ISSN 2723-6471

Edge-Deployable Multi-Class Surveillance Video Anomaly Classification Using a MobileNetV2–LSTM Spatial–Temporal Framework

S Aasha Nandhini[1,*], R Karthick Manoj [2], Vasantakumaren S R[3]

[1]*Department of Electronics and Communication Engineering, SSN School of Engineering, Shiv Nadar University, Chennai, India.*

[2]*Department of EEE, AMET University, Chennai, India*

[3]*Faculty of Data Science and IT, INTI International University, 71800 Nilai, Malaysia*

(Received: March 1, 2026; Revised: May 5, 2026; Accepted: July 25, 2026; Available online: August 18, 2026)

Abstract

Automated surveillance systems require models that can recognize abnormal events accurately while maintaining efficient inference on resource-constrained hardware. This study proposes a lightweight spatial–temporal classification framework that integrates MobileNetV2 and Long Short-Term Memory networks for identifying five surveillance activity classes: Normal, Fighting, Burglary, Road Accident, and Explosion. Surveillance videos were preprocessed through frame sampling, resizing, normalization, and fixed-length sequence construction. MobileNetV2 was employed to extract compact frame-level spatial features, while the LSTM modeled temporal dependencies across consecutive frames. The framework was evaluated using accuracy, class-wise precision, recall, F1-score, macro F1-score, weighted F1-score, confusion-matrix analysis, and probability-based anomaly scoring. Experimental results on 463 video sequences show that the proposed model achieved an accuracy of 97.41%, a macro F1-score of 0.9612, and a weighted F1-score of 0.9744. Most classification errors occurred between visually similar human-centered anomaly classes, particularly Fighting and Burglary, while Normal and Explosion sequences were recognized with very high reliability. The trained model was converted into TensorFlow Lite format and deployed on a Raspberry Pi 4, achieving approximately 6–8 frames per second with an average latency of 120–150 ms per frame. These findings indicate that the proposed MobileNetV2–LSTM framework provides an effective balance between multi-class event recognition, temporal representation learning, and near-real-time edge inference. However, further validation on larger and more diverse surveillance datasets is required to confirm its generalizability under unconstrained operational conditions.

Keywords: Video Anomaly Classification, MobileNetV2, Long Short-Term Memory, Edge Computing, Intelligent Surveillance, TensorFlow Lite, Spatial–Temporal Learning, Process Inovation

1. Introduction

The widespread adoption of closed-circuit television and networked cameras has significantly increased the volume of surveillance video generated in public spaces, transportation systems, campuses, commercial facilities, and urban infrastructure. Although these systems provide continuous visual coverage, their effectiveness remains highly dependent on the ability of human operators to observe multiple video streams simultaneously. Prolonged monitoring can lead to fatigue, delayed responses, and overlooked incidents, particularly when abnormal events occur only briefly within long periods of routine activity. Automated video anomaly recognition has therefore become an important research direction for identifying potentially dangerous events and supporting timely operational responses [1], [2], [3].

Early surveillance-video analysis methods commonly relied on handcrafted motion descriptors, trajectory information, background modeling, and predefined behavioral rules. However, these techniques often exhibited limited robustness when applied to environments involving illumination changes, camera movement, occlusion, crowded scenes, and variations in human activity. Deep learning has subsequently enabled models to learn hierarchical visual representations directly from video data, reducing dependence on manually designed features. Convolutional neural networks, autoencoders, and fully convolutional architectures have demonstrated improved capability in recognizing

***Corresponding author: S Aasha Nandhini (aashanandhinis@ssn.edu.in)**
DOI: https://doi.org/10.47738/jads.v7i3.1471

irregular visual patterns and separating normal activities from abnormal observations [4], [5]. Nevertheless, methods that process individual frames or primarily reconstruct visual appearances may not adequately represent how an event develops over time.

Temporal information is essential because many surveillance events cannot be reliably interpreted from a single image. Activities such as fighting, burglary, road accidents, and explosions are characterized not only by their visual appearance but also by motion intensity, interaction patterns, and sequential changes across consecutive frames. Recent studies have therefore explored temporal attention mechanisms, multi-timescale feature learning, weakly supervised recognition, and representation-learning frameworks capable of capturing longer-range event dependencies [2], [6], [7], [8], [9]. These developments have improved anomaly recognition performance, but many existing approaches remain designed primarily for binary normal-versus-abnormal detection or anomaly-score estimation rather than explicit identification of the event category.

The distinction between anomaly detection and anomaly classification is operationally important. A generic abnormality alert may notify an operator that an unusual event has occurred, but it does not indicate the type of response required. For example, fighting may require security intervention, a road accident may require medical assistance, and an explosion may require immediate evacuation and emergency coordination. Reconstruction-based methods, including convolutional autoencoders, typically define anomalies as deviations from learned normal patterns and therefore do not inherently produce class-specific semantic labels [6], [10]. A supervised multi-class framework can instead learn discriminative representations for both normal activity and several predefined anomaly categories, producing information that is more directly actionable.

Another unresolved challenge concerns computational feasibility. High-capacity three-dimensional convolutional networks, attention-based architectures, and large representation-learning models can provide strong spatial–temporal modeling, but their memory requirements and inference costs may restrict deployment on low-power surveillance hardware [7], [8], [9]. Transmitting continuous video to cloud servers may also introduce communication latency, bandwidth consumption, and privacy concerns. Edge-based inference addresses these limitations by processing surveillance data close to the camera. However, an edge-deployable model must balance classification performance with model size, inference latency, memory consumption, and processing throughput. MobileNetV2 offers an efficient spatial feature extractor through depthwise separable convolution and inverted residual blocks, while Long Short-Term Memory networks provide a practical mechanism for modeling temporal dependencies across frame-level representations.

Accordingly, this study proposes a lightweight spatial–temporal framework that integrates MobileNetV2 and LSTM for supervised classification of surveillance-video sequences into five categories: normal activity, fighting, burglary, road accident, and explosion. MobileNetV2 extracts compact visual features from individual frames, whereas the LSTM models their temporal evolution before multi-class prediction. The trained model is subsequently converted into TensorFlow Lite format for deployment on a Raspberry Pi platform. The study evaluates the framework using overall and class-wise classification metrics, confusion-matrix analysis, probability-based anomaly scoring, inference latency, and processing throughput. Through this design, the study addresses the need for an anomaly-classification system that combines semantic event identification, temporal modeling, and practical execution under constrained computational resources.

2. Literature Review

Video anomaly analysis has evolved from conventional feature-engineering approaches toward representation-learning models capable of extracting complex visual patterns directly from surveillance footage. Earlier methods commonly employed optical flow, object trajectories, motion histograms, background subtraction, and manually defined descriptors to identify deviations from normal activities. Although these approaches were computationally manageable, their performance was often sensitive to illumination changes, camera perspectives, occlusion, crowd density, and variations in event duration [1], [4], [11]. Reconstruction-based deep learning subsequently became a prominent approach, in which autoencoders, convolutional autoencoders, and generative models were trained to reconstruct normal observations and identify anomalous sequences through elevated reconstruction errors [6], [10], [12]. However,

reconstruction quality does not always correspond directly to anomaly separability because sufficiently expressive models may also reconstruct abnormal observations with relatively low error [13].

To improve spatial representation, convolutional neural networks have been employed to learn discriminative features from individual frames or short video segments. Three-dimensional CNNs extend conventional spatial convolution by jointly processing temporal dimensions, thereby capturing appearance and motion within a unified architecture [14], [15]. Other studies have incorporated multi-scale feature extraction, residual networks, vision transformers, and pretrained visual-language representations to improve recognition under complex surveillance conditions [7]–[9], [16]. Despite their strong representational capacity, these architectures frequently require substantial memory, high computational throughput, and hardware acceleration. Their practical deployment on embedded surveillance platforms is consequently limited, particularly when continuous inference must be performed locally with restricted processing and energy resources [17].

Hybrid CNN–recurrent architectures provide an alternative strategy by separating spatial and temporal representation learning. Within these models, a CNN extracts frame-level feature vectors, while recurrent units such as recurrent neural networks, gated recurrent units, or LSTM networks model sequential dependencies among the extracted representations [18], [19]. LSTM is particularly suitable for surveillance-video analysis because its gating mechanism can retain relevant temporal information while reducing the vanishing-gradient problem associated with conventional recurrent networks. Previous CNN–LSTM frameworks have demonstrated promising performance in violence recognition, accident detection, suspicious-activity identification, and general action classification [2], [20], [21]. Nevertheless, many reported models employ computationally intensive CNN backbones or evaluate performance exclusively in conventional computing environments, without systematically considering deployment latency, throughput, memory consumption, or model conversion for edge hardware.

Recent research has therefore increasingly emphasized lightweight architectures and edge-oriented model optimization. MobileNet-based networks reduce computational cost through depthwise separable convolutions, while model quantization, mixed-precision computation, pruning, and TensorFlow Lite conversion can further reduce storage and inference requirements [22], [23]. Existing edge-surveillance studies indicate that lightweight models can support responsive local inference, but many remain restricted to binary anomaly detection, single-event recognition, or frame-level classification without explicit temporal modeling [5], [17], [24]. Consequently, there remains a need for a unified framework that combines an efficient spatial backbone, sequence-level temporal learning, multi-class semantic prediction, and deployment-oriented evaluation. The proposed MobileNetV2–LSTM framework addresses this requirement by classifying normal activity and four anomaly categories while evaluating both predictive performance and operational feasibility on a resource-constrained edge platform.

3. Methodology

This study develops a supervised spatial–temporal classification framework for recognizing five surveillance-video activity classes: Normal, Fighting, Burglary, Road Accident, and Explosion. The overall workflow consists of four sequential stages: video preparation, spatial feature extraction, temporal representation learning, and edge-device deployment. Unlike reconstruction-based anomaly detectors that estimate abnormality only from reconstruction error, the proposed framework directly learns class-specific decision boundaries from labeled video sequences [6]. The complete experimental workflow should be presented through a revised version of the existing figure 1, as specified below.

 ISSN 2723-6471

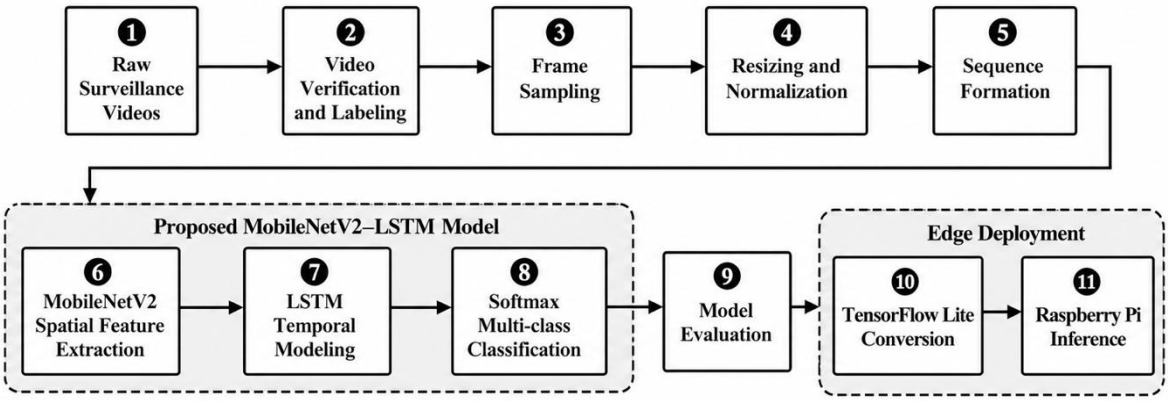

Figure 1. Overall workflow of the proposed MobileNetV2–LSTM surveillance-video classification framework

3.1. Dataset Preparation and Video Sequence Construction

The experimental dataset consists of 100 surveillance videos collected from public surveillance-video repositories and publicly available annotated video sources. Each video was assigned to one of five classes: Normal, Fighting, Burglary, Road Accident, or Explosion. Existing event annotations were retained when available, whereas videos collected from public online sources were manually reviewed and labeled. Clips with unclear activities, overlapping event categories, severe corruption, or insufficient visual evidence were excluded to reduce labeling ambiguity. The resulting videos were subsequently segmented into fixed-length temporal sequences, producing 463 evaluable video sequences.

To preserve class proportions during model development, the resulting sequences were divided using a stratified 80:20 training–testing split. Stratification was applied at the class level so that every activity category remained represented in both partitions. To prevent information leakage, segments originating from the same source video should remain within only one partition. This video-level separation is essential because randomly distributing highly similar segments from a single video across the training and testing sets could produce overly optimistic performance estimates.

Each video was decoded into a sequence of frames using OpenCV. Frames were sampled at a fixed temporal interval and resized to 64×64 pixels to reduce computational and memory requirements. Although higher input resolutions may retain more fine-grained visual information, the selected resolution provides a practical balance between representational quality and edge-device feasibility. Each pixel value was normalized into the interval [0,1] as follows:

$$\widetilde{x_t} = \frac{x_t}{255}, \tag{1}$$

x_t denotes the original RGB frame at time step t, and $\widetilde{x_t}$ denotes its normalized representation. The normalized frames were arranged into sequences of $T = 30$ consecutive frames. An input sequence is therefore represented as

$$X_i = \{\widetilde{x_{i,1}}, \widetilde{x_{i,2}}, \dots, \widetilde{x_{i,T}}\}, \tag{2}$$

$X_i \in R^{T\times H\times W\times C}$, $H = W = 64$, and $C = 3$. Each sequence inherits the activity label of its corresponding temporal segment. Fixed-length sequence construction ensures that the model receives inputs with consistent temporal dimensions while preserving short-term activity progression. Table 1 summarizes the dataset-processing and system configuration used during model development and deployment. Model training and preprocessing were performed using Google Colab or a GPU-enabled computing system equipped with a multi-core CPU, at least 8 GB of RAM, and an NVIDIA GPU to accelerate the training process. For deployment and real-time inference, the proposed system was implemented on a Raspberry Pi 4 Model B with 4 GB of RAM. Video input can be obtained using either a Pi Camera or a USB webcam, while the Raspberry Pi is powered by a 5V 3A USB-C adapter. This configuration provides a lightweight edge-computing platform suitable for real-time video anomaly detection.

Table 1. Dataset and preprocessing configuration

Component	Specification
Development Machine	Google Colab / GPU-enabled system
Processor	Multi-core CPU
Memory	Minimum 8 GB RAM

Graphics	NVIDIA GPU (for faster training)
Deployment Device	Raspberry Pi 4 Model B
RAM	4GB
Camera Module	Pi Camera or USB webcam
Power Supply	5V 3A USB-C adapter

The original dataset explanation may be retained as the factual basis, but it should be replaced by the more structured description above. The manuscript must also report the exact number of source videos and generated sequences per class when these counts are available.

3.2. MobileNetV2–LSTM Spatial–Temporal Classification

The proposed architecture separates video representation learning into spatial and temporal stages. MobileNetV2 is employed as the spatial backbone because its depthwise separable convolutions and inverted residual blocks provide a lower computational burden than conventional high-capacity CNN architectures [22]. Meanwhile, the LSTM layer models temporal relationships among consecutive frame-level features and retains activity information across the observed sequence [18]. For every normalized frame $\widetilde{x_{i,t}}$, the classification head of MobileNetV2 is removed, and the remaining backbone produces a compact feature vector:

$$f_{i,t} = \phi\left(\widetilde{x_{i,t}}; \theta_M\right), \tag{3}$$

$\phi(\cdot)$ denotes the MobileNetV2 feature extractor, θ_M represents its parameters, and $f_{i,t} \in R^d$ is the spatial feature vector for frame t. The feature vectors extracted from all frames are concatenated chronologically:

$$F_i = \left[f_{i,1}, f_{i,2}, \ldots, f_{i,T}\right] \in R^{T \times d} \tag{4}$$

The ordered feature sequence is then passed to the LSTM. At each time step, the LSTM updates its input, forget, output, cell, and hidden states according to

$$\begin{aligned} i_t &= \sigma\left(W_{if_t} + U_{ih_{t-1}} + b_i\right), \\ g_t &= \sigma\left(W_{gf_t} + U_{gh_{t-1}} + b_g\right), \\ o_t &= \sigma\left(W_{of_t} + U_{oh_{t-1}} + b_o\right), \\ \widetilde{c_t} &= \tanh\left(W_{cf_t} + U_{ch_{t-1}} + b_c\right), \\ c_t &= g_t \odot c_{t-1} + i_t \odot \widetilde{c_t}, \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{5}$$

i_t, g_t, and o_t denote the input, forget, and output gates; c_t is the cell state; h_t is the hidden state; σ represents the sigmoid function; and $\odot$ denotes element-wise multiplication. The final hidden representation h_T summarizes the spatial–temporal information contained within the sequence. It is passed to a fully connected output layer followed by softmax activation:

$$\begin{aligned} z_i &= W_{yh_{i,T}} + b_y, \\ p_{i,c} &= \frac{}{\exp\left(z_{i,c}\right)\sum_{j=1}^{C}\exp\left(z_{i,j}\right), \qquad C = 5, \end{aligned} \tag{6}$$

$p_{i,c}$ is the probability that sequence i belongs to class c. The final predicted label is

$$\hat{y}_i = \arg\max_{c \in \{1,\ldots,C\}} p_{i,c}. \tag{7}$$

The model is optimized using categorical cross-entropy:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(p_{i,c}\right), \tag{8}$$

 ISSN 2723-6471

N is the number of training sequences, $y_{i,c}$ is the one-hot encoded ground-truth label, and $p_{i,c}$ is the predicted probability. Training was performed using the Adam optimizer for 20 epochs with a batch size of 2. A cosine-decay learning-rate scheduler was employed to gradually reduce the learning rate during optimization. Mixed-precision training using the mixed_float16 policy was applied to reduce GPU memory consumption and accelerate training. However, mixed precision should be described as a training optimization, while edge inference optimization should be attributed to TensorFlow Lite conversion and any quantization procedure actually applied [25].

Figure 2 illustrates the proposed MobileNetV2–LSTM architecture for five-class video anomaly classification. Each input sample consists of a sequence of 30 RGB video frames, with each frame resized to (64 x 64) pixels, resulting in an input tensor of (30 x 64 x 64 x 3). The frames are processed independently but with shared weights through the time-distributed MobileNetV2 backbone, which extracts spatial features from each frame. The resulting frame-level features are arranged into a feature matrix of size (30 x d), where (d) denotes the dimensionality of the extracted feature vector. These sequential features are then passed to the LSTM layer to model temporal dependencies across the 30 frames. The final hidden representation produced by the LSTM is subsequently processed by a dense layer and a five-class softmax output layer. The five output classes are Normal, Fighting, Robbery, Road Accident, and Explosion.

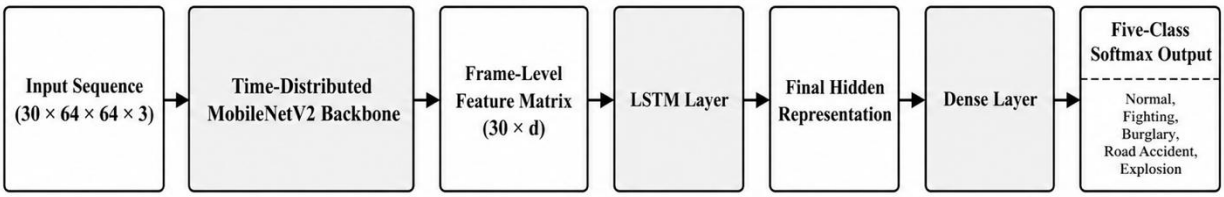

Figure 2. Detailed architecture of the MobileNetV2–LSTM model

4. Results and Discussion

The proposed MobileNetV2–LSTM framework was evaluated on 463 surveillance-video sequences distributed across five activity classes: Normal, Fighting, Burglary, Road Accident, and Explosion. The evaluation examined overall predictive performance, class-specific behavior, temporal anomaly scoring, and computational feasibility on a Raspberry Pi edge device. The reported results demonstrate strong sequence-level classification performance, although the findings should be interpreted in relation to the relatively limited and curated dataset.

4.1. Overall Classification Performance

The proposed MobileNetV2–LSTM model correctly classified 451 out of 463 test sequences, resulting in an overall accuracy of 97.41% and an error rate of 2.59%. The macro F1-score of 0.9612 indicates that the model maintained strong and relatively balanced performance across the five activity classes, while the weighted F1-score of 0.9744 reflects the overall classification performance while accounting for the different numbers of samples in each class. The overall classification performance of the proposed model is presented in table 2.

Table 2. Overall classification performance of the proposed model

Metric	Value
Test sequences	463
Correct predictions	451
Misclassified sequences	12
Accuracy	97.41%
Error rate	2.59%
Macro F1-score	9.612
Weighted F1-score	9.744

As shown in table 2, 451 of the 463 evaluated sequences were correctly classified, while only 12 sequences were misclassified. The resulting accuracy of 97.41% demonstrates that the model can reliably classify the five target activities. The relatively small difference between the macro F1-score (0.9612) and weighted F1-score (0.9744) further indicates that the high overall performance was not solely caused by the classes with larger support. The detailed performance for each activity class is presented in table 3.

ISSN 2723-6471

Table 3. Class-wise performance of the MobileNetV2–LSTM model

Class	Support	Precision	Recall	F1-score
Burglary	50	0.8571	0.9600	0.9057
Explosion	54	0.9643	1.0000	0.9818
Fighting	134	0.9846	0.9552	0.9697
Normal	184	1.0000	1.0000	1.0000
Road Accident	41	1.0000	0.9024	0.9487

As shown in table 3, the model achieved its best performance on the Normal class, with perfect precision, recall, and F1-score of 1.0000. The Explosion class also achieved excellent performance, with a recall of 1.0000 and an F1-score of 0.9818. Fighting obtained an F1-score of 0.9697, while Road Accident achieved an F1-score of 0.9487. Burglary obtained the lowest F1-score at 0.9057, mainly because some sequences from other abnormal classes were incorrectly predicted as Burglary. The distribution of correct and incorrect predictions across the five classes is visualized in figure 3.

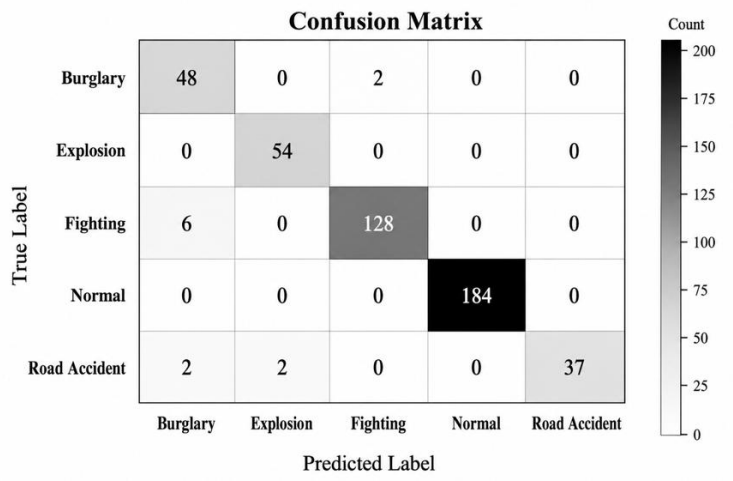

Figure 3. Confusion matrix of the proposed MobileNetV2–LSTM model

As shown in figure 3, the diagonal elements represent correctly classified sequences, whereas the off-diagonal elements represent misclassifications. The model correctly classified 48 of 50 Burglary sequences, 54 of 54 Explosion sequences, 128 of 134 Fighting sequences, 184 of 184 Normal sequences, and 37 of 41 Road Accident sequences. The numerical values corresponding to the confusion matrix in figure 3 are provided in table 4.

Table 4. Numerical representation of the confusion matrix

True class	Predicted Burglary	Predicted Explosion	Predicted Fighting	Predicted Normal	Predicted Road Accident
Burglary	48	0	2	0	0
Explosion	0	54	0	0	0
Fighting	6	0	128	0	0
Normal	0	0	0	184	0
Road Accident	2	2	0	0	37

As shown in table 4, the diagonal contains 451 correct predictions, corresponding to the reported accuracy of 97.41%, while the 12 off-diagonal values represent the classification errors. The largest individual error occurs for the Fighting class, where six sequences are incorrectly classified as Burglary. For Road Accident, four sequences are misclassified, with two predicted as Burglary and two as Explosion. No Normal sequence is misclassified, confirming the perfect recall of the Normal class.

 ISSN 2723-6471

Overall, the results demonstrate that the proposed MobileNetV2–LSTM model effectively distinguishes normal scenes from multiple types of abnormal activities while also providing multiclass anomaly recognition. This capability enables the system to identify specific anomaly categories—Burglary, Explosion, Fighting, and Road Accident—rather than producing only a binary normal–abnormal decision, which can support more appropriate responses in practical surveillance applications.

4.2. Error Patterns and Temporal Anomaly Behavior

The remaining classification errors were concentrated among activity classes with partially overlapping visual characteristics. The detailed distribution of these errors and their possible causes is presented in table 5.

Table 5. Concentration of off-diagonal classification errors

True class	Predicted class	Error count	Possible explanation
Fighting	Burglary	6	Rapid human movement and occlusion may resemble forced entry or physical struggle
Burglary	Fighting	2	Human-centered motion may overlap with aggressive interaction patterns
Road Accident	Burglary	2	Sudden scene movement may resemble suspicious human or object activity
Road Accident	Explosion	2	Impact, smoke, or visual disturbance may resemble explosion-related events

As shown in table 5, the largest source of error was the confusion between Fighting and Burglary, with six Fighting sequences incorrectly classified as Burglary and two Burglary sequences incorrectly classified as Fighting. This reciprocal confusion suggests that the model occasionally has difficulty distinguishing human-centered abnormal activities involving rapid movement, close interaction, or partial occlusion. Road Accident sequences accounted for four additional errors, with two classified as Burglary and two as Explosion. These errors may occur when vehicle movement, impact, smoke, or abrupt changes in the visual scene produce patterns that overlap with other anomaly categories. The error distribution indicates that the primary limitation of the proposed model is not the discrimination between normal and abnormal activity, but rather the fine-grained separation of visually similar anomaly subclasses. This observation is consistent with the class-wise performance shown in figure 4, where the precision, recall, and F1-score of each activity class can be compared simultaneously.

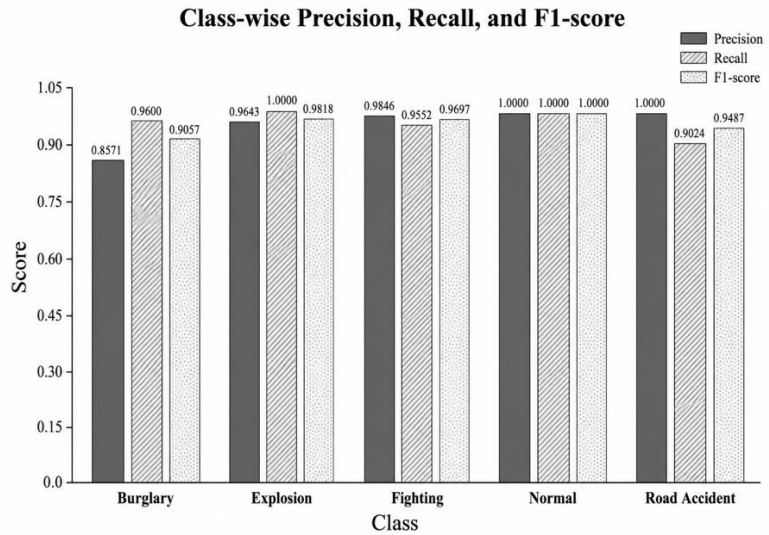

Figure 4. Class-wise precision, recall, and F1-score

As shown in figure 4, the model achieved strong performance across all five classes. The Normal class obtained perfect precision, recall, and F1-score, while Explosion also achieved a high F1-score of 0.9818. Fighting achieved an F1-score of 0.9697, followed by Road Accident at 0.9487 and Burglary at 0.9057. The lower precision of Burglary (0.8571), compared with its recall (0.9600), indicates that several sequences belonging to other classes were incorrectly

 ISSN 2723-6471

assigned to Burglary. This observation is consistent with the confusion matrix presented previously. Beyond the final class prediction, the proposed model also produces a probability-based anomaly score for each temporal sequence. This temporal behavior is illustrated in figure 5 using representative normal and abnormal sequences.

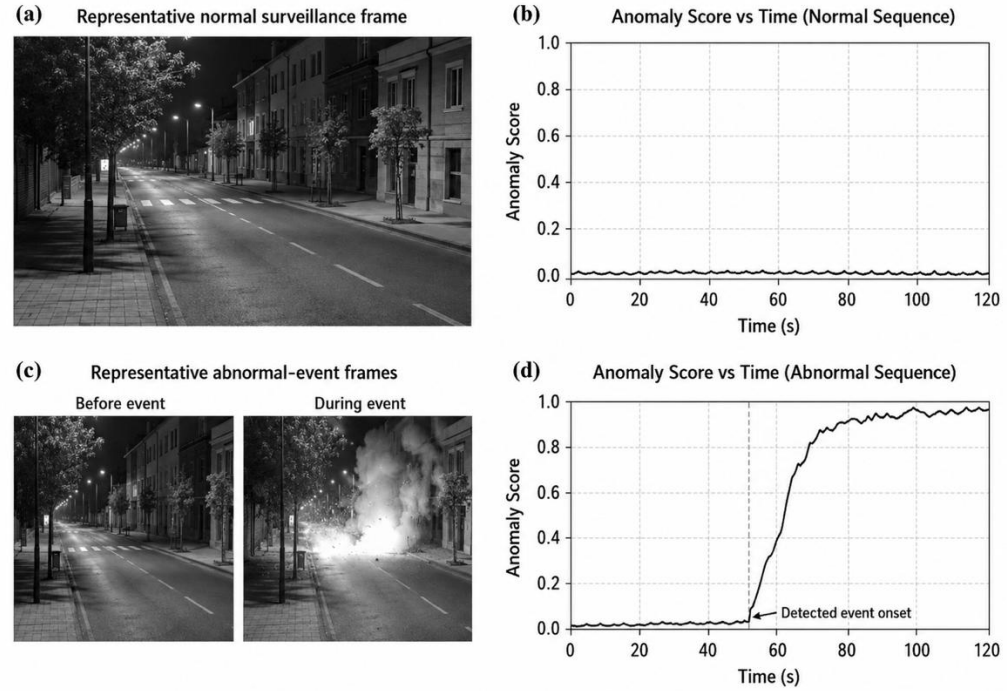

Figure 5. Temporal anomaly-score comparison for normal and abnormal sequences

As shown in figure 5(a) and figure 5(b), the representative normal sequence maintains an anomaly score close to zero throughout the observation period, indicating stable classification as a normal scene. In contrast, Figure 5(c) and Figure 5(d) illustrate an abnormal sequence containing an explosion event. Before the event, the anomaly score remains close to zero; when the abnormal event begins, the score increases sharply and subsequently remains at a high level. The detected event onset therefore corresponds to a clear temporal change in the model's output.

The temporal response demonstrates that the model can provide information beyond a single video-level classification by indicating when abnormal activity emerges within a sequence. In the representative explosion example, the strong increase in anomaly score corresponds to the appearance of smoke and intense visual disturbance. If the approximately 96% confidence value is obtained directly from the model's softmax output, it may be reported as the model's predicted probability for the Explosion class; otherwise, the confidence value should not be inferred solely from the confusion matrix. The discriminative capability of the model was further evaluated using one-versus-rest receiver operating characteristic (ROC) curves. The results are presented in figure 6.

As shown in figure 6, all five classes exhibit ROC curves substantially above the diagonal reference line, indicating strong discrimination between each target class and the remaining classes. The Normal class achieved the highest discrimination with an AUC of 1.00, followed by Explosion with 0.98, Fighting with 0.97, Road Accident with 0.96, and Burglary with 0.95. These results are consistent with the class-wise precision, recall, and F1-score values reported in Table 3 and further demonstrate the strong classification capability of the proposed MobileNetV2–LSTM model.

Overall, the evaluation results indicate that the proposed framework performs particularly well in distinguishing normal scenes from abnormal activities and maintains high discrimination across the five target classes. The remaining errors are primarily associated with visually similar abnormal events rather than confusion between normal and abnormal scenes. Future improvements could therefore focus on motion-aware attention mechanisms, object-level localization, optical-flow features, temporal feature enhancement, and additional training samples for difficult class pairs such as Fighting–Burglary and Road Accident–Explosion.

ISSN 2723-6471

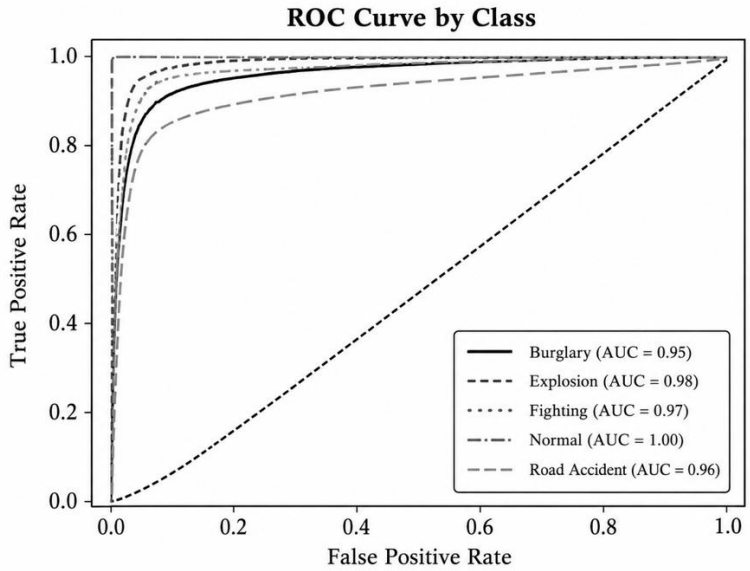

Figure 6. One-versus-rest ROC curves for the five activity classes

4.3. Edge-Device Performance and Practical Feasibility

Following model training, the proposed MobileNetV2–LSTM model was converted into TensorFlow Lite (TFLite) format and deployed on a Raspberry Pi 4 Model B with 4 GB RAM. The resulting edge-inference performance is summarized in table 6.

Table 6. Edge-deployment performance

Deployment measure	Result
Deployment device	Raspberry Pi 4 Model B
Device memory	4 GB RAM
Model format	TensorFlow Lite
Average inference latency	120–150 ms per frame
Processing throughput	Approximately 6–8 FPS
Deployment category	Near-real-time edge inference

As shown in table 6, the deployed model achieved an average inference latency of approximately 120–150 ms per frame, corresponding to a processing throughput of approximately 6–8 frames per second (FPS). These results demonstrate that the model can perform local inference directly on the Raspberry Pi without requiring continuous cloud-based processing or transmission of video data to a remote server. This edge-based configuration can reduce dependence on network connectivity and potentially improve privacy by keeping surveillance data locally processed.

The measured processing rate should, however, be interpreted in the context of the application requirements. A throughput of 6–8 FPS is below the 25–30 FPS commonly used for conventional surveillance video streams; therefore, the system is more accurately described as providing near-real-time or responsive edge inference, rather than full-frame-rate real-time processing. The proposed implementation is particularly suitable for applications in which activity recognition is performed on sampled frame sequences rather than on every frame of a high-frame-rate video stream.

The deployment results indicate that the MobileNetV2–LSTM architecture provides a favorable balance between classification performance and computational requirements. MobileNetV2 serves as a relatively lightweight spatial feature extractor, reducing the computational cost of processing individual frames, while the LSTM captures temporal dependencies across the sequence of extracted frame-level features. Converting the trained model to TensorFlow Lite further facilitates deployment on resource-constrained edge hardware.

 ISSN 2723-6471

Nevertheless, the reported computational performance should be interpreted together with the experimental limitations. In particular, the reported accuracy was obtained from a dataset containing only 100 curated source videos, which may not adequately represent the diversity of real-world surveillance environments. High performance may partly result from relatively clear class boundaries, recurring visual settings, or limited environmental variation. Similarly, the reported 6–8 FPS throughput and 120–150 ms latency should be understood as measurements for the specified Raspberry Pi configuration and implementation rather than universal performance values for all edge devices.

Therefore, the current results demonstrate the feasibility of deploying the proposed MobileNetV2–LSTM framework for near-real-time edge-based surveillance analysis, but they should not yet be interpreted as evidence of broad real-world generalization. Further evaluation using larger and more diverse datasets, unseen camera locations, crowded scenes, different illumination conditions, occlusions, varying camera viewpoints, and cross-dataset testing is required to establish the robustness and generalizability of the proposed framework.

5. Conclusion

This study presented a lightweight spatial–temporal framework that integrates MobileNetV2 and LSTM for classifying surveillance-video sequences into five activity categories: Normal, Fighting, Burglary, Road Accident, and Explosion. MobileNetV2 was used to extract efficient frame-level spatial representations, while the LSTM modeled temporal dependencies across consecutive frames. The proposed framework achieved an accuracy of 97.41%, a macro F1-score of 0.9612, and a weighted F1-score of 0.9744 across 463 evaluated sequences. Most classification errors occurred between visually similar activities, particularly Fighting and Burglary, indicating that the principal challenge lies in fine-grained discrimination among human-centered anomaly classes rather than in separating normal from abnormal activity.

Conversion to TensorFlow Lite enabled the model to operate on a Raspberry Pi 4 with an average latency of approximately 120–150 ms per frame and a processing rate of 6–8 FPS. These findings demonstrate that the architecture provides a practical balance between semantic event classification, temporal modeling, and computational efficiency. The model is therefore suitable as a foundation for near-real-time surveillance applications in resource-constrained environments, although its current performance should not yet be interpreted as evidence of universal real-world generalization.

The primary limitation of this study is the relatively small and curated dataset, which may contain clearer class boundaries than unconstrained surveillance environments. Future research should evaluate the framework using larger benchmark datasets, unseen camera locations, crowded scenes, low-light conditions, and cross-dataset validation. Further improvements may include attention-based motion representation, object localization, model quantization, and expanded edge-performance profiling to improve discrimination among similar anomalies and establish deployment reliability under more diverse operational conditions.

6. Declarations

6.1. Author Contributions

Conceptualization: S.A.N., R.K.M.; Methodology: S.A.N., V.S.R.; Software: R.K.M., V.S.R.; Validation: S.A.N., R.K.M.; Formal Analysis: S.A.N.; Investigation: R.K.M., V.S.R.; Resources: S.A.N., V.S.R.; Data Curation: R.K.M.; Writing – Original Draft Preparation: S.A.N.; Writing – Review and Editing: R.K.M., V.S.R.; Visualization: V.S.R.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Pavithra, B. Paul Steve Mithun, T. R. Sofen, C. Shanmuga Prasath, and V. Saravana, "Real-time suspicious activity detection in bank and ATM environments using meta-learning and edge computing," *ICDSAAI*, vol. 2025, no. Mar., pp. 1–6, Mar. 2025, doi: 10.1109/ICDSAAI65575.2025.11011591.

[2] T. Ogino, H. Fukuda, and P. Leger, "Stepwise distributed execution for load reduction in machine learning systems," *ISPDC*, vol. 2025, no. Jul., pp. 127–131, Jul. 2025, doi: 10.1109/ISPDC67428.2025.00028.

[3] K. Pawar and V. Attar, "Application of deep learning for crowd anomaly detection from surveillance videos," *Confluence*, vol. 2021, no. Jan., pp. 506–511, Jan. 2021, doi: 10.1109/Confluence51648.2021.9377055.

[4] H. Wahab, I. Mehmood, H. Ugail, A. K. Sangaiah, and K. Muhammad, "Machine learning based small bowel video capsule endoscopy analysis: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 143, no. Jun., pp. 191–214, Jun. 2023, doi: 10.1016/j.future.2023.01.011.

[5] W. Yang, Z. Jiaogen, Y. Jun, and G. Jihong, "Survey of anomaly detection methods in surveillance videos based on deep learning," *J. Image Graph.*, vol. 30, no. 3, pp. 615–640, Mar. 2025, doi: 10.11834/jig.240329.

[6] U. M. Muna, S. Biswas, S. A. A. M. Zarif, P. J. Deori, T. Tajwar, and S. Shatabda, "Vision transformer embedded video anomaly detection using attention driven recurrence," *Array*, vol. 27, no. Sep., pp. 1–12, Sep. 2025, doi: 10.1016/j.array.2025.100471.

[7] D. Atienza, C. Bielza, J. Diaz-Rozo, and P. Larranaga, "Efficient anomaly detection in a laser-surface heat-treatment process via laser-spot tracking," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 1, pp. 405–415, Feb. 2021, doi: 10.1109/TMECH.2020.3024613.

[8] B. Tong, Y. Li, and X. Chen, "A neural network-based intelligent system for substation surveillance video analysis with edge and IoT integration," *EURASIP J. Wireless Commun. Netw.*, vol. 2025, no. Nov., pp. 1–18, Nov. 2025, doi: 10.1186/s13638-025-02528-y.

[9] K. Pawar and V. Attar, "Automated surveillance model for video-based anomalous activity detection using deep learning architecture," *Adv. Intell. Syst. Comput.*, vol. 2020, no. Sep., pp. 327–334, Sep. 2020, doi: 10.1007/978-981-15-6067-5_36.

[10] X. Yuan, "MEN-AD: Multimodal zero-shot detection of mucus anomalies in endoscopy," *Commun. Comput. Inf. Sci.*, vol. 2025, no. Jul., pp. 85–97, Jul. 2025, doi: 10.1007/978-981-95-0017-8_8.

[11] X. Guo, L. Song, W. Zhu, F. Du, and Z. Ma, "Review of low-shot industrial image anomaly detection," *Comput. Eng. Appl.*, vol. 61, no. 13, pp. 26–45, Jul. 2025, doi: 10.3778/j.issn.1002-8331.2408-0230.

[12] M. Yasar Arafath and A. Niranjil Kumar, "Quantum computing based neural networks for anomaly classification in real-time surveillance videos," *Comput. Syst. Sci. Eng.*, vol. 46, no. 2, pp. 2489–2508, Feb. 2023, doi: 10.32604/csse.2023.035732.

[13] J. Wang, M. Wang, Q. Liu, G. Yin, and Y. Zhang, "Deep anomaly detection in expressway based on edge computing and deep learning," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 3, pp. 1293–1305, Mar. 2022, doi: 10.1007/s12652-020-02574-y.

[14] O. Alruwaili, A. Armghan, K. Salem Alshudukhi, A. Flah, and I. Pergl, "Cloud computing based intelligent video surveillance framework for logistics security," *IEEE Access*, vol. 12, no. Oct., pp. 150604–150622, Oct. 2024, doi: 10.1109/ACCESS.2024.3471688.

[15] S. B. Venkata, C. H. Wong, V. S. Karwande, G. N. Divyaraj, S. Singh, and R. V. Patil, "Metaheuristic-tuned GraMNet architecture for enhanced video-based anomaly detection using UCF50 dataset," *ISAC3*, vol. 2025, no. Aug., pp. 1–8, Aug. 2025, doi: 10.1109/ISAC364032.2025.11156508.

[16] S. Gayathri, G. Shanthakumari, A. Velvizhi V., R. Anitha, B. Priyadharshini, and A. Kiruthika, "Design and development of IoT-based domestic surveillance system with real-time anomaly detection and chatbot integration," *ICCCT*, vol. 2025, no. Apr., pp. 1–7, Apr. 2025, doi: 10.1109/ICCCT63501.2025.11019793.

[17] J. Wang, M. Wang, G. Yin, and Y. Zhang, "DAD: Deep anomaly detection for intelligent monitoring of expressway network," *Lect. Notes Comput. Sci.*, vol. 2020, no. Sep., pp. 65–72, Sep. 2020, doi: 10.1007/978-3-030-62223-7_6.

[18] K. M. Biradar, M. Mandal, S. Dube, S. K. Vipparthi, and D. K. Tyagi, "Triplet-set feature proximity learning for video anomaly detection," *Image Vis. Comput.*, vol. 150, no. Dec., pp. 1–12, Dec. 2024, doi: 10.1016/j.imavis.2024.105205.

[19] A. Ruggeri, A. Ficara, G. Sollazzo, G. Bosurgi, and M. Fazio, "A comparative analysis of deep learning approaches for road anomaly detection," *ISCC*, vol. 2024, no. Jul., pp. 1–6, Jul. 2024, doi: 10.1109/ISCC61673.2024.10733559.

[20] A. Kumar, H. Hashmi, S. A. Khan, and S. Kazim Naqvi, "SSE: A smart framework for live video streaming based alerting system," *SMART*, vol. 2021, no. Dec., pp. 193–197, Dec. 2021, doi: 10.1109/SMART52563.2021.9675306.

[21] M. P. S. Bhatia and S. R. Sangwan, "Soft computing for anomaly detection and prediction to mitigate IoT-based real-time abuse," *Pers. Ubiquitous Comput.*, vol. 28, no. 1, pp. 123–133, Jan. 2024, doi: 10.1007/s00779-021-01567-8.

[22] S. R. Mishra, T. K. Mishra, A. Sarkar, and G. Sanyal, "Detection of anomalies in human action using optical flow and gradient tensor," *Smart Innov. Syst. Technol.*, vol. 2020, no. Jan., pp. 561–570, Jan. 2020, doi: 10.1007/978-981-13-9282-5_53.

[23] R. Singh *et al.*, "iMAppGAN: Integrated motion appearance generative adversarial networks for video anomaly detection," *Commun. Comput. Inf. Sci.*, vol. 2025, no. Jul., pp. 388–402, Jul. 2025, doi: 10.1007/978-981-96-6972-1_27.

[24] V. A. Kotkar and V. Sucharita, "Fast anomaly detection in video surveillance system using robust spatiotemporal and deep learning methods," *Multimedia Tools Appl.*, vol. 82, no. 22, pp. 34259–34286, Aug. 2023, doi: 10.1007/s11042-023-14840-0.

[25] H. Khan, W. F. Alfwzan, R. Latif, J. Alzabut, and R. Thinakaran, "AI-based deep learning of the water cycle system and its effects on climate change," *Fractal and Fractional*, vol. 9, no. 6, Art. no. 361, pp. 1-22, 2025.