International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Detection and Localization of Anomalies in Crowded Environments

Sumanth S N
Department of ISE
R V College of Engineering
Bengaluru, Karnataka-560059

Poornima Kulkarni
Asst. Professor
Department of ISE
R V College of Engineering
Bengaluru, Karnataka-560059

***Abstract*—This project introduces a lightweight, offline system for detecting anomalies in human movement within surveillance video. Implemented in Python, it leverages Farneback dense optical flow to extract motion features specifically magnitude and direction from consecutive frames. A Decision Tree Classifier, trained on labeled motion data, identifies abnormal behaviors within frame segments. Anomalies are visually highlighted in real-time using color-coded overlays, offering intuitive feedback. The system is optimized for resource-limited devices like laptops and Raspberry Pi, making it suitable for edge deployment. Its modular design supports easy model replacement or enhancement. Future improvements may include deep learning integration, better tracking, and more user-friendly interfaces.**

Index Terms—Anomaly Detection, Optical Flow, Surveillance Video Analysis, Motion Feature Extraction, Decision Tree Classifier, Real-Time Processing, Offline Monitoring, Farneback Algorithm, UCSD Ped1 Dataset, Foreground Segmentation.

I. INTRODUCTION

In recent years, the need for automated surveillance systems has grown significantly due to increasing security demands in public spaces such as transportation hubs, shopping centers, and urban environments. Traditional surveillance systems rely heavily on human monitoring, which is prone to fatigue and inefficiency, particularly in identifying subtle abnormal events in real-time. This challenge has paved the way for intelligent video analytics systems capable of detecting anomalies automatically without constant human oversight.

Anomaly detection in surveillance video involves identifying patterns of motion or behavior that deviate significantly from normal activity. These deviations may indicate events such as theft, intrusion, or violence, project proposes a lightweight, offline-capable anomaly detection system based on motion analysis and classical machine learning. It leverages Farneback dense optical flow to extract motion vectors from video frames and applies a Decision Tree Classifier trained on labeled motion features to detect abnormalities. Unlike deep learning models that require large datasets and computational resources, this system operates efficiently on standard hardware with minimal dependencies.

The system processes surveillance footage by breaking it into patches, extracting motion characteristics such as direction histograms and magnitude, and classifying each patch as normal or anomalous. Visual feedback is provided by overlaying colored indicators on video frames, allowing users to quickly identify abnormal regions. This patch-based approach ensures fine-grained localization of anomalies while maintaining real-time processing capabilities. Importantly, the design emphasizes standalone functionality, with no reliance on internet connectivity, cloud services, or external APIs. This makes it suitable for deployment in edge environments with limited resources and strict privacy requirements.

Overall, the proposed anomaly detection framework offers a practical and efficient solution for video surveillance, balancing performance, interpretability, and system simplicity. It serves as a foundational platform that can be further extended with advanced techniques such as deep learning, object detection, and real-time stream processing to meet evolving security and monitoring needs.

II. LITERATURE REVIEW

Memory-augmented autoencoders have been used to store normal patterns in a memory bank for input reconstruction, with anomalies producing higher reconstruction errors. This enables unsupervised detection with improved robustness and accuracy [1]. Similarly, self-supervised Transformers can learn normal frame orders and detect temporal inconsistencies, reducing reliance on labeled data and enhancing generalization [12]. Contrastive learning approaches also improve anomaly detection by distinguishing normal from pseudo-anomalous clips, strengthening feature boundaries and improving resilience to lighting or camera changes [13].

Modeling spatial-temporal patterns is another common strategy. Hybrid CNN-LSTM architectures learn sequences of normal activity over time to detect deviations, performing well in crowded scenes [4]. Dual-branch ConvLSTM systems combine frame reconstruction and future frame prediction, capturing subtle or evolving spatio-temporal anomalies [2]. Unsupervised contrastive learning can further enforce temporal consistency across frames, flagging anomalies without labels and adapting to scene changes [8]. Spatial-temporal regularization in the latent space also helps maintain consistency, while adaptive thresholding reduces false detections [15].

Graph-based approaches have been effective for human and crowd behavior modeling. Skeleton-based GNNs learn normal joint trajectories to detect abnormal human movements, even in cluttered or low-quality videos [5]. GCNs model crowd interactions by representing individuals and their relations as graph nodes and edges, enabling detection of social anomalies like isolation or erratic movement [7]. Motion-aware GANs can also model typical motion patterns and flag deviations in real time, emphasizing temporal coherence for rapid or sudden behaviors [14].

Transformers and attention-based models excel at capturing long-range temporal dependencies. By processing video sequences as a whole, they detect gradual or weak anomalies that CNN or LSTM-based models may miss [6]. Similarly, SlowFast networks adapted for anomaly detection use dual temporal streams to capture slow scene context and fast motion, making them suitable for rapid unusual events [9]. Dual-branch models that merge scene-level and object-level features also enhance detection in complex environments by combining global context with individual motion patterns [11].

Finally, data collection and multi-view approaches can improve performance in challenging scenarios. UAV-Crowd introduces a drone-based dataset and a multi-view detection framework, fusing features from different perspectives to handle occlusions and complex scenes effectively [10]. Two-stream networks that disentangle appearance and motion cues help differentiate harmless changes from behavioral anomalies, reducing false alarms in structured video environments [3].

III. PROBLEM STATEMENT

The primary problem addressed in this project is the automated detection of anomalous motion patterns in surveillance video footage, particularly in densely populated or public environments such as pedestrian walkways or campus corridorsFurthermore, many existing automated systems require complex infrastructure such as internet connectivity, cloud-based storage, and high-end hardware to function effectively constraints that are not suitable for resource-limited or sensitive environments. The core challenge lies in developing a lightweight, self-contained solution that can accurately identify unusual activities or behaviors, such as erratic movement or abnormal object presence, without relying on deep learning models or external dependencies. The proposed solution aims to overcome this challenge by utilizing classical computer vision techniques, specifically Farneback optical flow for motion estimation and a Decision Tree Classifier for anomaly prediction.

IV. METHODOLOGY

The anomaly detection system follows a sequence of phases to identify unusual motion patterns in surveillance footage using the UCSD dataset. The pipeline includes data ingestion, preprocessing, feature extraction, classification, and visualization.

A. Data Ingestion Phase

The system loads grayscale .tif frames and corresponding ground truth masks from the UCSD dataset, ensuring proper temporal ordering. This ensures synchronization between video data and annotations for accurate frame-wise evaluation.

B. Preprocessing Phase

Optical flow is computed using Farneback's method to capture motion vectors, while foreground regions are extracted using OpenCV's BackgroundSubtractorMOG. Motion data is refined through direction binning (8 directions plus static) and magnitude thresholding to filter noise and retain meaningful motion information.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

C. Feature Extraction Phase
Frames are divided into small patches (e.g., 10×10), and features are extracted for each patch: histogram of motion directions, number of foreground pixels, average motion magnitude, and spatial coordinates. These features form a vector representing patch behavior for classification.

D. Classification Phase
A pre-trained Decision Tree Classifier labels each patch as normal or anomalous using the extracted features. Anomalous patches are flagged for visualization. The decision tree provides interpretable, fast predictions suitable for real-time processing.

E. Visualization and Evaluation Phase
Anomalous regions are highlighted with bounding boxes, and predictions are compared with ground truth. Performance is measured using precision, recall, F1-score, and Equal Error Rate (EER) at both frame and pixel levels, providing a comprehensive evaluation of detection accuracy.

V. IMPLEMENTATION

The anomaly detection system was implemented in Python using standard scientific libraries with a modular design for clarity, scalability, and offline execution. It processes surveillance footage from the UCSD dataset, computes motion vectors, extracts spatial-temporal features, classifies behavior with a Decision Tree, and visualizes anomalies in real time. The codebase is efficient, enabling execution on low-power devices like a Raspberry Pi. Each step is implemented as a dedicated module within a streamlined pipeline.

A. Data Loading Implementation
Grayscale .tif frames and corresponding ground truth masks are loaded using OpenCV. A Python script traverses dataset folders, sorts frame filenames, and stores them sequentially in memory, ensuring structured access for frame-wise analysis.

B. Preprocessing Implementation
Optical flow measure is a technique used to compute via cv2.calcOpticalFlowFarneback to obtain pixel-level motion vectors, which are converted to magnitude and angle representations. Foreground motion is isolated using cv2.createBackgroundSubtractorMOG(). Angles are quantized into nine bins (eight directions plus static), and a motion threshold filters out insignificant movements for further analysis.

C. Feature Extraction Implementation
Frames are divided into patches (e.g., 10×10 pixels), and features are extracted from patches with significant foreground activity. Features include motion direction histograms, number of active pixels, average motion magnitude, and patch coordinates. These are combined into feature vectors, collected in batches, and passed to the classifier for real-time decisions.

D. Classification Implementation
The pre-trained Decision Tree Classifier (from Scikit-learn) is loaded via joblib. Each patch-level feature vector is classified as normal or anomalous. Anomalous patches are recorded with their coordinates for visualization. The classifier can also be retrained using the modular training script.

E. Visualization and Evaluation Implementation
Anomalous patches are highlighted with bounding boxes using cv2.rectangle(), and the frame with overlays is displayed in an OpenCV window. Metrics such as precision, recall, F1-score, and Equal Error Rate (EER) are calculated by comparing predictions with ground truth and can be printed or plotted using matplotlib, providing both visual and quantitative feedback.

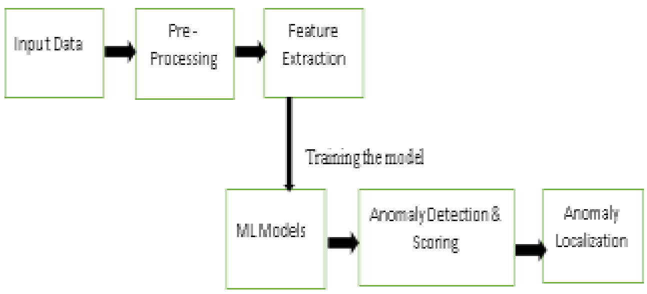

Fig 1: Anomaly Detection System Pipeline Overview of the process

VI. RESULTS & DISCUSSION

The proposed anomaly detection system provides an efficient solution for identifying unusual crowd activity in surveillance videos, tested on the UCSD dataset. By utilizing simple yet informative motion features, such as optical flow direction and magnitude, the system avoids reliance on deep learning or high-performance hardware, making it lightweight and suitable for edge devices. The Decision Tree classifier enables fast, interpretable predictions, ideal for real-time monitoring. A major advantage of this system is its fully offline operation. All stages—from video input and preprocessing to feature extraction and classification—are handled locally, requiring no internet connectivity or external servers. This makes it suitable for high-security or low-connectivity

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

scenarios, such as remote surveillance or military applications. The modular design also allows easy adaptation and extension, supporting customization for different datasets and deployment strategies requirements.

Fig 1: A frame showing skater skating on the road

Fig 2: A frame showing cyclist moving among pedestrian

Metrics	Values
Frame Precision	0.8671
Frame Recall	0.1217
Frame F1 Score	0.2134
True Positives	414
False Positives	258
Pixel Precision	0.6161
Pixel Recall	0.5679
Pixel F1 Score	0.5910

Table 1: Performance Metrics

Results of frame & pixel level metrics

Despite its simplicity, the optical flow-based motion representation captures rich temporal information across frames. Background subtraction refines the focus on meaningful movement, filtering out static regions and irrelevant noise. Converting motion directions into histogram bins provides a compact yet descriptive feature vector, enabling effective classification even with limited input. However, the system has limitations. The classical Decision Tree classifier, while fast, may underperform in complex scenarios with subtle anomalies or overlapping objects. Fixed patch sizes may also introduce spatial rigidity, potentially missing detections in crowded or highly dynamic scenes. Future improvements could include adaptive patching or more advanced classifiers such as Random Forests or lightweight CNNs.

The visualization component is essential for real-time monitoring, with bounding boxes over anomalous regions providing immediate feedback and supporting manual verification. Standard evaluation metrics—including precision, recall, F1-score, and EER—ensure measurable and comparable results across configurations and datasets. Overall, the system strikes a balance between computational efficiency and detection accuracy. It demonstrates that classical computer vision techniques, when applied intelligently, can provide practical solutions for anomaly detection in constrained environments. Its effectiveness on the UCSD dataset highlights the robustness of the pipeline and its potential for broader deployment in real-world surveillance scenarios.

VII. CONCLUSION

The anomaly detection system developed in this project successfully identifies and localizes abnormal behavior in surveillance videos. By integrating classical computer vision techniques—optical flow estimation, background subtraction, and patch-based feature extraction—with a Decision Tree classifier, the system achieves real-time detection while maintaining computational efficiency. Its fully offline, self-contained design allows operation on low-power hardware without internet access, cloud services, or external APIs, making it suitable for resource-constrained or security-sensitive environments. The visual overlay of detected anomalies enhances interpretability, providing a clear interface for monitoring unusual crowd activity.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

While the current system is lightweight and interpretable, several enhancements could improve accuracy and adaptability. Future work may explore deep learning models, such as autoencoders or CNN-LSTM architectures, to capture complex motion patterns and rare behaviors. Human-specific object detection using models like YOLO could increase precision in crowded or noisy scenes. Performance can be further improved through real-time video streaming, multithreaded processing, and GPU acceleration. Additionally, anomaly severity scoring and GUI-based dashboards with live statistics would enhance usability and allow for more granular analysis. These improvements could transform the system into a more robust, intelligent, and scalable anomaly detection framework.

REFERENCES

[1] Mehran, R., Oyama, A., & Shah, M. "Abnormal crowd behavior detection using social force model." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[2] Mahadevan, V., Li, W., Bhalodia, V., & Vasconcelos, N. "Anomaly detection in crowded scenes." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[3] Lu, C., Shi, J., & Jia, J. "Abnormal event detection at 150 FPS in MATLAB." IEEE International Conference on Computer Vision (ICCV), 2013.

[4] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., & Davis, L. S. "Learning temporal regularity in video sequences." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] Ionescu, R. T., Smeureanu, S., Alexe, B., & Popescu, M. "Detecting abnormal events in video using narrowed normality clusters." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[6] Liu, W., Luo, W., Lian, D., & Gao, S. "Future frame prediction for anomaly detection–a new baseline." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[7] Sultani, W., Chen, C., & Shah, M. "Real-world anomaly detection in surveillance videos." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] Xu, D., Ricci, E., Yan, Y., Song, J., & Sebe, N. "Learning deep representations of appearance and motion for anomalous event detection." British Machine Vision Conference (BMVC), 2015.

[9] Chong, Y. S., & Tay, Y. H. "Abnormal event detection in videos using spatiotemporal autoencoder." International Symposium on Neural Networks (ISNN), 2017.

[10] Sabokrou, M., Khalooei, M., Fathy, M., & Adeli, E. "Adversarially learned one-class classifier for novelty detection." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[11] Tang, Y., Wu, Y., & Chen, Y. "Integrating optical flow and convolutional neural networks for real-time anomaly detection." Multimedia Tools and Applications, 2020.

[12] Sun, Y., Yuan, Y., & Wang, Q. "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications." International Conference on World Wide Web (WWW), 2018.

[13] Ravanbakhsh, M., Nabi, M., Mousavi, H. S., Frosio, I., & Kautz, J. "Training adversarial discriminators for cross-channel abnormal event detection in crowds." IEEE Winter Conference on Applications of Computer Vision (WACV), 2017.

[14] Ionescu, R. T., Smeureanu, S., & Alexe, B. "Unmasking the abnormal events in video." IEEE International Conference on Computer Vision (ICCV), 2017.

[15] Chang, Y., Lan, W., & Hsu, W. H. "Clustering-based feature learning on abnormal crowd behavior detection." ACM International Conference on Multimedia, 2017.

[16] Xu, M., Guo, X., Yu, X., & Wu, J. "Abnormal event detection in crowded scenes using sparse coding." Multimedia Tools and Applications, 2018.

[17] Chong, Y. S., & Tay, Y. H. "Robust anomaly detection in videos with self-supervised prediction." Asian Conference on Computer Vision (ACCV), 2020.

[18] Wu, B., & Nevatia, R. "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors." International Journal of Computer Vision, 2007.

[19] Kratz, L., & Nishino, K. "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[20] Adam, A., Rivlin, E., Shimshoni, I., & Reinitz, D. "Robust real-time unusual event detection using multiple fixed-location monitors." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2008.