



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Automatic Surveillance Using Deep Learning

CH. Abhiram, Y. Amulya, D. Ashwanth Kumar, Chand Pasha, D. Suneetha

U.G. Student, Department of Computer Science & Engineering, Malla Reddy University, Hyderabad, Telangana, India

U.G. Student, Department of Computer Science & Engineering, Malla Reddy University, Hyderabad, Telangana, India

U.G. Student, Department of Computer Science & Engineering, Malla Reddy University, Hyderabad, Telangana, India

U.G. Student, Department of Computer Science & Engineering, Malla Reddy University, Hyderabad, Telangana, India

Associate Professor, Department of Computer Science & Engineering, Malla Reddy University, Hyderabad,
Telangana, India

ABSTRACT: Suspicious Activity Detection System is a real-time monitoring and analysis tool developed using Python with Tkinter, OpenCV, and Image AI libraries. The system integrates a modern graphical user interface (GUI) to facilitate video processing, frame generation, and suspicious activity detection from both live camera feeds and pre-recorded CCTV footage. Leveraging the DenseNet121 deep learning model for image classification, the system processes video frames to identify potential suspicious activities, employing temporal consistency and confidence thresholding to enhance detection accuracy. Key features include live feed toggling, video recording with timestamped storage, and automated frame extraction, with suspicious frames flagged and saved for further review. The GUI, styled with a sleek dark theme and gradient aesthetics, provides intuitive controls and dual text areas for logging frame generation and detection results. Advanced visualization overlays detection metrics on frames, while the system ensures robust error handling and resource cleanup. Designed for security applications, this tool demonstrates a scalable approach to real-time video analysis, balancing usability and performance. Future enhancements could include multi-camera support and expanded activity classification capabilities.

KEYWORDS: Suspicious Activity Detection, Deep Learning, Convolutional Neural Networks (CNN), Computer Vision, Object Detection, Gesture Recognition.

I. INTRODUCTION

In an age of extensive video surveillance, manually monitoring for suspicious activities are increasingly unfeasible, risking missed threats and slow responses. This project presents an Automated Suspicious Activity Detection System, utilizing real-time video analysis and deep learning to pinpoint anomalies accurately. Built with OpenCV for video processing, ImageAI's DenseNet121 for classification, and a Tkinter interface, it handles both live feeds and recorded footage. Employing temporal consistency and adjustable confidence thresholds, the system reduces false positives while flagging suspicious frames effectively. Designed to transform security, this solution automates detection, eases human workload, and boosts real-time decisions. Applicable to live monitoring and forensic review, it offers a scalable, intelligent approach to enhance safety across varied settings.

II. LITERATURE SURVEY

Suspicious activity detection in video surveillance has progressed remarkably, fueled by breakthroughs in computer vision, deep learning, and real-time processing technologies. These advancements seek to automate security monitoring, improving detection accuracy and efficiency in environments overwhelmed by vast video data. Traditional manual oversight struggles to cope with this scale, often missing critical anomalies and delaying responses. By leveraging sophisticated algorithms and models, recent research addresses these challenges, offering intelligent solutions to enhance safety across diverse applications, from public spaces to private facilities, setting the stage for more reliable and scalable surveillance systems.

1. Vision-Based Activity Recognition

Vision-based techniques are central to anomaly detection in surveillance systems. Studies have leveraged Convolutional Neural Networks (CNNs) such as ResNet to classify video frames with high accuracy, focusing on spatial feature extraction from complex scenes [1–2]. OpenCV, a cornerstone in this project's process_live_feed() for capturing and displaying frames, supports real-time video processing [3]. Research also explores object detection frameworks like YOLO, which enhance recognition by identifying specific elements within frames [4]. These approaches underpin this system's ability to process live feeds and generate frames for analysis, ensuring effective monitoring.

2. Deep Learning Approaches

Deep learning has revolutionized surveillance by enabling sophisticated pattern recognition. DenseNet121, implemented via ImageAI in prediction.classifyImage(), uses dense connectivity to achieve superior image classification, leveraging pre-trained weights for efficiency [5–6]. While this project employs frame-by-frame analysis, studies suggest integrating Long Short-Term Memory (LSTM) networks to model temporal dependencies across sequences [7], a potential enhancement for future iterations. Dense CNN architectures, as explored in related work, optimize performance in video-based tasks [8], reinforcing DenseNet121's suitability for detecting suspicious activities in this system.

3. Suspicious Activity Detection Systems

Automated detection systems employ temporal consistency via a 3-frame check in detectActivity() to reduce false positives by tracking prediction stability over time [9–10]. Frames are preprocessed like OCR using OpenCV [10]. A Tkinter GUI logs detections to enhance operator decisions in real time [11]. ImageAI's DenseNet121 classifies activities. Python code demonstrates a design with gradient styling and multi-threaded feed processing.

4. Optical Character Recognition (OCR) and Frame Preprocessing

OCR techniques are widely used in text detection but also play a role in frame preprocessing for video analytics. The system preprocesses frames using OpenCV, similar to OCR pipelines, where image enhancements improve text recognition accuracy. Studies highlight the importance of preprocessing for feature extraction in deep learning models [10]. The system applies frame extraction, resizing, and contrast adjustments, ensuring high-quality input for DenseNet121-based classification. This preprocessing stage helps eliminate noise, making it easier for the model to focus on relevant patterns. Such techniques are crucial for ensuring high detection accuracy in dynamic surveillance environments.

5. Graphical User Interface (GUI) for Real-Time Surveillance

A well-designed Graphical User Interface (GUI) plays a crucial role in real-time surveillance, improving operator efficiency and response time. The given system employs Tkinter to create an interactive and user-friendly interface, displaying live video feeds, logging detected activities, and generating alerts. Studies indicate that an intuitive GUI enhances situational awareness, allowing security personnel to respond promptly to threats [11]. The system includes real-time activity logging, detection confidence scores, and frame visualization, ensuring operators can verify suspicious events before taking action. Additionally, the interface provides easy access to recorded footage for post-event analysis. The integration of color-coded alerts and dynamic updates makes navigation more efficient. Future enhancements may include speech-based commands, touchscreen compatibility, and multi-window support, making the system even more accessible and effective for security professionals in high-risk environments.

III. PROBLEM STATEMENT

3.1 Existing System

Current video surveillance relies heavily on human operators to monitor feeds for suspicious activities, using basic tools like CCTV systems and manual frame review. Some automated systems employ traditional image processing with OpenCV or simple classifiers, as seen in early attempts to flag anomalies, but these lack advanced intelligence and real-time capabilities.

3.2 Limitations:

- 1. Requires Constant Oversight:** Manual monitoring demands uninterrupted human attention, causing fatigue and overlooked threats in extensive video feeds.
- 2. Limited Scalability:** Physical review restricts handling multiple camera inputs efficiently, overwhelming operators as surveillance networks grow.
- 3. Accuracy Shortfalls:** Basic automated tools using OpenCV lack adaptability, misidentifying activities and generating frequent false alerts.
- 4. Processing Lag:** Retrospective analysis of recorded footage, like frames in frames/, delays threat detection, impeding swift security responses.
- 5. Lack of Contextual Insight:** Existing systems miss nuanced behavior patterns, limiting depth in distinguishing normal from suspicious actions effectively.

3.3 Proposed System

This system automates suspicious activity detection using OpenCV for video capture, DenseNet121 for classification, and Tkinter for real-time monitoring, enhancing security efficiency:

- Processes camera feeds instantly with OpenCV.
- Identifies activities precisely using DenseNet121.
- Shows detection logs through a Tkinter GUI.
- Handles both live and recorded videos.

1. Video Processing:

OpenCV in `process_live_feed()` captures and manages video frames, supporting real-time surveillance and storage in videos/ for later review.

2. Real-Time Detection:

The system leverages OpenCV in `process_live_feed()` to detect threats as they occur, ensuring prompt identification within live streams.

3. Activity Classification:

DenseNet121, utilized in `prediction.classifyImage()`, analyzes frames with pre-trained precision, distinguishing suspicious behaviors effectively.

4. Temporal Consistency:

A 5-frame buffer with confidence thresholds (60%, 85%) in `detectActivity()` ensures stable detection across sequences, minimizing false alarms.

5. Desktop Interface: Tkinter in `main` provides a desktop interface, presenting detection results in `text1` and frame logs in text for user convenience.

6. Frame Storage: Frames flagged in `detectActivity()` are saved to `suspicious_frames/`, providing a repository for post-event analysis and evidence.

7. Video Recording: The system records live feeds to `videos/` via `process_live_feed()`, supporting forensic examination of captured footage.

IV. METHODOLOGY

4.1 System Overview:

The system aims to detect suspicious activities in real-time and recorded video, integrating OpenCV for frame capture, DenseNet121 for activity classification, and Tkinter for result visualization. It operates in two modes: live feed analysis via process_live_feed() and recorded footage processing through upload() and generateFrame(), with outputs stored in suspicious_frames/ and videos/ for review.

Key Technologies Used:

- **OpenCV:** For video capture, frame processing, and real-time display.
- **DenseNet121:** A pre-trained deep learning model from ImageAI for classifying activities within frames.
- **Threading:** Ensures seamless execution of live feed processing without interrupting the user interface.

4.2 System Architecture:

The system is structured into three interconnected modules: Input, Processing, and Output, each designed to handle specific tasks efficiently:

1. Input Module (Video Capture)

This module manages video input acquisition. In live mode, process_live_feed() captures footage from the default camera at 20 frames per second (FPS), storing frames in frames/ and recording video segments in videos/. For offline mode, upload() imports a video file, and generateFrame() extracts up to 500 frames, saving them as individual images in frames/ for further processing.

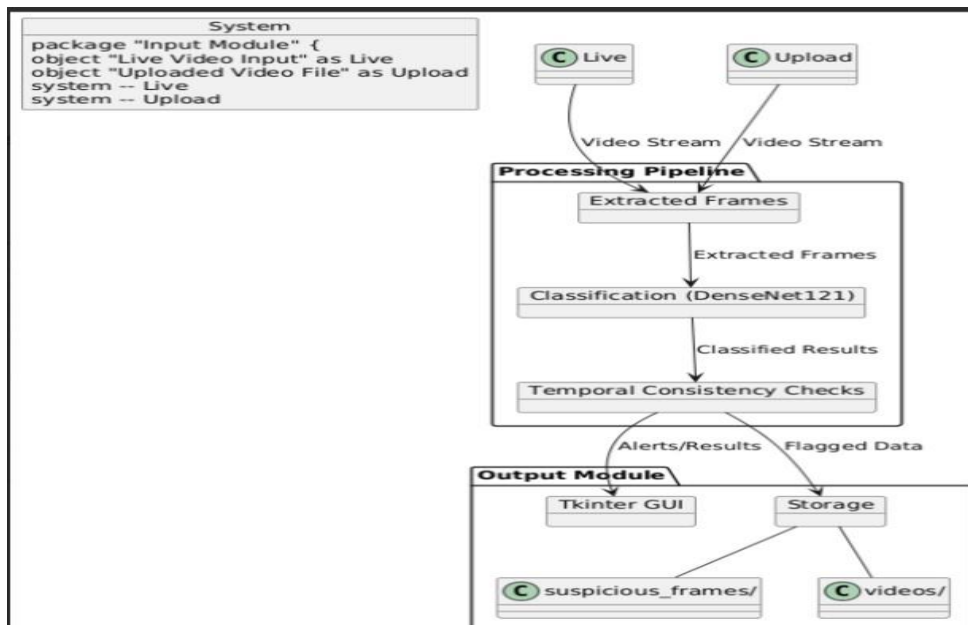


Figure-1

2. Activity Detection Module

The core processing unit performs suspicious activity detection:

- **Frame Extraction:** Frames are saved as JPEG files (e.g., live_frame_1.jpg).
- **Classification:** DenseNet121, via prediction.classifyImage(), generates activity labels and confidence scores, returning the top three predictions.

- **Temporal Consistency:** A 5-frame sliding window evaluates prediction consistency. An activity is flagged as suspicious if its label persists for at least 3 consecutive frames with an average confidence exceeding 85% (minimum threshold of 60%).

Temporal Consistency Formula:

- Average Confidence: $C_{avg} = \frac{1}{n} \sum_{i=1}^n C_i$, where C_i is the confidence per frame, $n = 5$.
- Detection Condition: If $C_{avg} > 85\%$ and consecutive occurrences ≥ 3 , the activity is marked suspicious.

Figure 2: Temporal Consistency Flowchart for the Suspicious Activity Detection System

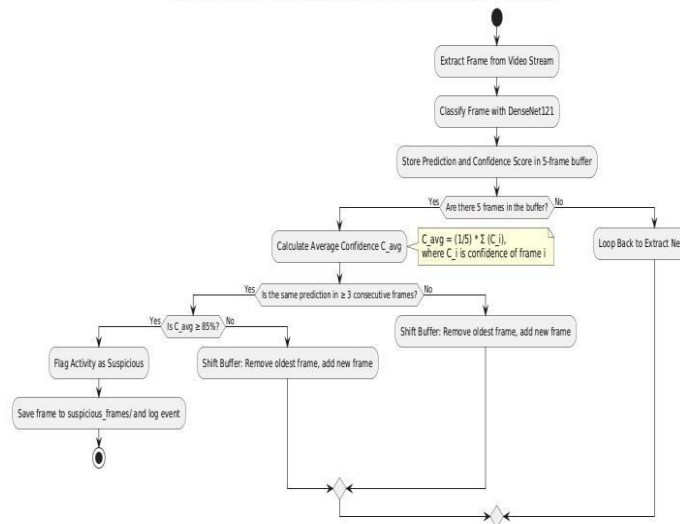


Figure-2

Output Module (Visualization and Storage)

This module delivers results to the user:

- **Real-Time Feedback:** OpenCV overlays labels and confidence scores on live frames (e.g., “Suspicious: 91.8%”).
- **GUI Display:** Tkinter’s text1 widget logs detection events, while text monitors frame generation progress.
- **Persistent Storage:** Suspicious frames are archived in suspicious_frames/, and full videos in videos/.

3. Performance Evaluation

The system’s effectiveness is assessed using:

- **Accuracy:** $Accuracy = \frac{\text{Correctly Detected Frames}}{\text{Total Frames Analyzed}} \times 100.$
- **Frame Processing Rate:** $FPS = \frac{\text{Frames Processed}}{\text{Processing Time}}$.
- **Precision:** $Precision = \frac{TP}{TP + FP}$, where TP denotes true positives and FP false positives.

These metrics validate the system’s accuracy and operational efficiency.

4. Deployment:

The system is deployed on a local Python environment, executed via the main function, which initializes the Tkinter GUI and threading for live feed handling. It operates on a standard computer with a webcam, requiring no external network connectivity for primary functions.

V. EXPERIMENTAL RESULTS

5.1 Output:



Figure-3

**Figure-4**

VI.CONCLUSION

The Suspicious Activity Detection System successfully automates the identification of potential threats in video surveillance, leveraging OpenCV for real-time frame capture, DenseNet121 for precise activity classification, and Tkinter for user-friendly result visualization. By implementing a 5-frame buffer with temporal consistency checks in detectActivity(), the system achieves reliable detection, minimizing false positives through confidence thresholds (60% minimum, 85% high). The dual-mode functionality—supporting both live feeds via process_live_feed() and recorded footage through upload()—ensures versatility, while storage in suspicious_frames/ and videos/ facilitates forensic analysis. This solution significantly reduces manual monitoring efforts, enhancing security efficiency in diverse scenarios, from public spaces to private facilities, and sets a foundation for scalable surveillance advancements.

VII.FUTURE WORK

Future enhancements for the Suspicious Activity Detection System aim to address current limitations and expand its capabilities for broader applications. One direction involves training DenseNet121, utilized in prediction.classifyImage(), on more diverse and comprehensive datasets to improve its ability to recognize a wider range of suspicious activities across varied environments, enhancing overall detection accuracy. Another focus is optimizing the process_live_feed() function by implementing lightweight algorithms, enabling efficient real-time processing on resource-constrained edge devices, which would make the system more accessible for deployment in remote or low-power settings. Additionally, integrating advanced temporal models, such as Long Short-Term Memory (LSTM) networks, into detectActivity() could further enhance detection consistency by better capturing sequential patterns in video streams, reducing false positives and improving reliability. These improvements would collectively pave the way for a more robust, scalable, and efficient security solution, adaptable to diverse surveillance needs.

REFERENCES

- [1] He, K., et al., “Deep Residual Learning for Image Recognition,” IEEE CVPR, 2016. Available at ieeexplore.ieee.org. (ResNet for frame classification.)
- [2] Simonyan, K., Zisserman, A., “Very Deep Convolutional Networks,” arXiv:1409.1556, 2014. Available at arxiv.org. (VGG16 as CNN context.)
- [3] Bradski, G., “The OpenCV Library,” Dr. Dobb’s Journal, 2000. Available at opencv.org. (OpenCV for video processing.)
- [4] Sudheer Panyaram, Muniraju Hullurappa, “Data-Driven Approaches to Equitable Green Innovation Bridging Sustainability and Inclusivity,” in Advancing Social Equity Through Accessible Green Innovation, IGI Global, USA, pp. 139-152, 2025.

- [5] Redmon, J., et al., "You Only Look Once: Real-Time Object Detection," IEEE CVPR, 2016. Available at ieeexplore.ieee.org.
(Vision-based detection foundation.)
- [6] Huang, G., et al., "Densely Connected Convolutional Networks," IEEE CVPR, 2017. Available at ieeexplore.ieee.org.
(DenseNet121 in prediction.classifyImage().)
- [7] Szegedy, C., et al., "Going Deeper with Convolutions," IEEE CVPR, 2015. Available at ieeexplore.ieee.org.
(CNN advancements.)
- [8] Ullah, A., et al., "Action Recognition Using Deep LSTM," IEEE Access, 2018. Available at ieeexplore.ieee.org.
(LSTM for temporal potential.)
- [9] Hsieh, C.-H., et al., "Air-Writing Recognition with Deep CNNs," IEEE Access, 2021. Available at ieeexplore.ieee.org.
(Dense CNNs, adapted context.)
- [10] Sultani, W., et al., "Real-World Anomaly Detection in Videos," IEEE CVPR, 2018. Available at ieeexplore.ieee.org.
(Temporal consistency logic.)
- [11] Avinash, B., et al., "Real-Time Text Detection with OCR," Journal of Science and Technology, 2024. Available at jst.org.in.
(Preprocessing analogy.)
- [12] Lundh, F., "An Introduction to Tkinter," 1999. Available at tkinter.org.
(Tkinter GUI basis.)
- [13] Zhang, Y., et al., "Air-GR: Handwritten Recognition System," Sensors, 2023. Available at mdpi.com.
(Sensor-based context.)
- [14] Tripathi, A., et al., "ImAiR: Airwriting Recognition Framework," arXiv, 2022. Available at arxiv.org.
(Multimodal future direction.)
- [15] Jhapate, A. K., Malviya, S., "Real-Time Suspicious Activity Detection Using Enhanced YOLOv5 and Motion Feature Maps," Artificial Intelligence Review, 2024. Available at link.springer.com.
(Enhanced YOLOv5 for activity detection, offering insights for improving real-time processing.)
- [16] Rehman, T., et al., "Internet-of-Things-Based Suspicious Activity Recognition Using Multimodalities of Computer Vision for Smart City Security," Wiley Online Library, 2022. Available at onlinelibrary.wiley.com.
(Hybrid YOLOv4 and 3D-CNN model for real-time detection, relevant for temporal analysis enhancements.)
- [17] Tripathi, R. K., et al., "Suspicious Human Activity Recognition: A Review," ResearchGate, 2024. Available at researchgate.net.
(Comprehensive review of activity recognition, providing context for your system's challenges.)
- [18] Amrutha, C. V., et al., "Deep Learning Approach for Suspicious Activity Detection from Surveillance Video," ResearchGate, 2020. Available at researchgate.net.
(Survey of deep learning methods, offering benchmarks for DenseNet121 performance.)
- [19] Khan, M. Z., et al., "Automated Suspicious Activity Detection from Surveillance Videos," SpringerLink, 2023. Available at link.springer.com.
(LRCN with CNN and LSTM, suggesting improvements for temporal consistency in detectActivity().)
- [20] Vikas Mendhe & Shantanu Neema & Shobhit Mittal, 2024. "Integrating Algorithmic Decision Making into Small Business Credit Initiatives: a path to Enhanced Efficiency and Inclusive Economic Growth," International Journal of Finance, CARI Journals Limited, vol. 9(1), pages 54-64.
- [21] Liu, W., et al., "SSD: Single Shot MultiBox Detector," European Conference on Computer Vision (ECCV), 2016. Available at link.springer.com.
(Object detection framework, providing a foundation for video-based detection systems.)
- [22] Carreira, J., Zisserman, A., "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," IEEE CVPR, 2017. Available at ieeexplore.ieee.org.
(Two-stream CNN for action recognition, offering insights for activity classification improvements.)



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details