consequential action [16].

V. FUTURE SCOPE

- Train and validate a transport-specific model using legally obtained, consented or otherwise appropriately authorized data [4].
- Add multicamera tracking or edge processing where it improves latency and is justified by privacy and resource constraints.
- Integrate vehicle location, route and operational context to help authorised teams identify the relevant service [17].
- Add multilingual, accessible passenger reporting and optional anonymous report submission.
- Introduce configurable escalation policies, acknowledgement timers and integration with authorised control-room systems.
- Provide privacy-preserving analytics, retention automation, model-version tracking and periodic bias/performance audits.
- Conduct a limited supervised pilot with a transport operator, independent evaluation and a documented safety case before broader deployment [19].

VI. CONCLUSION

Quiet Watch proposes an AI-assisted passenger-safety monitoring workflow that connects video processing, candidate-event detection, prioritised alerts, incident records and human review [7]. Its intended benefit is to improve situational awareness and make incident handling more structured, not to replace operators or emergency services. The proposed React.js, Node.js/Express.js, MongoDB and Python/OpenCV architecture provides a practical modular foundation for a college prototype [20]. However, responsible use depends on measured performance, secure evidence handling, transparent policies, human oversight and compliance with applicable rules [5]. The next phase should focus on integration, representative testing, latency and false-alarm measurement, and a controlled deployment evaluation. The guiding principle is to detect earlier, understand better, travel safer.

REFERENCES

[1] OpenCV, "OpenCV Documentation," official documentation, https://docs.opencv.org/ (accessed 23 Oct. 2026).
[2] TensorFlow, "TensorFlow Tutorials," official documentation, https://www.tensorflow.org/tutorials (accessed 23 Oct. 2026).
[3] National Institute of Standards and Technology (NIST), "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, Jan.2023.
[4] European Commission, "Ethics Guidelines for Trustworthy AI," High-Level Expert Group on Artificial Intelligence, 2019.
[5] Node.js, "Node.js Documentation," official documentation, https://nodejs.org/docs/latest/api/.
[6] Express.js, "Express Documentation," official documentation, https://expressjs.com/.
[7] MongoDB, "MongoDB Manual," official documentation, https://www.mongodb.com/docs/manual/.
[8] Ms. Snehal H. Chaflekar Ms. Mukta D. Khirwadkar, "Syntax – Directed Translation of Organic Molecular Name to Intermediate Graphical Language", International Journal of Multidisciplinary Educational Research(IJMER), Volume 3, Issue 3(3), 2014
[9] Ms. Mukta D. Khirwadkar Ms. Snehal Chaflekar, "Two-Dimensional Graphical representation for IUPAC Name using Syntax Directed Translation", 3rd IEEE International Conference on Computation of Power, Energy, Information and Communication ICCPEIC, Adhiparashakthi Engineering College, Tamilnadu, 2014, DOI: 10.1109/ICCPEIC.2014.6910396.
[10] Ms. Mukta D. Khirwadkar Ms. Snehal H. Chaflekar, "Translation of organic compound to 2D Graphical Representation using SDT", International Journal on Recent and innovation Trends in Computing and Communication (IJRITCC), Volume 3, Issue 5, 2015 DOI: 10.17762/ijritcc.v3i5.4393
[11] Nikhil S Mangrulkar, Snehal H Chaflekar, "Real Time Detection Of Network Attacks Using Signature Based Approach", International Journal of Computer Science & Engineering Technology (IJCSET), Volume 6, Issue 6, 2015.
[12] S. Chaflekar and R. Raipure, "Application of Cloud System on Secure Cloud Storage," *International Research Journal of Engineering and Technology (IRJET)*, vol. 2, no. 6, pp. 105–109, Sep. 2015.
[13] Ms Snehal Chaflekar, "Concurrency Issues in Object-Oriented Modeling", International Research Journal of Engineering and Technology (IRJET), Volume 3, Issue 2, February 2016
[14] Ms. Snehal Chaflekar, Ms. Deepika Bhatiya, Ms. S. Wankhede, "Simulation of Routing Protocols in Wireless Ad-hoc Networks", International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), Volume 3, Issue 3, March 2016, https://www.technoarete.org/common_abstract/pdf/IJERCSE/v3/i3/69.pdf
[15] Ms Snehal H Chaflekar, Ms Ashwini Lokhande, Ms Priyanka Gomase, Ms Rupali Shinganjude, "Compiler Architecture and Design Issues", International Journal of Computer Science Trends and Technology (IJCST), Volume 5, Issue 3, 2017
[16] Snehal Chaflekar, Ankit Mandpe, Mangesh Ganvir, Aishwarya Sahare, "Payroll system Using Fingerprint and Face Detection", International Research Journal of Engineering and Technology (IRJET), Volume 5 Issue 3, e-ISSN: 2395, 2018, https://www.irjet.net/archives/V5/i3/IRJET-V5I3273.pdf
[17] Prof. Rupali Shinganjude Prof. Snehal Chaflekar, "Review paper on Application of Data Minig on Healthinformatics", International Journal of

Computer Science and Engineering (IJCSE), Volume 6 Issue 8, 2018, https://ijcrt.org/papers/IJCRT_190623.pdf

[18] Dhanshree Moundeka Prof.S.H.Chaflekar, Suyesha Tokekar, Gaurav Lambat, Pranjali Nasre, Vishakha Hude, Akanksha Korde, "An Android Application for Animal Welfare", International journal of Scientific Research in Science, Engineering and Technology(IJSRSET), Volume 6, Issue 8, 2019, DOI: 10.32628/IJSRSET196249

[19] Snehal Chaflekar, Shivani Pawade, Swati Thakur, Ankush Vaidya, Anshul Gajapure, Mahima Chaudhari, "Chat-Bot for Blind Student", International Journal of Advanced Research in Science, Communication and Technology, Volume 3 Issue 8, 2023,

[20] S. Chaflekar, R. Rewatkar, K. T. V. Reddy, U. Pacharaney, M. Adak, S. Khedkar; "Job scheduling approach in load balancing in cloud computing environment". *AIP Conf. Proc*. 10 December 2024; 3188 (1): 040016. https://doi.org/10.1063/5.0240268

[21] S. Chaflekar and R. Rewatkar, "Model Agnostic Predictive Smart Load Balancer in Microservices Environment,"*2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI)*, Wardha, India, 2024, pp. 1-6, doi: 10.1109/IDICAIEI61867.2024.10842849.

[22] S. Chaflekar, R. Rewatkar, K. T. V. Reddy; "Load balancing in cloud environment": Issues and challenges. *AIP Conf. Proc.* 14 February 2025; 3162 (1): 020053. https://doi.org/10.1063/5.0241726

[23] Snehal Chaflekar and Rajendra Rewatkar, "Novel Load Prediction in Microservice Architecture using Attention Mechanism based Deep LSTM Networks (AMDLN)", *International Journal of Innovative Research and Scientific Studies (IJIRSS),* Vol. 8 No. 3 (2025), https://doi.org/10.53894/ijirss.v8i3.6751.

[24] S. Chaflekar, R Rewatkar, M Chaudhari, S Patil, R Shinganjude, T Panse, "Real-Time Cloud Deployment of Flask-AMDLNR: An AI-Driven Application-Layer Load Balancer for Microservices", 3rd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIHEI), Wardha, India, 2025, DOI:10.1109/IDICAIHEI65991.2025.11378789