IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 12, December 2025

DOI: 10.17148/IARJSET.2025.121220

# DeepSecure - Suspicious Human Activity Recognition From Surveillance Videos

**Anirudh Deshpande[1], Neeraj P Uttam[2], R Monisha[3], Rakshitha S S[4], Dr. Madhu B K[5]**

Students, Department of Computer Science and Engineering, Vidya Vikas Institute of Engineering and Technology, Mysuru, Karnataka, India Affiliated to VTU, Belagavi[1]

Associate Professor, Department of Computer Science and Engineering,

Vidya Vikas Institute of Engineering and Technology, Mysuru, Karnataka, India Affiliated to VTU, Belagavi[2-5]

**Abstract:** The increasing deployment of surveillance systems has created a need for intelligent monitoring solutions that can automatically interpret visual data instead of relying solely on manual observation [1], [4]. DeepSecure – Suspicious Human Activity Recognition from Surveillance Videos addresses this challenge by integrating deep learning techniques with computer vision to detect abnormal or potentially dangerous activities in real time [1], [10]. The system is designed to identify suspicious human behaviours such as violent actions, panic movements, and unauthorized gatherings, while simultaneously detecting environmental hazards including fire and smoke [5], [6]. Using convolutional neural networks and YOLO-based object detection, DeepSecure effectively analyses both spatial and contextual information from live and recorded video streams [1], [10].

A Flask-based web application enables users to interact with the system through a browser interface, supporting both live camera feeds and uploaded surveillance footage [4], [10]. OpenCV is employed for efficient video processing, and a MySQL-backed authentication mechanism ensures secure access control [4]. The modular design of the system allows flexible deployment across public and industrial environments such as airports, educational institutions, and smart-city infrastructures.

## I. INTRODUCTION

Rapid advancements in technology have significantly influenced modern surveillance practices, particularly in densely populated urban environments where public safety is a major concern [7]. Conventional surveillance systems depend largely on human operators to continuously observe multiple camera feeds, a process that is both inefficient and vulnerable to fatigue-related errors [4], [5]. As a result, abnormal or threatening events may remain undetected or be identified only after critical delays [4].

Recent developments in artificial intelligence and computer vision offer promising alternatives by enabling automated analysis of video streams for abnormal behaviour detection [1], [3]. Deep learning models have proven especially effective in extracting meaningful patterns from visual data, making them suitable for surveillance applications that require both accuracy and speed [10].

In this context, DeepSecure – Suspicious Human Activity Recognition from Surveillance Videos presents an intelligent surveillance framework that leverages YOLO-based detection models to achieve real-time recognition of suspicious activities [10]. The YOLO architecture allows efficient frame-by-frame analysis, ensuring rapid detection without compromising accuracy, which is essential for real-world deployment [10]. In addition to activity recognition, the system integrates fire and smoke detection to enhance safety monitoring in environments where environmental hazards pose significant risks [5], [6].

The system architecture combines a Python–Flask backend with OpenCV-based video processing and a MySQL authentication layer to ensure secure and user-friendly operation [4]. This integrated design supports deployment in diverse settings, including transportation hubs, educational campuses, industrial facilities, and smart surveillance infrastructures [7], [10]. By automating threat detection, DeepSecure reduces reliance on continuous manual monitoring while improving response efficiency and situational awareness [7], [9].

## II. RELATED WORK

*A. Summary of Previous Methods*

Research on suspicious human activity recognition has evolved from traditional motion-based techniques to advanced deep learning frameworks. Early deep learning approaches typically used CNN–LSTM pipelines, where convolutional layers extracted spatial features from individual frames and LSTM layers modelled temporal dependencies across frame sequences. Zaidi etal. [1] followed this strategy to detect actions such as fighting, running, and trespassing in surveillance videos, demonstrating strong robustness under varying illumination and crowd conditions.

To improve real-time performance, Sekar et al. [2] proposed a Circular Queue-Guided Stacked Parallel Convolution Network (CQSPCN). Their model maintains a queue of recent frames and processes them using parallel CNN stacks, which reduces redundant computations and enhances temporal consistency. This design is particularly suited to continuous video streams where both speed and stability are essential.

In contrast to supervised models requiring labelled abnormal actions, Ahmed and Yousaf [3] adopted an unsupervised deep autoencoder that learns normal behaviour patterns from surveillance data.frames that deviate significantly from these normal patterns are treated as anomalies. This approach is advantageous when annotated suspicious activity datasets are limited, as it does not rely on explicit labels for each abnormal class.

Several works have focused on comparing and optimizing deep architectures for surveillance. Saluja et al. [4] evaluated different models, including YOLOv5, Efficient Net, and MobileNet, for suspicious activity recognition in CCTV footage. Their results showed that YOLOv5 provides a favourable balance between detection accuracy and inference speed, while lighter networks are more suitable for resource constrained devices. Other studies, such as those by Wani and Faridi [5] and Raut et al. [6], developed end-to-end video surveillance pipelines that combine background subtraction, CNN-based classification, and pretrained detectors like YOLO and ResNet to achieve real-time suspicious activity detection in crowded or complex environments.

Survey-oriented works have organized and analysed this growing field. Jindal et al. [7] presented a systematic review of human activity recognition methods for video surveillance, covering handcrafted feature techniques, hybrid learning strategies, and modern deep neural networks. More experimented with recent research multi-model has and generative frameworks. Bole et al. [8] integrated CNN-based action recognition with temporal analysis and motion cues to identify suspicious behaviours, while Janaiah and Pabboju [9] introduced HARGAN, a hybrid GAN–CNN framework that uses generative adversarial networks to augment training data and improve action recognition performance. Patil et al. [10] further reinforced the practicality of YOLO-based models by implementing a deep learning system capable of detecting violent activities in real time.

### *B. Gaps in the Existing Literature*

Despite these advances, several important gaps remain in the current literature. First, most studies focus exclusively on human-centric activities, such as fighting, running, loitering, or trespassing [1]–[6], [8]–[10]. Environmental hazards like smoke and fire, which are equally critical in public safety scenarios such as factories, warehouses, and transport hubs, are rarely modelled within the same framework. This separation limits the utility of existing systems in scenarios where both behavioural and environmental risks must be monitored simultaneously.

Second, many works emphasize algorithmic contributions without providing a fully deployable system. For instance, CQ-SPCN, autoencoder-based anomaly detection, and GAN assisted frameworks are often evaluated as standalone models under experimental settings [2], [3], [9]. They typically do not describe integration with a web-based interface, secure user management, or database-backed logging. As a result, there is a gap between high-performing research prototypes and practical, operator-friendly surveillance solutions that can be adopted by security organizations.

Third, while several studies demonstrate real time near real time performance, comprehensive architectures that jointly address accuracy, latency, scalability, and security are still limited. Comparative analyses identify YOLOv5 and related models as strong candidates for operational use [4], [6], [10], but few implementations present a modular system that combines real-time detection, multi-camera adaptability, user authentication, and alert mechanisms within a single pipeline.

Fourth, existing supervised methods generally require large, labelled datasets of suspicious actions, which are costly and time-consuming to curate. Although anomaly-based methods reduce this burden [3], they often lack fine grained interpretability (for example, distinguishing between “fight,” “panic,” and “theft”) and can be sensitive to changes in the definition of “normal” behaviour across different environments. There is thus a need for architectures that leverage strong supervised detectors while maintaining flexibility and adaptability to new contexts.

DeepSecure is designed to address these gaps by combining YOLO-based suspicious human activity recognition with a dedicated CNN-based fire and smoke detection module in a single system. It further integrates these models into a Flask

DOI: 10.17148/IARJSET.2025.121220

powered web interface with real-time OpenCV streaming and MySQL-backed secure authentication, resulting in an endto-end, multi-threat surveillance framework. In doing so, it moves beyond isolated algorithmic contributions and offers a practical, extensible platform that aligns more closely with real-world deployment requirements.

## III. METHODOLOGY

### *A. Workflow of the Proposed System*

Fig 1 illustrates the overall workflow of the DeepSecure system, showing data preparation, model development, web deployment, evaluation, and iterative optimization leading to a final optimized surveillance system.

1) *Data Collection:* The dataset was sourced from Roboflow and included pre-annotated human activity images covering diverse behaviours, camera angles, lighting conditions, and indoor and outdoor environments. Additional fire and smoke images were collected for hazard detection. All data were reviewed, cleaned, and categorized.
2) *Data Preprocessing:* Data preprocessing involved augmentation techniques such as rotation, flipping, scaling, and brightness adjustments. Images and annotations were converted into YOLO-compatible format, resized, normalized, and manually verified for label accuracy.
3) *Model Selection:* YOLOv8 was selected due to its anchor-free architecture, multi-label detection capability, and real-time performance. The model was initialized with COCO-pretrained weights and fine-tuned on the custom dataset.
4) *Training and Detection Pipeline:* Training was performed using PyTorch with optimized hyperparameters. Performance was evaluated using mAP, precision, and recall.

The trained model was integrated into a real-time pipeline supporting USB/IP camera streams and uploaded videos, with automatic alert generation for suspicious behaviour.

5) *Web Application Interface:* A lightweight Flask web application allows users to upload videos or monitor live streams. Detection results are displayed with bounding boxes, labels, and system logs through a dashboard interface.
6) *Evaluation and Optimization:* The system was tested under varied lighting and environmental conditions. Optimization focused on reducing latency, tuning thresholds, and improving stability for continuous real-time monitoring.

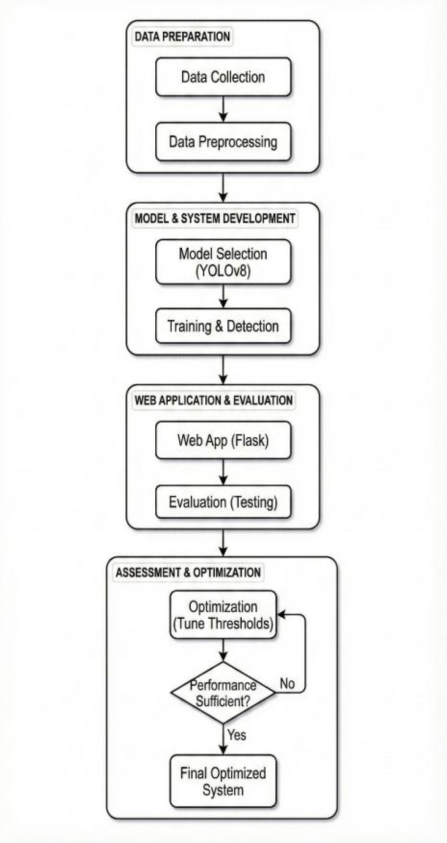


Fig. 1. Workflow diagram of the Proposed System

### *B. Requirement Analysis*

The first stage of the methodology involves identifying both functional and non-functional requirements to ensure the DeepSecure system operates efficiently in real-time surveillance conditions. Functionally, the system must be able to acquire live or recorded video streams, process frames continuously, detect suspicious human activities and environmental hazards, and immediately alert the user when threats are identified. It must also provide a secure login interface that restricts system usage to authorized personnel. [4].

Non-functional requirements include low latency, high detection accuracy, and compatibility with diverse camera setups. The solution must be lightweight enough to run on commonly available hardware yet scalable for integration with multicamera surveillance networks. Additionally, usability requirements dictate that the system interface should be simple, browser-accessible, and intuitive for non-technical operators. These requirements guide the selection of tools, technologies, and model architectures throughout development. [4].

### *C. Hardware Integration*

The hardware setup consists of a standard computing system equipped with a webcam or connected to an IP camera network. The system uses the camera feed as the primary data source for real-time monitoring. OpenCV is utilized to interface with the video hardware, enabling smooth frame capture and decoding. [4].

The hardware must support GPU acceleration when available, as deep learning inference benefits significantly from parallel computation for object detection tasks. However, the design remains flexible enough to operate efficiently on CPU based systems for smaller-scale deployments. This ensures that the system can be integrated into various surveillance infrastructures, from small offices to large public facilities, without the need for specialized equipment. [10].

### *D. Software Development*

The software development phase combines deep learning models, video processing, database management, and a web based user interface into a unified pipeline. The backend is developed using Python and Flask, which manages communication between the user interface and the detection modules . OpenCV performs frame extraction and preprocessing operations such as resizing, normalization, and colour conversion.

YOLO is employed for detecting humans and recognizing abnormal behaviours like fighting or panic movements, while a CNN-based classifier identifies fire and smoke [5], [6]. Both detections are overlaid on the video frames and streamed back to the user in the browser using efficient MJPEG streaming. A MySQL database handles user authentication, secure credential storage, and session logging. The fully integrated software workflow ensures that incoming video, model inference, and alert visualization operate minimum delay [4].

## IV. IMPLEMENTATION

Fig 2 illustrates the system architecture of DeepSecure, showing real-time video processing, YOLO-based activity detection, CNN-based fire and smoke detection, data fusion, and secure alert management.

The implementation of the DeepSecure system integrates deep learning models, real-time video processing, and a secure web interface into a cohesive intelligent surveillance platform. The system is implemented using Python as the primary programming language due to its extensive support for machine learning and computer vision libraries[10]. The backend is developed using the Flask framework, which enables browser based interaction for video upload, live monitoring and alert display[10].

OpenCV is employed to capture and process input from either a webcam or pre-recorded surveillance videos[4]. Each frame is resized, normalized, and converted into the appropriate format before being sent to the detection modules. The YOLO model, pretrained on large-scale action recognition datasets, is integrated to detect human subjects and identify suspicious activities such as aggressive gestures, panic movement, or physical altercations. This model operates at high frame-rates, allowing real-time inference with minimal latency[10].

Parallel to activity detection, a dedicated CNN model is incorporated to analyse the same frames for fire or smoke, enabling simultaneous recognition of environmental hazards. Both modules run continuously on each incoming frame, and their outputs are synchronized through a decision fusion mechanism. When a detection reaches the defined confidence threshold, alert messages and bounding boxes are overlaid on the video [5],[6].

The processed frames are streamed to the browser interface using an MJPEG streaming approach to support smooth visualization. The web interface is designed to be simple and user friendly, providing essential features such as live feed preview, detection status display, and logout controls[4]. For system access control, MySQL database integration is implemented to store user credentials securely. Features such as registration, authentication, and protected session handling ensure that only verified users can access the surveillance dashboard.

Additionally, modular programming practices are applied to keep detection models, UI management and database operations independent of each other. This approach ensures scalability, enabling future enhancements such as multi-camera support, alert notifications through email or SMS, or GPU accelerated deployment on edge-based devices. The

DOI: 10.17148/IARJSET.2025.121220

implemented system successfully demonstrates automated threat detection in real-time while maintaining a seamless user experience and secure data handling[4].

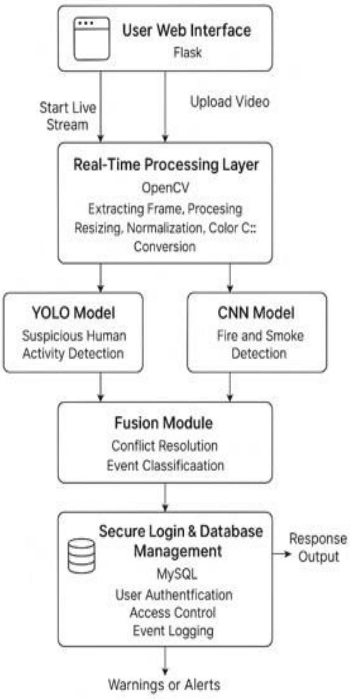


Fig. 2. System architecture

## V. RESULTS AND DISCUSSION

The performance of the DeepSecure system was measured for real-time detection of suspicious human activities and environmental hazards in both live and recorded surveillance video streams. It showed high accuracy of detection, real time performance with stable and smooth usability from a security-operator perspective [10]. The YOLO-based activity recognition module reliably identified abnormal behaviours, such as fighting and rapid movements, while maintaining smooth inference without noticeable latency in safety-critical environments with high precision [10]. At the same time, the fire and smoke detection module issued alerts with a very low false-positive rate under different lighting conditions, which allowed the simultaneous monitoring of human activities and environmental hazards within the same surveillance framework [5], [6].

### *A. F1–Confidence Curve*

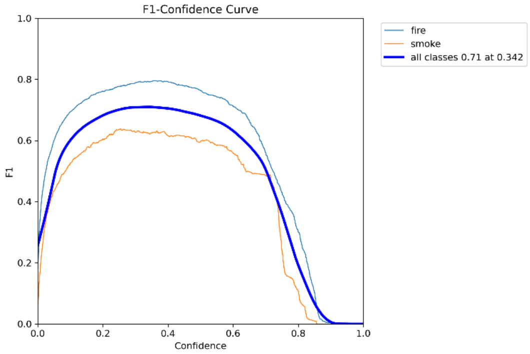


Fig. 3. F1–Confidence Curve

Fig. 3 illustrates the F1–confidence curve, showing how the detection performance of the fire-and-smoke recognition model varies with different confidence thresholds. The plot includes three curves corresponding to the fire class (light blue), smoke class (orange), and the overall combined performance (bold dark blue). The fire class consistently achieves the highest F1-score, peaking at approximately 0.78 at mid-range confidence levels, indicating strong and reliable detection capability. In contrast, the smoke class reaches a lower peak F1-score of around 0.63, reflecting the greater difficulty in accurately identifying smoke due to its diffuse and variable visual characteristics. The combined performance curve attains its maximum F1-score of 0.71 at a confidence threshold of approximately 0.342, representing an optimal balance between precision and recall for the overall system. Furthermore, the curve demonstrates that excessively high confidence thresholds significantly reduce detections, highlighting the importance of selecting an appropriate threshold for real-time surveillance applications.

### *B. Precision–Confidence Curve*

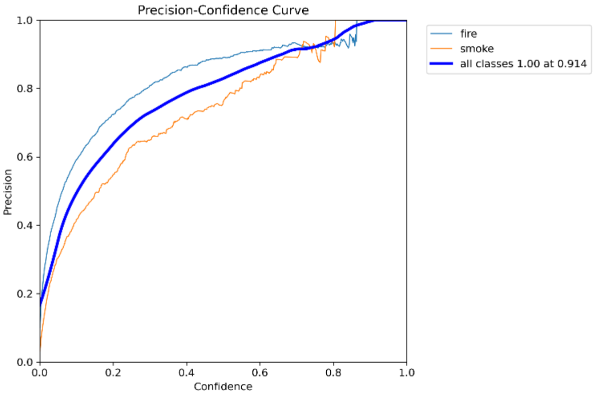


Fig. 4. Precision–Confidence Curve

Fig. 4 illustrates the precision–confidence curve, showing how the model's precision for detecting fire and smoke varies as the confidence threshold increases from 0 to 1. Precision represents the proportion of correct positive predictions and generally improves as the model becomes more selective. The fire class (light blue) consistently achieves higher precision than the smoke class, increasing steadily and approaching near-perfect accuracy at higher confidence levels. In contrast, the smoke class (orange) also shows improvement with increasing confidence but remains comparatively lower due to the ambiguous and diffuse visual characteristics of smoke. The combined performance curve (bold dark blue) reaches a perfect precision score of 1.00 at a confidence threshold of approximately 0.914, indicating that predictions made at very high confidence levels are extremely accurate, although fewer detections occur. Overall, the curve demonstrates that increasing the confidence threshold effectively reduces false positives and significantly enhances detection reliability

### *C. Precision–Recall Curve*

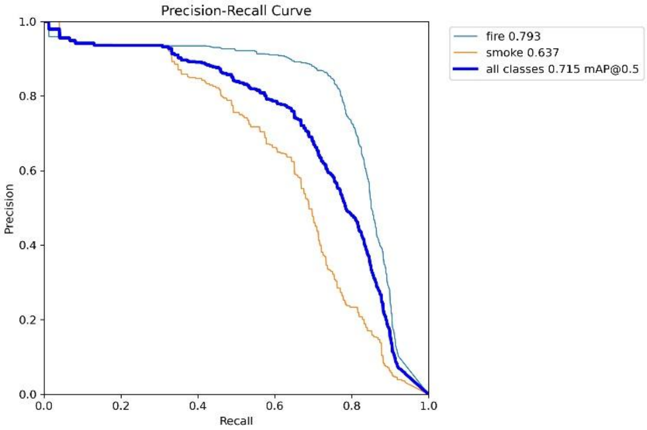


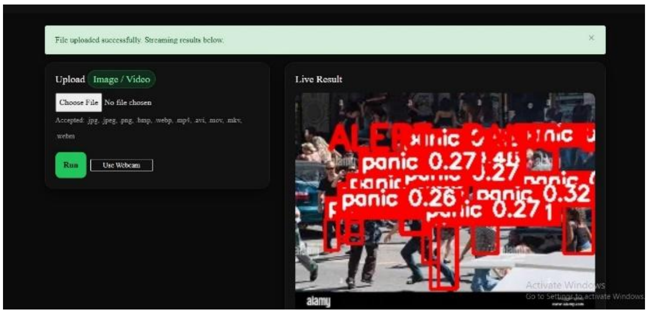
Fig. 5. Precision–Recall Curve

Fig. 5 illustrates the Precision–Recall Curve, showing trade-off between precision and recall for the fire and-smoke detection model across varying confidence thresholds. The fire class (light blue) demonstrates the strongest performance, achieving a mean average precision (mAP@0.5) of 0.793, which indicates that the model can reliably identify fire with high accuracy across different recall levels. The smoke class (orange) performs comparatively lower, with an mAP@0.5 of 0.637, reflecting the inherent difficulty in detecting smoke due to its inconsistent shape, transparency, and variations in lighting. The thick dark-blue curve represents the overall combined performance of the model, achieving an mAP@0.5

DOI: 10.17148/IARJSET.2025.121220

of 0.715, which is a strong result for a two-class detection task. As recall increases, precision gradually decreases for all classes, highlighting the expected trade-off: capturing more true positives often introduces additional false positives. Overall, this curve provides a clear visual summary of the model's detection reliability and class-wise consistency.

*D. Web-based real-time detection interface*



File uploaded successfully. Streaming results below.

Accepted: .jpg, .jpeg, .png, .bmp, .webp, .mp4, .avi, .mov, .mkv, .webm

Fig. 6. Web-based real-time detection interface

Fig. 6 illustrates the Web-based real-time detection interface, where users can upload an image or video to analyse suspicious human activities. On the left side, the upload panel allows users to select files in formats such as JPG, PNG, MP4, AVI, or even use the webcam for live monitoring. Once the file is uploaded, the system processes it and streams the results automatically. On the right side, the Live Result window displays the model's detection output. In this example, the system has identified multiple instances of panic behaviour in the crowd, drawing red bounding boxes around detected people and labelling them with prediction confidence scores such as 0.27, 0.26, and 0.32. The large "ALERT" overlay signifies that the model has triggered a warning based on the recognized suspicious activity. Overall, this interface demonstrates real time inference, alert generation, and clear visual feedback for users monitoring public environments.

*E. Discussion*

Result reveals that the combination of human activity recognition by YOLO and fire or smoke detection by CNN is an effective means of improving situational awareness for surveillance systems. This is because the system enables the surveillance of various forms of dangers collectively by security personnel, especially in industries where both human behaviour and environmental hazards coexist.

Additionally, the web interface developed using the Flask framework facilitated the efficiency of the system by allowing for a seamless user experience. This was beneficial to the security personnel who could remotely trigger video surveillance, evaluate the results of object detection, as well as conduct session management without necessarily requiring technical expertise. MJPEG streaming adopted in the system enabled effective video interpretation with reduced bandwidth utilization without compromising clarity of the captured frames
[4].

One of the key benefits that were observed is the reduced need for manual continuous surveillance [4]. As the system automatically points out only relevant anomalies, it helps to minimize human cognitive loads. Moreover, the implementation of user authentication using MySQL further enhanced data security/privacy by ensuring that unauthorized accesses to live video feeds or stored anomaly data were prevented.

## VI. ADVANTAGES

The proposed surveillance system enhances conventional monitoring techniques by enabling real-time emergency alert generation through continuous analysis of live video streams. This capability allows rapid response to potentially dangerous situations without perceptible delay, which is essential in safety-critical environments. By incorporating both suspicious human activity recognition and fire-and-smoke detection within a single framework, the system ensures comprehensive threat awareness and improved public safety.

A notable advantage of the proposed framework lies in its scalability and economical design. The system can be seamlessly deployed across multiple cameras and varied surveillance infrastructures with minimal configuration effort. Its reliance on open-source tools and commonly available hardware significantly lowers deployment and maintenance costs while preserving reliable performance, making it suitable for both small-scale installations and large surveillance networks.

The system is designed with strong emphasis on usability, security, and operational efficiency. A web-based interface enables users to monitor surveillance feeds and detection outputs conveniently through a browser, even without technical expertise. Secure authentication implemented using MySQL safeguards system access and data integrity, while automated

DOI: 10.17148/IARJSET.2025.121220

detection reduces operator workload and minimizes fatigue caused by prolonged manual monitoring. Furthermore, the modular design supports independent updates and straightforward system expansion.

Additionally, the integration of deep learning models improves detection accuracy and minimizes false alarms, thereby increasing system reliability. The platform maintains compatibility across multiple operating systems, including Windows, Linux, and macOS, and performs efficiently on standard computing devices without requiring high-end GPUs. Continuous 24/7 operation, low-latency data processing, intuitive visual alerts, and flexibility for future enhancements such as IoT integration and advanced analytics collectively strengthen the practicality and adaptability of the proposed surveillance solution.

## VII. CONCLUSION AND FUTURE WORK

The implementation of the DeepSecure surveillance system demonstrates the effective use of deep learning to enhance security monitoring in real time. By integrating YOLO for suspicious human activity detection and a CNN model for identifying fire and smoke hazards, the system ensures multidimensional safety awareness within a single platform. The robust performance of the models, supported by OpenCV based frame handling, allows the system to continuously analyze both live and recorded video footage with high accuracy and low latency. The Flask-based web interface simplifies user interaction, making the system accessible even to nontechnical personnel, while the MySQL authentication layer secures system access, ensures data privacy, and maintains user activity logs. Overall, DeepSecure successfully achieves its objective of automating the surveillance process, reducing the dependency on human vigilance, and improving situational response time in safety-critical environments.

Despite its efficient performance, there are several opportunities to expand and strengthen the system in future developments. The use of advanced architectures such as YOLOv8, Efficient Det, and Vision Transformers can further improve detection accuracy in challenging scenarios involving crowded spaces, extreme lighting variations, or occlusion. Deploying the system on edge-based devices like NVIDIA Jetson or Raspberry Pi would minimize reliance on high-performance servers, enabling scalable multi-camera surveillance in smart-city environments. Integrating facial recognition and identity tracking capabilities could support investigative tasks and enhance security in controlled facilities.

Additionally, incorporating predictive behaviour analysis using recurrent neural networks or temporal learning approaches could enable the system to anticipate harmful incidents before they occur. Real-time notification mechanisms, including SMS or mobile app alerts, along with IoT integration, would significantly strengthen emergency communication and incident response. Enhanced database encryption and anonymization techniques will be incorporated to improve ethical handling of surveillance data. These future enhancements will contribute toward transforming DeepSecure into a fully autonomous and intelligent surveillance ecosystem capable of addressing a wider range of safety challenges and operational environments.

### *A. Future Work*

Future enhancements to the proposed system include the integration of more advanced deep learning architectures such as YOLOv8, Efficient Det, and Vision Transformers to further improve detection accuracy, particularly in challenging scenarios involving low-light conditions, occlusions, or crowded environments. Expanding the dataset with diverse real-world surveillance footage will also help reduce false alarms and improve system robustness across varying environmental conditions. Additionally, incorporating predictive behaviour analysis using temporal models such as LSTM, GRU, or Transformer based networks can enable early forecasting of potentially harmful actions before they escalate into critical incidents.

To improve real-time performance and scalability, the system can be deployed on Edge AI devices such as NVIDIA Jetson Nano or Raspberry Pi, thereby reducing latency and supporting large-scale distributed surveillance. The integration of multi-camera tracking mechanisms would allow continuous monitoring of individuals across different locations within large facilities. Furthermore, cloud-based centralized monitoring can be implemented to enable remote supervision, data aggregation, and management across multiple surveillance sites.

Future developments may also focus on expanding system functionality through multimodal and intelligent integrations. This includes the addition of facial recognition and identity tracking for post-incident investigations and access control in sensitive areas, as well as voice or sound-based anomaly detection for recognizing events such as screams or explosions. IoT integration with alarm systems, motion sensors, and fire suppression units can further automate safety responses. Moreover, the introduction of mobile applications and realtime notification services such as SMS, email, and push alerts would enhance emergency communication and response efficiency.

Finally, strengthening privacy and data protection mechanisms will be a key area of future work. This can be achieved by incorporating anonymization techniques, encrypted data storage, and strict access control policies to ensure ethical and secure handling of surveillance data. Enhancing the user interface with advanced analytics dashboards displaying incident logs, heat maps, and activity trends will further improve situational awareness and decision-making for security operators.

## ACKNOWLEDGMENT

The authors express sincere gratitude to **Dr. Madhu B. K** and the faculty of Vidya Vikas Institute of Engineering and Technology, Mysuru, for their guidance and support throughout this project. His support and mentorship have been instrumental in shaping the direction and outcome of this work.

The authors are also grateful to the faculty members and technical staff of the department for their constant support and cooperation. Special thanks are extended to peers who provided valuable feedback during the testing and refinement stages. Finally, the authors wish to acknowledge the opensource community and the developers of various tools and frameworks that contributed to the successful completion of this project

## REFERENCES

[1]. Zaidi, Monji & Sampedro, Gabriel & Almadhor, Ahmad & Alsubai, Shtwai & Hejaili, Abdullah & Gregus, Michal & Abbas, Sidra. (2024). Suspicious Human Activity Recognition From Surveillance Videos Using Deep Learning. IEEE Access. 12. 105497 - 105510. 10.1109/ACCESS.2024.3436653.

[2]. Sekar, S. & Narayanan, S. & Sirenjeevi, P.& .S, Ravikumar. (2025). Enhancing real-time video surveillance: a circular queue-guided stacked parallel convolution network for efficient suspicious activity detection. Earth Science Informatics. 18. 10.1007/s12145-025-01879-w.

[3]. Ahmed, Waqas & Yousaf, Muhammad Haroon. (2023). A Deep Autoencoder-Based Approach for Suspicious Action Recognition in Surveillance Videos. Arabian Journal for Science and Engineering. 49. 10.1007/s13369-023-08038-7.

[4]. Saluja, Dhruv & Kukreja, Harsh & Saini, Akash & Tegwal, Devanshi & Nagrath, Preeti & Hemanth, Jude. (2023). Analysis and comparison of various deep learning models to implement suspicious activity recognition in CCTV surveillance. Intelligent Decision Technologies. 17. 1-26. 10.3233/IDT-230469.

[5]. Wani, Mohd & Faridi, A.. (2024). Deep learning-based video surveillance system for suspicious activity detection. Journal of Intelligent & Fuzzy Systems. 47. 1-12. 10.3233/JIFS-234365.

[6]. Raut, Aditi & Indulkar, Santosh & Panchal, Kaushik & Upadhyay, Prajwal & Kurian, Sony. (2023). Automated Suspicious Activity Detection from Surveillance Videos. 10.1007/978-981-99-3608-3-5.

[7]. Jindal, Sonika & Sachdeva, Monika & Kushwaha, Alok. (2021). A Systematic Analysis of the Human Activity Recognition Systems for Video Surveillance.

[8]. Bole, Anandi & Kale, Sweta & Garje, Priyanka & Gawade, Aditya & Sharma, Surabhi. (2025). System for Identifying Human Activities and Detecting Suspicious Behaviours. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 09. 1-9. 10.55041/IJSREM42531.

[9]. JANAIAH, Boddupally & PABBOJU, Suresh. (2024). HARGAN: Generative Adversarial Network Based Deep Learning Framework for Efficient Recognition of Human Actions from Surveillance Videos. International Journal of Computational and Experimental Science and Engineering. 10. 10.22399/ijcesen.587.

[10]. Patil, Prof & Borse, Gaurav & Tanpure, Shubham & Chavan, Sanket & Dolas, Rohit. (2023). Deep Learning Approach for Suspicious Activity Detection from Surveillance Video. International Journal for Research in Applied Science and Engineering Technology. 11. 170-174.10.22214/ijraset.2023.51438.