Sentinel Vision: Smart AI Surveillance System

Enhancing Surveillance Efficiency with Intelligent AI Agents for Automated, Real-Time Person, and Vehicle Monitoring

Dineshkumar L, Hemanth N, Navaneeth S, Sridhar G, Dhinaharan S

Department of Artificial Intelligence and Data Science,
Sri Shakthi Institute of Engineering and Technology, Coimbatore
ldineshkumar38@gmail.com, hemanthviky@gmail.com , navaneeth.wb@gmail.com, isrisan54@gmail.com, dhinabru@gmail.com

Abstract— **This project creates an intelligent surveillance system that processes video feeds to detect and analyze mask usage in real-time. Through the combination of computer vision and deep learning models, the system identifies individuals, determines mask compliance, and performs contextual analysis within monitored environments. The system employs a multi-layered architecture: video processing modules analyze frames, YOLO-based detection models identify persons and mask status, and a web interface visualizes results. Among the key innovations are real-time compliance monitoring, demographic estimation capabilities, and comprehensive analytics dashboards. Implementation outcomes demonstrate improved monitoring efficiency, reduction of manual observation needs, and enhanced safety protocol enforcement compared to traditional surveillance methods.**

Index Terms—**AI Agents, Computer Vision, YOLO, Real-Time Surveillance, OpenCV, Face Detection, Safety Monitoring, Mask Detection.**

I. Introduction

In the modern era of public safety and security, institutions across various sectors—education, healthcare, transportation, and public spaces—require robust surveillance systems to ensure compliance with safety protocols and maintain security. Traditional surveillance methods often rely on manual observation, leading to inefficiencies, human error, and potential security lapses. With the advancement of artificial intelligence (AI) and computer vision technologies, automated surveillance has become a viable solution; however, existing systems often lack real-time processing capabilities, contextual awareness, and adaptability to specific safety requirements.

The Sentinel Analyse project introduces an intelligent surveillance system that automatically processes video feeds to monitor mask compliance and safety protocols in real-time. By integrating computer vision, deep learning, and real-time analytics, the system provides a comprehensive solution for safety monitoring. The system architecture employs Flask for the backend, React for the frontend, and YOLO for object detection, creating a scalable, efficient, and high-precision surveillance solution

II. Motivation and Problem Statement

Despite the presence of AI-based surveillance solutions, current systems have several limitations:

- ***Lack of Real time processing:*** Most surveillance systems rely on post-processing of video data, leading to delayed detection of safety issues and compromised response times.
- ***Manual Monitoring Dependency:*** Traditional surveillance requires continuous human observation, which is not only labor-intensive but also prone to human error and fatigue
- ***High Implementation Costs:*** Existing AI-powered surveillance solutions often require expensive hardware and specialized infrastructure, making them inaccessible to small institutions and organizations.
- ***Limited Adaptability:*** Current systems struggle to adapt to varying environmental conditions and lighting, leading to inconsistent detection rates and false positives.
- **Security Protocol Enforcement Gaps:** Many surveillance solutions lack comprehensive features for monitoring and enforcing safety protocols, such as mask compliance, in real-time.
- **Complex Integration:** Integrating AI-based surveillance systems with existing security infrastructure often requires significant technical expertise and customization, creating barriers for adoption.

To solve these issues, the system we introduce offers an AI-based platform for Real-time analyzing that offers:

- ✓ Real-time Video Processing: Leveraging YOLO and computer vision for instant detection and analysis of safety protocols.
- ✓ Multi-layered Detection: Combining YOLO and Haar Cascade for accurate person and mask detection across varying conditions.
- ✓ Scalable Architecture: Using Flask and React for efficient handling of multiple video streams and real-time data visualization.
- ✓ User-friendly Interface: Providing an intuitive dashboard for easy monitoring and management of surveillance data.
- ✓ Adaptive Processing: Implementing intelligent algorithms to adjust to different lighting and environmental conditions.

- ✓ Comprehensive Analytics: Offering detailed insights and reports for data-driven decision making in safety management.
- ✓ Human-Centric Design: Creating a system that complements human security personnel, reducing their workload while enhancing their effectiveness in maintaining safety protocols.

III. MAJOR CONTRIBUTIONS

- ✓ ***This paper makes several significant contributions to the field of AI-powered surveillance systems:***

A New Multi-layered Detection Framework for Real-time Surveillance: Our system combines YOLO and Haar Cascade detectors, operating under real-time constraints to provide accurate person and mask detection across varying environmental conditions

Hybrid Computer Vision Processing System: Integrates deep learning (YOLO) with traditional computer vision (Haar Cascade) for high-precision detection with robust performance in diverse lighting and environmental conditions.

Scalable Real-time Processing Architecture: Utilizes Flask for efficient video stream processing and React for real-time data visualization, enabling the system to handle multiple video feeds simultaneously without performance degradation.

User-friendly Surveillance Management Interface: Provides an intuitive dashboard that allows security personnel to monitor, analyze, and manage surveillance data in real-time, enhancing their effectiveness in maintaining safety protocols.

Adaptive Processing Algorithms: Implements intelligent algorithms that automatically adjust to different environmental conditions, reducing false positives and improving overall detection accuracy**.**

Performance Comparison with Traditional Methods: Our implementation results demonstrate significant improvements in detection accuracy, reduced false positive rates, and enhanced real-time monitoring capabilities compared to conventional surveillance systems.

IV. SYSTEM ARCHITECTURE AND TECHNICAL APPROACH

- ***The proposed AI-powered surveillance system architecture is multi-layered, designed for intelligent video analysis and real-time monitoring:***

1. Ingestion and Video Preprocessing

Video Ingestion and Preprocessing Video feeds are captured from various sources and pre-processed using frame extraction, normalization, and quality enhancement techniques to ensure optimal input for detection algorithms. This layer handles video format conversion, resolution adjustment, and frame rate optimization to maintain consistent processing across different input sources.

.

2. Multi-layered Detection Processing YOLO-based Object Detection

Identifies persons and potential mask wearers in video frames. Haar Cascade Detection: Provides complementary face detection capabilities, especially in challenging lighting conditions. Demographic Analysis: Estimates age and other demographic characteristics using computer vision techniques.

3. Real-time Monitoring and Analysis Flask Backend

Processes video streams in real-time, handling multiple concurrent connections. React Frontend: Provides real-time visualization of detection results and analytics. Data Processing Pipelines: Analyzes detection data to generate actionable insights and compliance reports.

4. Scalability and Deployment Flask Server

Handles dynamic scaling of video processing tasks. React-based Dashboard: Offers intuitive monitoring and management capabilities. Local Storage: Manages temporary video processing data and analytics results.

V. ENTIRE APPLICATION WORKFLOW BETWEEN EACH MODULES

Key Contributions of the Proposed System for Real-time Surveillance Processing

- ✓ The proposed system employs a multi-layered detection framework that focuses on real-time video processing to ensure high accuracy and reliability in safety monitoring across various environments.
- ✓ It achieves this through the integration of YOLO and Haar Cascade detectors, multi-layered query processing, and optimized video stream handling, maintaining low-latency response times
- ✓ The system logs all user activity and detection results across the frontend, API gateway, and backend layers to facilitate

compliance, debug issues, and track system performance.

- ✓ The system logs all user activity and detection results across the frontend, API gateway, and backend layers to facilitate compliance, debug issues, and track system performance.
- ✓ A feedback mechanism allows continuous improvement of detection accuracy based on user evaluations and real-world performance data, enabling adaptivity to different environments.
- ✓ The system implements intelligent caching of processed video frames, optimizing redundant processing and reducing computational overhead while maintaining real-time performance
- ✓ The proposed system utilizes Flask for serverless deployment, enabling flexible scaling, security, and modular integration, maintaining high reliability for enterprise-level surveillance applications.

V. MODEL CONTEXT PROTOCOL (MCP) FOR AI-DRIVEN SURVEILLANCE OPTIMIZATION

Layer	**Description**
Real-time Context Awareness & Preprocessing	Identifying video stream characteristics (lighting, resolution, environment) and applying domain-specific preprocessing using adaptive algorithms
Optimized Detection with Multi-layer Prediction	Implementing a multi-model prediction system that combines YOLO and Haar Cascade for improved detection accuracy while minimizing computational overhead
Decision Validation and Safety Assurance	Ensuring detection results comply with safety protocols and validating the accuracy of AI-generated insights through cross-verification
Efficient Resource Management	Implementing intelligent caching and parallel processing using Flask’s multi-threading capabilities to optimize real-time video processing performance

The Model Context Protocol (MCP) establishes workflows for AI-powered surveillance systems, aiming to create a high-quality, efficient, and accurate security monitoring solution. The layers are as follows:

- ➢ Real-time Context Awareness & Preprocessing – identifying video stream characteristics (lighting, resolution, environment) and applying domain-specific preprocessing using adaptive algorithms.
- ➢ Optimized Detection with Multi-layer Prediction - implementing a multi-model prediction system that combines YOLO and Haar Cascade for improved detection accuracy while minimizing computational overhead.
- ➢ Decision Validation and Safety Assurance - ensuring detection results comply with safety protocols and validating the accuracy of AI-generated insights through cross-verification.
- ➢ Efficient Resource Management - implementing intelligent caching and parallel processing using Flask's multi-threading capabilities to optimize real-time video processing performance.

VII. Comparison with Existing Approaches

Feature	Traditional Surveillance	Generic AI-Based Surveillance	Proposed Sentinel Analyze System
Real-time Processing	No (post-processing)	Limited (Delayed)	High (Immediate Detection)
Detection Accuracy	| Low (Human Dependent)	Moderate	High (Multi-layered Detection)
Scalability	Fixed Camera Setup	Limited by Hardware	| Flexible (Multi-stream Processing)
Ease of Use	Complex	Requires Technical Expertise	User-friendly Dashboard
Analytics Capabilities	Manual Review Only	Basic Statistics	Comprehensive Insights
Cost Efficiency	Expensive (24/7 Monitoring)	Moderate	High (Optimized Processing)

VIII. Impact and Future Applications

➢ This research presents a highly effective, scalable, and cost-efficient AI-powered surveillance solution that addresses critical safety monitoring needs in educational institutions, healthcare facilities, transportation hubs, and public spaces. This system enhances security by:

- ***Reducing human labour through automated real-time monitoring and compliance verification.***
- ***Minimizing human error through AI-driven detection and verification procedures.***
- ***Strengthening safety protocol adherence through continuous real-time monitoring.***
- ***Facilitating intuitive management through a user-friendly dashboard interface.***

In future research, this system can be expanded to include multi-modal AI platforms that integrate video, audio, and environmental sensors, thereby providing more comprehensive surveillance capabilities. Additional improvements in real-time processing and user customization will likely promote wider adoption of AI-based surveillance technologies across various security applications.

XI. Conclusion

❖ This paper introduces a novel AI-powered surveillance framework designed for real-time safety monitoring, bridging the gap between traditional surveillance systems and modern security needs. By integrating multi-layered detection, real-time processing, scalable architecture, and user-friendly interfaces, the system provides a transformative approach to intelligent surveillance. The proposed architecture demonstrates superior efficiency, accuracy, and compliance adherence compared to traditional manual monitoring methods, offering a compelling solution for modern organizations aiming to optimize safety protocols and security management.

X. ACKNOWLEDGMENT

❖ As a team, we would like to formally thank Sri Shakthi Institute of Engineering and Technology for providing the materials and research assistance for this work. We thank the mentors and faculty for their support, advice, and constructive feedback as this project developed. We would also like to thank our fellow collaborators for their consistent support, discussions, and technical contributions that positively influenced the project. We would similarly like to thank the open-source communities for YOLO, Flask, and React for their access to powerful tools and frameworks that enabled us to develop this system efficiently. Finally, we would like to extend our appreciation to our peers, friends, and family members for their encouragement and motivation to complete this working prototype in its entirety.

REFERENCE

- ✓ YOLOv8: An Improved Real-time Object Detection - https://arxiv.org/pdf/2207.02696
- ✓ OpenCV Haar Cascade for Face Detection - https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html
- ✓ Real-time Video Processing with Flask - https://flask.palletsprojects.com/en/3.0.x/patterns/streaming/
- ✓ React for Real-time Data Visualization - https://react.dev/learn
- ✓ Deep Learning for Computer Vision - https://www.deeplearningbook.org/
- ✓ Computer Vision: Algorithms and Applications - https://szeliski.org/Book/
- ✓ Real-time Video Analysis Techniques - https://www.sciencedirect.com/topics/computer-science/real-time-video-analysis
- ✓ Flask for Multi-threaded Processing - https://flask.palletsprojects.com/en/3.0.x/deploying/wsgi-standalone/
- ✓ Computer Vision with Python - https://www.pyimagesearch.com/
- ✓ Real-time Surveillance Systems Design - https://www.sciencedirect.com/topics/computer-science/surveillanc-system