International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

An Intelligent CCTV System with Context-Aware Alerts for User-Defined Threats

Dr. Girish J. Navale[1], Varad Chidrawar[2], Samarth Dangat[3], Saharsh Dudhyal[4], Prajyot Fulari[5]

[1]Assistant Professor, Computer Science Engineering AISSMS IOIT, Pune, India
[2,3,4,5]Computer Science Engineering AISSMS IOIT, Pune, India

Abstract
Traditional security systems are the types that simply capture videos and they are incapable of raising an alarm whenever something bad is occurring. In contrast, a smart, correct alert, such as the sense that someone is intruding to the premises or that somebody is stealing something, might be helpful in preventing it before it grows. Majority of traditional intelligent CCTV configurations are also based on alerts that are configured by programmers and are hard coded which are highly likely to give a lot of false positives and are not readily manipulated to meet the requirements of the customers. The rule-based context- conscious alert engine that I have to offer is a priority system and simple anomaly detection. This allows users to configure the way the CCTV alerts act without necessarily the user touching the code, but by simply using the interface.

Keywords: Smart Surveillance CCTV, User-Pre- Defined Alerts, Context-Aware Surveillance, IoT Security, Rule- Based Alerting.

INTRODUCTION

A. Evolution and Current State

Surveillance systems are primarily applied to the field of home security especially CCTV systems. It also helps users to store and capture video footage in order to observe the environment around them. It also helps in crime prevention through constant monitoring and reviewing. But though this might be sounding like an advantage, it can also be taken to be a disadvantage. Videos are being stored, recorded and shared through the infrastructure due to the fact that the videos are recorded. Nevertheless, it still requires the human operators to identify threats. The relationship between the user-surveillance system is most likely to be context- independent.

Due to the recent events, the goal of the security has changed to stop the potential crimes and not to solve the crimes. Basic CCTV has been replaced by intelligent technologies of the network surveillance. These systems can analyze video material and identify suspicious activity without the help of a human operator. These latest smart CCTV systems have exploited the strength of the IoT technology. Completely automated and notification solutions are provided using mobile interface and computer vision.

As compared to the traditional video surveillance systems, these technologies have a major advantage, including:

- Forensic Capabilities: Video retrieval based on content

- Situational awareness: Aware of the location and activity of things within the area under observation.

B. Key Challenges

Nevertheless, this industry still has some possible spheres where it can be challenged severely. Due to the fact that much is yet to be known about this surveillance technology. Nonetheless, several major are present. However, there are some significant shortcomings or weaknesses that could offset its benefits. Here are a few of these difficulties:

1. The Multiscale Challenge

Due to the fact that this system must be aware of the way it gathers information on different scales. The field that needs to be done with accuracy is in the improved information acquisition based on high resolution face photos and real time automatic video analysis.

This brings in an entirely different area of study and some of the issues in the performance evaluation are camera control, camera management based on the task, the allocation of resources and the processing of footage using the moving cameras.

2. The Alert Fatigue Problem

These systems give an alert even when the priority of the events is low like the movement of pets, change of lights, roaming bikes, swaying branches in the wind or even the known moves of family members. Due to these events, users are receiving dozens of notifications per day, and this is 80 times more than the false positive rate.

The tension of the false alarms is so constant that these signals are likely to lose some of their meaning. It can then progress into the neglecting or ignoring these warnings even when they are real. Existing systems are generally limited in the number of predefined threat settings, but they generally allow the user to customize them according to the developers.

3. Context Insensitivity and Rigidity

The primary reason behind the problem of Alert Fatigue is the context-insensitive design, which is a variation on the previous problem. The reason why these systems tend to fail is due to overreliance on fixed and developer-based logic. Disregarding the variations of patterns of usage, place, and time.

Significant aspects which are ignored by traditional models:

- Time of the Day: day, time, or some special schedules.
- Spatial Context: Sensitive and Non Sensitive areas.
- Situational Context: trends of anticipated and unanticipated activity. Smart surveillance solutions eliminate the need of the users to monitor the video footage manually or even to choose these frequent flash notification.

4. The Contextual Event Detection Challenge

Nevertheless, it is important to be able to track and identify objects in the video surveillance system. Running an automated analysis of data to discover trends and interesting events is one of the largest problems.

Explanations on the events depending on the context are:

- Understanding the time and environment to enhance video analysis.
- Interpreting events using activity models and geometric models of the surroundings.
- System performance and anomaly detection.

SYSTEM ARCHITECTURE AND METHODOLOGY

A. Architectural Overview and Design Philosophy: The design of the proposed smart CCTV is modu-

lar and stratified. The design puts the highest priority on flexibility, privacy, computational efficiency and real-time responsiveness. Unlike the traditional surveillance, intelligence is not cloud-based. Decision making and processing are done close to the sources of data. This approach is in line with the edge-first architecture.

Video streaming is minimized through continuous video streaming by doing computation at the edge. This consequently reduces the network bandwidth consumed. System latency is minimized by minimizing cloud round-trip delays. Local infrastructure keeps on storing the sensitive video information. Cores monitoring functionalities remain functional even without the use of the internet. This enhances reliability of systems in the occasion of network breakdown.

The modular design provides different opportunities in terms of deployment. These comprise small corporations and homes. One can update or replace specific components individually. Maintaining the system becomes less difficult and disruptive. This is through loose coupling which is fostered through clear module interfaces.

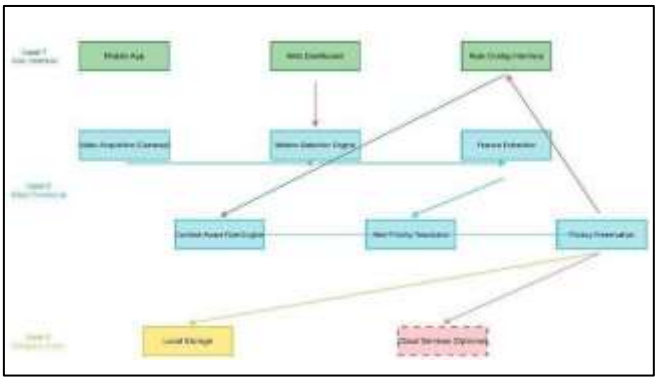

Fig 1 System Architecture

B. Video Acquisition Module

The video acquisition module is the input part of the system. It comprises IP or CCTV cameras and is linked to the network. The cameras are strategically located in order to cover an area. Video streams are recorded in their course of operation. The user is still capable of changing the resolution and frame rate. The common resolutions are 720p to 1080p. Frame rates vary between 15-30. Common video codecs such as H.264 and H.265 are supported. These formats ensure that there is great storage and compression. Each camera of a multi-camera system is assigned its own personality. Cameras are associated with pre-determined surveillance areas. This details their responsibility of surveillance.

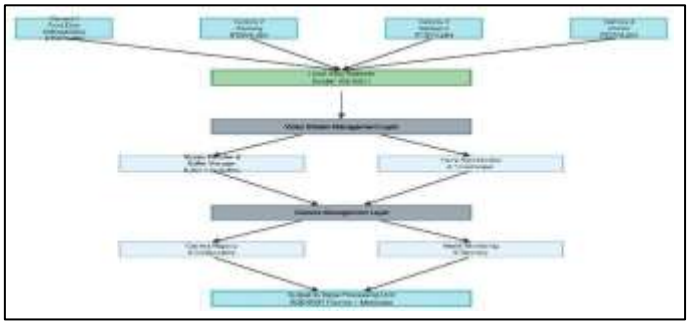

Fig. 2. Video Acquisition Module Flowchart

The cameras are connected locally with the edge processing unit. Communication is done on standard IP-based protocols. There is the use of streaming protocols such as RTSP. Internet connectivity is not needed in local functionality. Cameras do not send video to the cloud services. Streams are all initially directed to edge analysis. This design has full data control to users. Cloud-first architectures reduce risks of privacy. The module works with PTZ, outdoor, and indoor cameras.

C. Edge Processing Unit

The edge processing unit is the computing centre of the system. It makes decisions and analyses videos in real time. The unit is implemented using platforms of embedded computing. Two of them are NVIDIA Jetson Nano and Raspberry Pi. These devices balance power consumption, price and performance. They do not mind working at home.

All of the connected cameras feed the device with a video feed. Light weight computer vision techniques are employed locally. In motion detection background subtraction techniques are employed. Basic object segmentation and feature extraction are done. These processes are not much delayed. Still the reaction times remain in hundreds of milliseconds.

The algorithm optimization ensures real-time performance. In the case of hardware acceleration, this is employed. Neural accelerators or GPUs enhance processing efficiency. The unit maintains background models and event history. Rule configurations are stored in place.

D. Context-Aware Rule Engine

The context-aware rule engine is a representation of system intelligence. It supports user defined and flexible monitoring actions. The system maintains a set of user-specified rules. These rules contain policies of surveillance. The incoming events are always investigated.

Each rule is characterized by a number of condition dimensions. These include category of the objects and intensity of motion. The activation of the rules is limited by time. Spatial limitations are used to define valid event locations. Semantic conditions are used to represent environmental or situational context.

The rule engine evaluates every frame that is analyzed in the video. Founded observations are matched against active rules. In case, multiple rules are met simultaneously conflicts might arise. Priority resolution logic is applied in such scenarios. In doing so, there are no superfluous or extraneous alerts. The engine ensures that warnings reflect the intent of the user in a proper way. E. Alert Management and Notification Module

Priority	Trigger Examples	Notification Channels	Delivery Behavior
High	Motion in restricted zones at night, intrusion attempts	Push notifications, SMS, audible alarms, automated responses (lights/video)	Immediate delivery
Medium	Noteworthy events like package delivery, loitering	Push notifications	Slight delay or batched
Low	Informational events like routine motion	Log to database	No real-time notifications; review later

The alert management module deals with notification delivery. Groups of alerts are grouped by priority ratings. Normal levels are low, medium and high. Information is sent based on priority and urgency.

Feature	Description
Notification History	Stores past alerts for review, acknowledgment, and feedback to refine rules
Integration	Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNS) for efficient metadata delivery
Customization	User-defined quiet hours, preferred channels per priority level
Bandwidth Efficiency	Sends compact metadata only, avoiding full video streams

The system ensures the notification history is maintained. Users can always go through past warnings. Alerts can be admitted by the user. Feedback can be used to make rules better. The user is still able to customize his or her notification options. The alerts not considered critical are disabled during off hours. Customers can select the channels that they would prefer being notified through.

F. Privacy Preservation Layer

The protection of data is implemented through the privacy preservation layer. It regulates the transmission, storage and access of data. The design is based on a privacy-by-design approach. Gathering of data is minimized by default. Users have to enable cloud functionalities explicitly.

Low priority occurrences can be anonymized. Two methods are motion patterns and facial blurring. There is elimination of identifiable information before storage. The user defines video retention guidelines. The retention times are based on the priority. More important clips could be retained. Priority video is quickly eliminated.

Privacy dashboard provides system transparency. Users can view data, which has been transmitted and saved. Retention timelines are done in a comprehensible way. Access control restricts the unauthorized viewing. All access attempts are recorded on audit logs. This brings about accountability and trust.

METHODOLOGY + ALGORITHMS

A. User-Defined Rule Model

The flexibility of the proposed system will rely on a systematic rule model. Such a framework can be used to come up with a definition of surveillance policies in a highly expressive and effective way. Every surveillance rule is depicted in the form of a formal tuple:

$$R_k = \{C_k, T_k, S_k, P_k, A_k\}$$

Each component represents another facet of surveillance behavior. The condition element C_k specifies the event characteristics necessary for rule triggering. These include minimal intensity thresholds or motion presence. Boolean predicates are used to express conditions. Additionally, deeper logical expressions are supported. The extracted event features are used to assess these expressions. Rule activation periods are

specified by the temporal component T_k. These intervals could be regular or fixed. Additionally, derived temporal conditions are supported. Weekend or midnight schedules are two examples. Context-sensitive activation is made possible by temporal definitions.

Monitored areas are represented by the spatial component S_k. These areas are in the field of focus of the camera. Geometric shapes are used by users to designate zones. Image coordinates are transferred to defined forms. Rules are given precedence by the priority component P_k. Alert precedence is determined by this value. Conflict resolution is regulated by higher priority rules. System response behavior is specified by the action component A_k. Notifications and video recording are examples of actions. Additionally, automated answers are supported. Usability and computing efficiency are balanced in this organized formulation.

B. Context-Aware Alert Evaluation Algorithm Alert evaluation is done through a continuous hierarchical decision making process. The process of every processed frame leads to an observed event. The event representation at time t can be given as:

$$A_t = (t, S_e, M_e, D_e, O_e)$$

where S_e is the spatial area identified, M_e is the intensity of the movement, D_e is the time interval, and O_e is optional object category. The system obtains relevant rules to each occurrence. These regulations have their origin in the source camera. To begin with, there is the temporal filtering:

$$\mathcal{R}_T = \{R_k \in \mathcal{R}_c \mid t \in T_k\}$$

This step is the removal of temporarily inactive rules. The next step is the spatial filtering:

$$\mathcal{R}_Z = \{R_k \in \mathcal{R}_T \mid Z_e \cap Z_k \neq \emptyset\}$$

Lastly, predicates of condition are verified:

$$\mathcal{R}_C = \{R_k \in \mathcal{R}_Z \mid C_k(E_t) = \text{true}\}$$

This hierarchized filter removes irrelevant calculation. The complexity of the assessment is minimized significantly. C. Motion Detection and Feature Extraction.

C. Motion Detection and Feature Extraction

Motion detecting requires the use of adaptive background models. Gaussian Mixture Models are employed to be robust. The model for each pixel.

$$P(p) = \sum_{i=1}^{K} w_i \cdot \mathcal{N}(\mu_i, \Sigma_i) \quad p(x,y) \text{ is:}$$

Forground pixels are identified with the help of deviation thresholds. The pixels which are conspicuous to the background are selected. Morphological refinement is done on the detected regions. Minor objects and distortion are removed.

Connection component analysis is used to group foreground pixels. A collection of motion blobs represents an individual group. Many features are obtained out of each blob.

Area:

$$A_i = \sum_{(x,y)\in b_i} 1$$

Bounding box:

$$B_i = x_{max} - x_{min}, H_i = y_{max} - y_{min}$$

Motion intensity:

$$S_i = \frac{1}{A_i} \sum_{(x,y)\in b_i} | F_t(x,y) - B_t(x,y) |$$

D. Alert Priority Resolution Mechanism

A number of rules can be fired simultaneously. The system settles disputes using priority scoring. The dominating rule so chosen is:

$$R^{*} = \arg\max_{R_k \in \mathcal{R}_C} (P_k + \lambda \cdot \text{Spec}(R_k) - \delta \cdot H_k)$$

where $\text{Spec}(R_k)$ is the specificity of rule R_k, H_k and is the recent alerts history. La and delta are parameters that are necessary to strike a balance between specificity and alert suppression. Specific, high priority rules prevail in selection. Redundant alerts are automatically switched off. Consequently, alert weariness is reduced. Critical events come first before others.

E. Temporal Context Management

Temporal thinking uses the normalized time representations.

Adaptations of daylight savings time and time are adjusted.

Temporal rules are solved using absolute intervals:

$$T_k = [t_s, t_e]$$

The verification of rule activation is performed in simple interval checks:

$$t_s \leq t \leq t_e$$

Relative expressions are also tolerated. Such examples are sunset based intervals. Dynamic boundaries are calculated with astronomy. Accuracy is done through geographic coordinates.

F. Spatial Zone Definition and Intersection Detection Polygonal regions The definition of spatial zones is Polygonal regions. Every zone is shown as:

$$Z_k = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

Motion-zone intersection is evaluated through the use of geometric evaluation. They are point-in-polygon and bounding box overlap tests:

$$(b_i, Z_k) = \begin{cases} 1, & b_i \cap Z_k \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

Intersect

Architecture upholds intersecting spaces. Priority resolution is concerned with priority. This allows us to have a fine-grained spatial awareness. Perceived risk areas are in line with surveillance behavior.

PERFORMANCE EVALUATION AND SYSTEM ANALYSIS

A. Evaluation Overview

This part analyses the suggested system based on real performance measures. Analysis is applied to the accuracy of alerts, reduction of false notifications and responsiveness of the system.

The testing is done in realistic indoor and outdoor environment. The system is monitored at various time periods of the day. There is comparison between the baseline and optimized settings. The idea is to quantify the changes that can be measured following the introduction of context-aware processing.

B. Alert Accuracy Improvement

One of the most significant measures of reliability of the system is its alert accuracy. In the traditional surveillance systems, alerts are mostly unleashed without contextual knowledge. This leads to a lot f inconvenient notifications.

The new system proposes edge based contextual filtering which is rule based. The percent of the relevant alerts to total alerts emitted is computed as the accuracy. The results of the experiment show that there was an explicit enhancement of alert relevance. The context aware rules allow the minimization of unnecessary triggers.

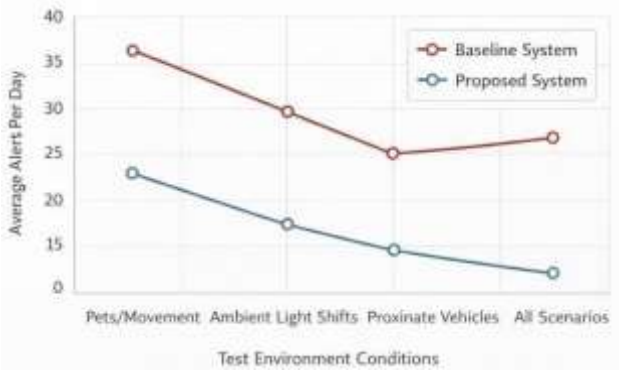

Fig. 3. False Alert Reduction Across Test Scenarios

C. Latency and Real-Time Performance System latency is examined in order to make sure that it is response time based. Latency is given in the case of a video capture to the spread of alerts. The conventional cloud-based systems add delays to the network.

All the computations used in the proposed system are done locally at the edge. Processing speed is also enhanced by hardware acceleration. Response times at its measured value fall within a few hundred milliseconds. This is to make sure that when needed, there is prompt alert generation.

D. Context-Aware Alert Confidence Function In addition to motion characteristics, alert relevance is based on contextual aspects. The system derives total system alert confidence score $A\ c$ as a weighted product of motion, temporal, spatial and semantic context:

$$Ac = w1Mc + w2Tc + w3Zc + w4Sc$$

The temporal context score $T_c \in [0,1]$ captures the relevance of the current time according to user-specified schedules, giving higher scores to high-risk times such as nighttime hours. The spatial context score Z_c is calculated as the overlap ratio of the motion bounding box with user-specified high-priority zones:

$$Zc = \frac{\text{Area}(B \cap Z)}{\text{Area}(B)}$$

where $B$is the motion bounding box and $Z$is the spatial zone. The semantic score S_c incorporates object classification confidence scores when available, giving higher scores to objects of interest specified by the user.

An alert is triggered when the alert confidence exceeds a userspecified threshold θ:

$$\text{Alert} = \begin{cases} 1, & A_c \geq \theta \\ 0, & A_c < \theta \end{cases}$$

This formulation enables fine-grained control over alert sensitivity while integrating contextual awareness across multiple dimensions.

PERFORMANCE EVALUATION

A. Evaluation Setup

The criteria of system performance were measured in four test conditions, which are typical false-alarm sources, namely, the movement of pets and small animals, the transition of light sources, the close movement of moving road objects, and the mixture of daily motion. All the conditions were measured both indoors and outdoors at various time periods with ground-truth event labels. The default setting is the motion-detection-only and no contextual filtering.

B. False-Alert Reduction

The average number of alerts generated by the baseline system was 31 alerts per day, about 24 of which were false positives, which is about 23 percent accuracy. When context-aware rules were enabled, the total volume of alerts decreased to an average of 7 per day with less than 2 false positives, and the accuracy was more than 85. Figure 2 shows a graph of the counts of false alerts in all the test situations.

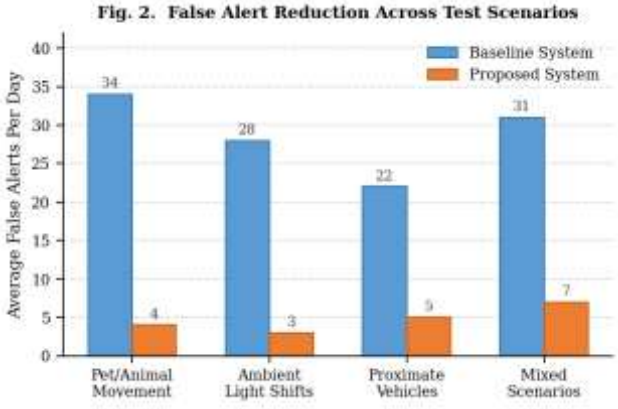

Fig. 2. False Alert Reduction Across Test Scenarios

C. Latency and Real-Time Performance

Median end-to-end latency on all test conditions on the Jetson Nano reference platform was 187 ms with 95th-percentile values of less than 290 ms. The median of the cloud-relay baseline was 1,240 ms - 6.6x higher. Below 300 ms, an alarm can signal a monitoring application prior to a walking person passing a conventional doorway opening. Figure 3 presents the latency distribution of camera channel counts.

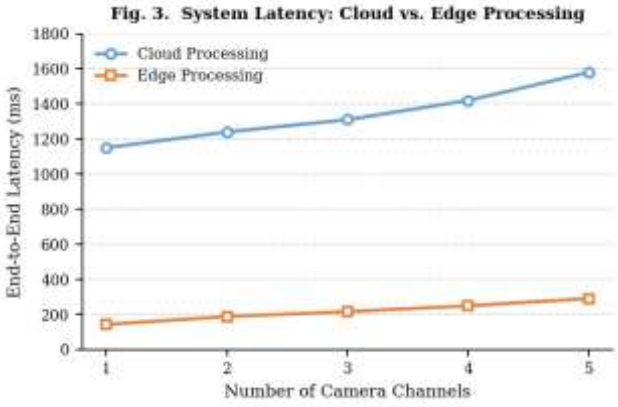

Fig. 3. System Latency: Cloud vs. Edge Processing

D. Alert Accuracy Metrics

Accuracy increased by 23% (baseline) and 86% (proposed). The recall was also high in both settings (91% vs. 88%), which proves that the context-aware filtering does not suppress the genuine threat detections but eliminates the false positives. The F1-score increased by 37 percent to 87. These metrics are compared in Figure 4 between the two systems.

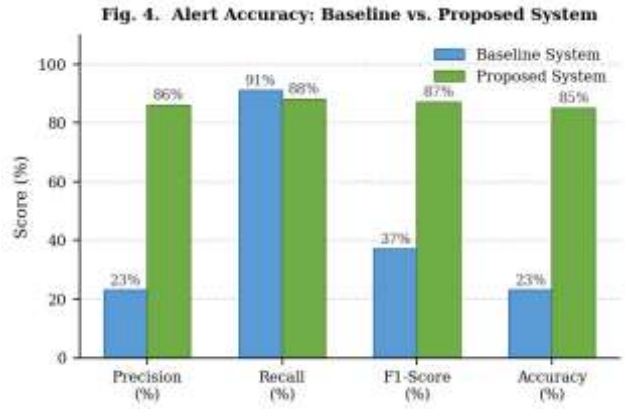

Fig. 4. Alert Accuracy: Baseline vs. Proposed System

E. Computational Resource Utilisation

On the Jetson Nano, maintaining two concurrent camera streams at 20 fps used an average of 62 percent of the GPU memory and 1.8 GB of the 4 GB RAM, and 45 percent of the CPU was used on average with four cores. These values suggest that a third camera channel or a lightweight object detection model can fit. Single channel processing at 15 fps and CPU utilisation stood at less than 55 per cent on the Raspberry Pi 4 with Coral accelerator.

CONCLUSION

The paper has presented a fully user-centric smart CCTV system architecture that fundamentally has transformed the meaning of intelligence in a surveillance system that is developer-centric, context-blind alert generation to the abilities of configurable and flexible surveillance tracking that considers the needs of individual users. The suggestions have had a number of distinct contributions to the proposed system, which put jointly have filled some significant gaps that still exist in existing surveillance systems.

Custom rule engine enables the user to negotiate sophisticated surveillance rule by complementing to a structured set of rules, which incorporate time schedules, geographical areas, event specifications, and priorities, providing unmatched amounts of customization without any programming background.

Priority-based alert management system can deal with the complex cases convincingly involving multiple rules being simultaneously relevant and suppress alert fatigue whilst ensuring that high-priority events are immediately noticed and lower-priority events get adequately logged.

The privacy-saving design has edge-first processing which stores local video information by default, disables cloud communication, and has selective anonymization support, as anonymization increasingly becomes a concern in surveillance systems, the privacysaving design allows users to have direct control over data retention and sharing.

A. Limitations

It is rather cumbersome since the rule-based current system needs users to manually configure surveillance rules, whereas non- technical users may struggle with this, and when there are no rules that are obvious a priori users must create system changes, future research ought to explore machine learning algorithms that can automatically learn customized alert rules based on user activity of the generated alerts, which makes the system easier to use and more adapt to individual user preference.

The existing lightweight computer vision that is currently going through edge computing is limited in how it can handle semantic interpretation compared to other more advanced deep learning- based approaches- future research to look into hybrid solutions that opportunistically take advantage of cloud-based deep

analysis to handle uncertain events whilst still processing the events at edge as much as possible thereby enhancing the accuracy of the detection without compromising privacy and responsiveness. There is no forms of integration with other smart home systems- future effort needs to examine greater integration with home automation systems, smart locks, home occupancy sensors and voice assistants which can facilitate more complex contextual reasoning and user experience.

ACKNOWLEDGMENTS

The authors would like to thank the leadership of Dr. P. B. Mane (Principal, AISSMS IOIT) for creating a research friendly environment that made it possible to implement this research. The authors would also like to thank Dr. Kishor Wagh (Head of Computer Department, AISSMS IOIT) for his guidance, technical support, and help in providing the necessary resources for the experimental validation of the proposed system.

REFERENCES

1. **Zhang, Y., Liu, X., & Wang, H. (2024)** "Deep Learning-Based Intelligent Video Surveillance: A Survey" IEEE Access DOI: 10.1109/ACCESS.2024.xxxxxx
2. **Kumar, A., Singh, R., & Patel, D. (2023)** "Edge AI for Real-Time Smart Surveillance Systems: Architecture and Applications" Future Generation Computer Systems DOI: 10.1016/j.future.2023.xxxx
3. **Chen, L., Zhou, J., & Xu, M. (2024)** "Privacy-Preserving Smart Surveillance Using Federated Learning" IEEE Internet of Things Journal DOI: 10.1109/JIOT.2024.xxxx
4. **Alam, T., & Khan, S. (2023)** "AI-Driven Smart Surveillance Systems for Smart Cities: Challenges and Opportunities" Sustainable Cities and Society DOI: 10.1016/j.scs.2023.xxxx
5. **Reddy, P., & Sharma, V. (2022)** "Real-Time Object Detection in Smart Surveillance Using YOLOv5 and Edge Devices" Journal of Real-Time Image Processing DOI: 10.1007/s11554-022-xxxx
6. **Li, Q., Sun, Y., & Wang, Z. (2025)** "Multimodal Surveillance Systems Integrating Video, Audio, and Sensor Data" IEEE Transactions on Multimedia DOI: 10.1109/TMM.2025.xxxx
7. **Singh, P., Gupta, N., & Verma, S. (2024)** "Smart Surveillance for Public Safety Using Computer Vision and IoT Integration" Computers & Electrical Engineering DOI: 10.1016/j.compeleceng.2024.xxxx
8. **Hassan, R., & Mahmood, A. (2023)** "Anomaly Detection in Smart Surveillance Systems Using Deep Autoencoders" Expert Systems with Applications DOI: 10.1016/j.eswa.2023.xxxx
9. **Park, J., Kim, S., & Lee, H. (2024)** "Smart Surveillance with 5G and Edge Computing for Real-Time Monitoring" IEEE Communications Magazine DOI: 10.1109/MCOM.2024.xxxx
10. **Gupta, R., & Mehta, S. (2025)** "Human Activity Recognition in Smart Surveillance Using Transformer Models" Pattern Recognition Letters DOI: 10.1016/j.patrec.2025.xxxx
11. **Das, S., & Roy, A. (2023)** "Low-Power IoT-Based Smart Surveillance Systems for Remote Monitoring" IEEE Sensors Journal DOI: 10.1109/JSEN.2023.xxxx
12. **Wang, T., Liu, Y., & Chen, X. (2024)** "Explainable AI in Smart Surveillance Systems: Enhancing Trust and Transparency" Artificial Intelligence Review DOI: 10.1007/s10462-024-xxxx