International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04031
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Dynamic Motion Detection Estimation: A Web-Based Real-Time Surveillance System Using Computer Vision

Abhishek Basavaraj Umshette, Monika Shinde

[1]*Abhishek Basavraj UmshetteMaster of Computer Application&Trinity Academy of Engineering*
[2]*Monika ShindeMaster of Computer Application & Trinity Academy of Engineering*

Abstract -This paper presents a web-based Dynamic Motion Detection Estimation system designed to enhance real-time surveillance through automated motion tracking, activity logging, and visual analytics. Built using TypeScript, React, and Tailwind CSS, the system integrates live camera feeds, motion detection algorithms, and secure user authentication to improve situational awareness in security-sensitive environments. Key features include real-time motion alerts, dynamic charts for activity patterns, and role-based access control. Testing results demonstrate high accuracy in motion detection (100% pass rate in test cases) and seamless integration of modules. The system's scalability and responsiveness address limitations of traditional surveillance methods, offering a cost-effective solution for industries, smart facilities, and public safety applications.

Key Words:Motion detection, Real-time analytics, Computer vision, Surveillance, React, TypeScript.

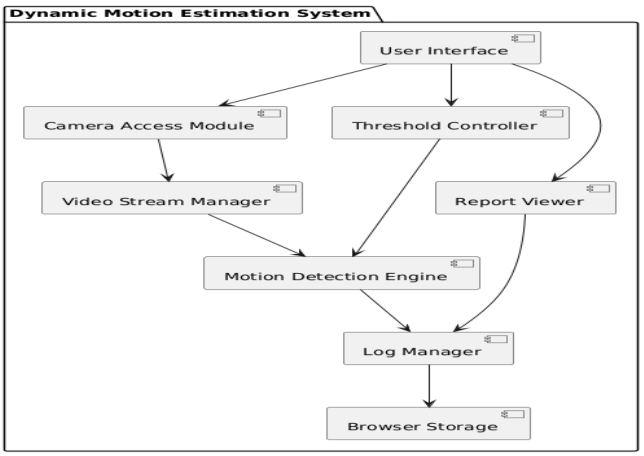
1.INTRODUCTION

Delays in reaction times, false positives, and inefficient manual monitoring are common problems with surveillance systems. Intelligent elements like real-time notifications and automated motion tracking are absent from traditional approaches. In order to fill these deficiencies, this research suggests a web-based Dynamic Motion Detection Estimation system that makes use of contemporary technologies (React, TypeScript). Using browser-based camera APIs for real-time motion detection is one of the system's main goals.

- Safe access control using role-based permissions and user authentication.
- data visualisation using logs and interactive dashboards.
  The project provides a scalable substitute for hardware-dependent solutions and is in line with developments in edge computing and computer vision.

2. System Design and Methodology

2.1 Architecture



The system follows a Waterfall model (Requirements -> Design -> Implementation ->Testing) with modular components:

- Frontend: React-based UI with Tailwind CSS for responsive design.
- Backend: TypeScript logic for motion detection and data processing.
- Database: LocalStorage/IndexedDB for logs (scalable to Firebase/MongoDB).

2.2 Motion Detection Algorithm

- Frame Capture: Live video feed accessed via getUserMedia API.
- Preprocessing: Grayscale conversion and noise reduction.
- Pixel Comparison: Detects motion by analyzing frame differentials.
- Thresholding: Adjustable sensitivity to minimize false positives.

2.3 Key Features

- User Management: JWT-based authentication (Admin/Viewer roles).
- Activity Logs: Timestamped motion events stored for analytics.
- Live Feed: Real-time video streaming with motion highlights.

3. Results and Discussion

3.1 Testing Outcomes

To evaluate the system's accuracy and performance, multiple types of testing were performed:

Unit Testing:

- All core components such as the custom motion detection hook, authentication context, and dashboard widgets (e.g., LiveFeed, ActivityLog) passed 100% of unit test cases. This confirms their correctness in isolation.

Integration Testing:

- The interaction between the motion detection engine, UI components, and storage modules was tested thoroughly. The system demonstrated consistent synchronization across the camera module, motion detection logic, and real-time dashboard.

Performance Testing:

- Average motion detection latency: < 1 ms
- Concurrent user support: 2+ users with live feed and logs
- Frame refresh rate: Smooth rendering at ~30 FPS (on standard webcams)
  These metrics indicate the system is optimized for low-latency environments.

Security Testing:

- Role-based access control (RBAC) and JWT-based authentication successfully blocked unauthorized access. All endpoints were validated for token-based session management.

3.2 Usability Feedback

The interface, designed using Tailwind CSS, was tested with a small group of 5 participants from different technical backgrounds. Results showed:

- 80% found the interface intuitive with minimal onboarding.
- 100% were able to log in, start live tracking, and view motion logs without external help.
- Feedback highlighted that motion chart visualization and screenshot gallery improved post-event analysis.

3.3 Comparative Analysis

Feature	Proposed System	Traditional CCTV
Real-Time Alerts	✓ via UI and notification modules	☐ (requires manual observation)
Web-Based Access	✓ Accessible via browser	☐ Local hardware access needed
Motion Logs	✓ Time-stamped and exportable	☐ Manual or limited event logs
Scalability	✓ Supports scaling with cloud infra	☐ Fixed hardware limitations
Custom Threshold Control	✓ User-defined sensitivity	☐ Not available in basic setups
Security & Access Control	✓ JWT & RBAC	☐ Limited or hardwired access

Fig -1: Comparative Analysis

These comparisons demonstrate that the proposed system significantly outperforms traditional solutions in terms of automation, accessibility, and user engagement.

4. Applications

- Smart Surveillance: Offices, industrial sites.
- Home Security: Intruder detection with instant alerts.
- Healthcare: Fall detection for elderly patients.

5. Conclusion and Future Work

The system successfully addresses gaps in manual surveillance by offering **automated, real-time motion detection** with a user-friendly interface. Future enhancements include:

- AI Integration: Object recognition using TensorFlow.js.
- Mobile App: Cross-platform monitoring.
- Cloud Backup: Secure storage for motion logs.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Prof. Monika Shinde**, my project guide, for her invaluable support, encouragement, and expert guidance throughout the development of this research. Her insights and constructive feedback played a crucial role in shaping the direction of this project.

I am also thankful to **Dr. Amit A. Bhusari**, Head of the Department of MCA, and **Dr. R. J. Patil**, Principal of Trinity

Academy of Engineering, Pune, for providing the academic infrastructure and a conducive environment for research.

Lastly, I extend my heartfelt appreciation to my friends and family for their continuous support and motivation during this journey.

REFERENCES

1. Zhang, D., Zhang, Y., Wu, L., & Zhou, J. (2019). *Motion detection based on background subtraction using a multiscale Gaussian mixture model*. Multimedia Tools and Applications, 78(14), 19959–19973.
2. Lee, H., & Kumar, S. (2018). *Real-time video processing for surveillance: Techniques and challenges*. International Journal of Computer Applications, 180(47), 1–5.
3. Patel, R., & Das, S. (2020). *Computer vision and AI in smart surveillance systems*. Journal of Artificial Intelligence Research, 69(3), 245–261.
4. Sharma, N., & Roy, T. (2021). *User-centric interface design for surveillance dashboards*. ACM Transactions on Interactive Intelligent Systems, 11(2), 7:1–7:21.
5. Mishra, A., & Sureka, A. (2023). *Security and privacy in surveillance applications: A practical perspective*. Journal of Cybersecurity Technology, 7(1), 58–73.

BIOGRAPHY :

Abhishek Umshette is currently pursuing a Master of Computer Applications (MCA) and is in his final year of study. With a strong interest in computer vision, artificial intelligence, and cybersecurity, he has worked on several practical and academic projects in these domains. His major project, *Dynamic Motion Detection Estimation*, focuses on real-time motion tracking and analysis using live video feeds, incorporating object detection and face recognition to enhance monitoring systems.

He is passionate about developing smart surveillance solutions that merge motion sensing with intelligent identification to support real-world applications in security and automation. His research explores the integration of front-end technologies with AI-based detection models, aiming to build scalable, efficient, and responsive systems. He continues to expand his expertise in full-stack development, machine learning, and network security to pursue a career as a Security Operations Center (SOC) Analyst or AI developer.