International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

AI-Based Surveillance System for Crowd Behavior and Riot Detection

Arin Talavadekar
B.E. AI&DS
Terna Engineering College
Navi Mumbai, India
arintalavadekar2223@ternaengg.ac.in

Maheep Kaur Chopra
B.E. AI&DS
Terna Engineering College
Navi Mumbai, India
maheepchopra2223@ternaengg.ac.in

Ashwini Panada
B.E. AI&DS
Terna Engineering College
Navi Mumbai, India
ashwinipanada2223@ternaengg.ac.in

Atharva Darke
B.E. AI&DS
Terna Engineering College
Navi Mumbai, India
atharvadarke2223@ternaengg.ac.in

Dr. Sandeep Raskar
Guide
Dept. of AI & DS, Terna Engineering College
Navi Mumbai, India
raskarsandeep@ternaengg.ac.in

Abstract—The increasing adoption of intelligent video surveillance systems in public security settings has intensified the need for automated approaches capable of monitoring crowded envi-ronments effectively. In conventional surveillance infrastructures, human operators are required to supervise multiple video streams simultaneously, which limits the reliable detection of infrequent yet critical events and often results in reduced performance due to fatigue and attention constraints. To mitigate these limitations, a two-level deep learning framework is proposed for the analysis of dense crowd scenes, with the objective of identifying both abnormal crowd behavior and explicit security threats such as weapons. In the first stage, a YOLOv8-based object detection model is utilized for real-time processing to enable person localization, crowd density estimation, and the detection of visible weapons. Frames that contain relevant activity are subsequently processed using an EfficientNet-B0 network, chosen for its favorable balance between recognition accuracy and computa-tional efficiency, to extract spatial feature representations. These features are then analyzed using an attention-based bidirectional gated recurrent unit (BiGRU) network, which models temporal dependencies across frame sequences to classify overall crowd behavior. The incorporation of an attention mechanism allows the system to emphasize temporally informative frames, thereby improving sensitivity to the onset and progression of abnormal activities. The behavioral classification module is trained and evaluated on the UCF-Crime dataset, and the experimental results demonstrate reliable performance in practical anomaly detection scenarios. Furthermore, the complete framework is implemented as a real-time, modular web application using the Flask framework, incorporating an interactive dashboard, alert generation, and event logging. This implementation illustrates the feasibility of deploying the proposed system in real-world surveillance environments.

Index Terms—Video anomaly detection, weapon detection, crowd monitoring, deep learning, intelligent video surveillance, YOLOv8, EfficientNet-B0, attention mechanism, bidirectional GRU, real-time systems.

I. Introduction

The widespread deployment of surveillance cameras in public spaces, driven by declining hardware costs and increasing security concerns, has led to an unprecedented growth in the volume of video data generated. It is estimated that billions of cameras are currently in operation worldwide, producing data at the scale of exabytes each day. Monitoring such a large volume of video manually is neither scalable nor cost-effective and is highly susceptible to human error caused by fatigue and limited attention spans [14], [15]. As a result, automatic video analysis through Intelligent Video Surveillance (IVS) has become a critical component of modern public safety systems, enabling faster incident response, continuous crowd observation, and more efficient allocation of security resources [13], [14].

IVS solutions are particularly relevant in environments characterized by high pedestrian density, such as airports, railway stations, metro systems, and other crowded public areas. In these settings, the early detection of suspicious or abnormal activity plays an essential role in minimizing the severity and overall impact of potential security incidents.

Anomalous behavior in crowd scenes can take many different forms. Some events involve relatively minor actions such as pickpocketing, while others correspond to serious incidents including fights, riots, stampedes, or explosions [14], [15]. Detecting these behaviors is challenging because they often occur in visually complex environments and can evolve rapidly over time [15]. In many cases, the same visual action may have different meanings depending on the surrounding context. For instance, a person running may indicate normal physical activity or may signal an attempt to escape from danger. High crowd density further complicates analysis by introducing frequent occlusions, which limit clear visual observation [14].

In addition, anomalous events occur much less frequently than normal behavior, leading to strong class imbalance in available data. This imbalance can cause learning algorithms to favor the normal class if not handled carefully [14], [15].

In addition to behavioral anomalies, the presence of explicit threats such as firearms or knives introduces a further layer of complexity in surveillance scenarios. Such objects may emerge prior to the onset of an anomalous event, during its progression, or as a direct consequence of the incident itself. Their reliable detection therefore demands accurate and fine-grained object recognition capabilities that function in conjunction with broader crowd behavior analysis [1]–[5].

To jointly address these challenges, a multistage deep learning framework is adopted that separates object detection, spatial feature extraction, and temporal behavior classification into distinct processing components. This modular architecture not only improves computational efficiency but also facilitates system extensibility, allowing individual modules to be refined or replaced as more effective models become available.

The main contributions of this paper are summarized as follows:

1) This work introduces a three-stage hybrid architecture that integrates a state-of-the-art YOLOv8-based object detection model for perceptual analysis [5], an EfficientNet-based network for spatial feature extraction [6], and a temporal behavior classification module implemented using an attention-based bidirectional GRU [8], [11], [12]. The integration of these components enables joint modeling of spatial and temporal information, thereby enhancing the detection of anomalous crowd behavior as well as the identification of specific threat-related events [14], [15].
2) The proposed system supports dual detection within a unified processing pipeline by simultaneously analyzing overall crowd behavior and identifying critical objects such as weapons. This integrated design enables the differentiation between normal and anomalous activities while providing early recognition of explicit threats. Such an approach is consistent with recent surveillance research that emphasizes the combined use of anomaly detection and object-specific recognition to enhance situational awareness and enable a multi-layered security response [1]–[4], [13]–[15].
3) This system was designed and deployed as an end-to-end framework, implemented as a Flask-based web application, and it supports real-time processing, alerting, and logging. Demonstrated herein is a practical and viable solution worthy of consideration for real-world surveillance environments, consistent with design principles observed in recent intelligent video surveillance research [13], [14].

The rest of this paper is organized as follows. Section II reviews related works on object detection, video feature learning, and anomaly detection. Section III elaborates on the specific architecture and individual components of the proposed framework. Section IV describes the experimental settings in terms of the dataset being used and the implementation details. Section V presents the quantitative results, analyzes the performance, and discusses the limitations. Section VI describes the real-time web application deployment. Finally, Section VII provides the conclusions and ideas about potential directions for future work.

II. Related Work

Our work intersects with and contributes to the contemporary progress at the forefront of several central domains in computer vision and deep learning, namely object detection, spatio-temporal feature learning, attention mechanisms, and benchmarks pertaining to video anomaly detection.

A. Object Detection for Surveillance

Object detection is the fundamental perceptual layer of most IVS systems, which offers a way to locate and classify entities relevant to applications [14], [24]. The YOLO family of models has significantly furthered this area by reformulating detection as a single regression problem, mapping image pixels directly to bounding boxes and class probabilities, hence achieving a high degree of accuracy while allowing real-time processing capabilities [2]. Early versions established this one-stage framework [2]. Subsequent variants, such as YOLOv3, proposed several new features for improving the detection accuracy of objects, particularly smaller ones, such as multi-scale predictions and a more complex backbone (Darknet-53) [3]. Later variants, such as YOLOv4 and YOLOv7, introduced various techniques in the form of “bag-of-freebies” and “bag-of-specials” that raised the performance bar further (e.g., data augmentation, more optimized activation functions, improved necks such as PANet) [1], [4]. Very recently, developed by Ultralytics, YOLOv8 has emerged as a state-of-the-art model [5]. It has incorporated significant architectural changes, such as an anchor-free detection head and a novel backbone called C2f (Cross Stage Partial bottleneck with two convolutions), inspired by CSPNet [5]. Its new design provides an improved accuracy–speed trade-off compared to its predecessors, making it effective in real-time challenging surveillance tasks with characteristics such as tracking several persons in highly crowded scenarios and the reliable detection of small-size but critical objects like weapons [1], [4], [5], [24].

B. Spatio-Temporal Feature Learning

Effective analysis of video data requires attention to the visual information contained in each individual frame as well as the way this information changes over time across a sequence of frames [14], [15]. Probably the most established and historically influential strategy is the two-stream architecture: one stream (usually a CNN) processes spatial information from static frames, while another stream (often an RNN or an optical-flow-based network) encodes temporal information. Compared to end-to-end 3D CNNs, such as C3D and I3D, that learn spatiotemporal features jointly and directly from video volumes [25], [26], the two-stream paradigm often provides more flexibility and better computational efficiency, especially

when leveraging powerful 2D CNN backbones pre-trained on large image datasets like ImageNet [22]. Such an approach facilitates the use of a robust spatial feature extractor built on mature 2D convolutional architectures. We also follow this paradigm in the presented work.

For the task of spatial feature extraction, we utilize EfficientNet. This family of models demonstrated that compound scaling—balancing network depth, width, and input resolution—yields significantly better accuracy and efficiency compared to the earlier practice of scaling only one dimension, as seen, for example, with ResNet models [6], [7]. The baseline EfficientNet-B0 model provides a desirable balance between computational cost and accuracy, making it an attractive choice as a feature extractor for real-time applications with potential resource constraints [6].

C. Attention Mechanisms in Vision and Sequence Modeling

To study how video content evolves over time, recurrent neural networks and their variants, including LSTMs and GRUs, are widely used [8], [9]. GRUs often give performance comparable to LSTMs with fewer parameters, hence a preferred choice for efficiency [8]. A BiGRU further strengthens the modeling by processing the sequence in both directions, enabling the hidden state of every time step to capture context from both forward and backward frames [12].

Such sequence models have been significantly empowered with the introduction of attention mechanisms. Initially popular in natural language processing for machine translation, attention allows a model to dynamically weight the relevance of different parts of an input sequence towards an output [10], [11]. For video analysis, this translates to a model learning to assign higher attention weights over frames or temporal segments where anomalous actions begin, peak, or exhibit the most discriminative cues [14], [15]. This attention on relevant information can improve classification performance, robustness to noise or irrelevant activities, and potentially interpretability. The model presented here includes a self-attention layer after the BiGRU to make full use of the above advantages.

D. Public Datasets for Anomaly Detection

The development of VAD has been closely coupled with public benchmark datasets. Initial datasets, such as UCSD Ped1/Ped2 and CUHK Avenue, primarily focused on appearance-based anomalies or short, pre-selected clips, and they have been extensively discussed in prior surveys on video anomaly detection [14], [15]. The CUHK Avenue dataset introduced more diversified, action-based anomalies, including running and object throwing [14], [15]. However, these datasets generally consist of short, curated video segments, limiting their complexity and realism.

In this study, we adopt a more modern and significantly challenging dataset, UCF-Crime [13]. It includes 1,900 long, raw real-world surveillance videos from 13 different anomaly categories, such as Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. Its scale, diversity, and realism—stemming from cluttered backgrounds, varied camera viewpoints, and long continuous recordings—make it a demanding and representative benchmark for evaluating modern VAD systems designed for real-world deployment.

III. Proposed Framework

The proposed framework is organized as a sequential three-stage processing pipeline that transforms raw video input into interpretable information related to crowd behavior and potential security threats. The modular design of the system enables each processing stage to be developed, evaluated, and refined independently, thereby facilitating incremental improvements as the overall framework evolves. An overview of the complete workflow, including data flow and key processing components, is presented in Fig. 1.

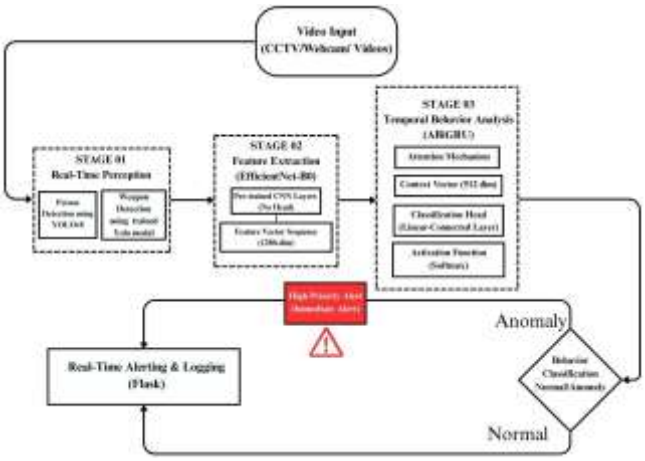

Fig. 1. Workflow diagram of the proposed surveillance system.

A. Stage 1: Perception with YOLOv8

The first stage functions as the primary perception component of the system and employs a pre-trained YOLOv8 model due to its high detection accuracy and real-time processing capability [5], [24]. Each incoming video frame is processed independently, enabling efficient localization and identification of objects of interest with minimal latency. Within this stage, two parallel tasks are carried out, both of which are essential for effective surveillance:

- **Person Detection and Tracking Support:** The model identifies all individuals present within each video frame and produces bounding box coordinates for every detected person [1]–[5]. These detections serve as a critical input to subsequent stages of the framework, enabling population counting and providing a real-time estimate of crowd density [14], [15]. Furthermore, the generated bounding boxes are compatible with multi-object tracking approaches, as tracking-by-detection methods depend on accurate frame-level object localization [20]. Although explicit tracking is not incorporated into the anomaly

detection pipeline in this work, the outputs generated by the YOLOv8 model allow such functionality to be integrated if required.

- **Weapon Detection:** A dedicated YOLOv8-based model is employed for weapon detection and is fine-tuned using a dataset comprising multiple weapon categories, with the current implementation primarily focused on firearms. Each video frame is examined to identify visible high-priority threat objects, enabling early threat recognition within the surveillance pipeline [5], [24]. When a weapon is detected with high confidence, the corresponding event can be forwarded directly for immediate alert generation, without waiting for behavior-level analysis. This design choice reflects the need for rapid response in scenarios involving explicit security threats [1]–[4], [14], [15].

The outputs generated by this stage for each frame include a list of detected objects, each associated with its bounding box coordinates (typically formatted as (x_{center}, y_{center}, width, height)), a class label ("person" or "weapon"), and a confidence score indicating the model's certainty in the detection. An example output frame with detected bounding boxes is shown in Fig. 2.

Fig. 2. Output frame from the YOLOv8 perception stage, showing detected bounding boxes.

B. Stage 2: Feature Extraction with EfficientNet

For frames relevant to the behavioral analysis pipeline—that is, frames containing individuals, sampled at a prescribed rate—spatial features are extracted using a pre-trained EfficientNet-B0 model. EfficientNet models are known for achieving high performance with significantly fewer parameters and floating-point operations (FLOPs) compared to other popular architectures like ResNet [6], [7]. We use the convolutional base of EfficientNet-B0, pre-trained on ImageNet, discarding its last fully connected classification layer to turn the network into a powerful feature extractor [6]. The input frames, which are usually resized to 224×224 pixels, are fed through the convolutional layers. The output from the last convolutional block—before the global average pooling in the original model—is subsequently pooled and flattened in order to obtain a fixed-size feature vector. Herein, every frame feeds into a 1280-dimensional feature vector. That vector efficiently captures the high-level representation of the frame's visual content, encoding information about textures, shapes, object parts, and spatial relationships, and can therefore be fed into the subsequent temporal modeling stage [22].

C. Stage 3: Temporal Analysis with Attention-based Bi-GRU (ABiGRU)

We use the sequence of frame-level feature vectors produced by the EfficientNet component, which is fed into our Attention-based Bidirectional GRU (ABiGRU) network. The network captures temporal relationships across frame sequences by integrating gated recurrent units (GRUs) with bidirectional recurrent processing [8], [12]. The incorporation of an attention mechanism enables the model to assign greater importance to temporally informative segments that contribute most to classification decisions [10], [11]. Collectively, these components allow the system to produce a final prediction that accurately reflects the overall behavior exhibited within a given video segment.

- **Bidirectional GRU Layers:** The temporal modeling component is implemented using two stacked bidirectional GRU layers [12], each comprising 256 hidden units. For illustration, a sequence consisting of 100 feature vectors extracted from 100 consecutive video frames is considered. The bidirectional GRU processes this sequence in both temporal directions, propagating information from past to future as well as from future to past. This design allows the hidden state at each time step to incorporate information from the full temporal context instead of relying only on earlier frames. The output is a sequence of feature representations where each time step contains a concatenation of the forward and backward hidden states.
- **Attention Mechanism:** A self-attention mechanism inspired by earlier work [10], [11] is applied after the BiGRU layers. For each hidden state produced by the BiGRU, the mechanism computes a relevance score by comparing it with all other hidden states, often using a small feed-forward network or a dot product similarity measure. These scores are normalized through a softmax function, which yields attention weights a_t for each time step t. The weights sum to 1 and represent how important each frame is for the final classification task.
  A context vector c is then computed as

$$c = \sum_t a_t h_t \quad (1)$$

  where h_t denotes the hidden state at time t. This context vector summarizes the most informative parts of the sequence and allows the model to concentrate on key moments, for example the start of a physical conflict or the first sign of smoke, while giving less weight to frames that do not contribute meaningfully.
- **Classification Head:** The context vector c, which contains the temporally weighted information, is passed

through one or more dense layers that form the classification head [22]. This produces the final prediction for the video segment. In the current binary scenario—"Normal" vs. "Anomaly"—a single output neuron with a sigmoid activation suffices.

- **Dropout for Regularization:** To prevent overfitting, especially considering that most anomaly datasets are relatively small and imbalanced, a dropout layer is systematically included [22]. Dropout randomly deactivates a portion of neuron activations during training so that the network does not rely on specific features or the co-adaptation of neurons. In this model, dropout is applied following the BiGRU layers and before the final classification layer to improve generalization on unseen videos.

A conceptual representation of the ABiGRU architecture, emphasizing the attention weighting process, is shown in Fig. 3.

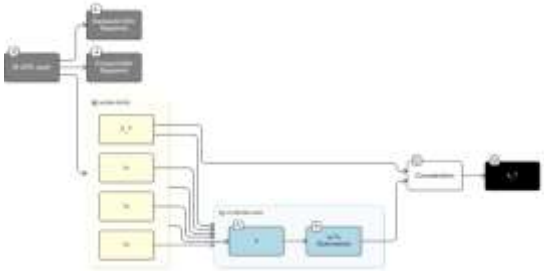

Fig. 3. Conceptual diagram illustrating the flow within the Attention-based Bi-GRU model.

IV. Experimental Setup

A. Dataset and Pre-processing

The key behavioral classification module (EfficientNet + ABiGRU) was subjected to extensive training and testing on the UCF-Crime dataset [13]. As mentioned earlier, this dataset is a large-scale, challenging benchmark consisting of 1,900 long, untrimmed real-world surveillance videos with 13 different anomaly categories: Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Road Accidents, Robbery, Shooting, Stealing, Shoplifting, and Vandalism, along with a sizeable collection of Normal activity videos captured under similar settings.

For the binary classification problem, i.e., "Normal" versus "Anomaly," all 13 anomaly categories were merged into a single class labeled "Anomaly," following common practice in anomaly detection research [14], [15]. Preprocessing was required because the dataset shows a strong class imbalance and because the lengths of the video recordings vary considerably [14], [15]. During preprocessing, representative segments were selected from both Normal and Anomaly videos so that the resulting training data captured a broader range of observed behaviors.

The final training dataset comprises sequences extracted from 400 anomalous videos and 150 normal videos. Each sequence contains 100 frames, corresponding to approximately 100 s of activity at the selected sampling rate. This sequence length was chosen to provide sufficient temporal context for effective behavior analysis while maintaining computational requirements within practical limits.

B. Implementation Details

The complete processing pipeline was implemented in Python, with the core deep learning components developed using the **PyTorch** framework.

- **Video Processing and Feature Extraction:** The OpenCV library was utilized for video loading and preprocessing. Each video stream was decoded, and frames were sampled uniformly at a rate of one frame per second. The sampled frames were converted to RGB format and resized to 224 × 224 pixels, matching the input resolution required by the pre-trained EfficientNet-B0 model. Spatial feature extraction was performed on a per-frame basis, yielding feature vectors with a dimensionality of 1280. Temporal sequences were subsequently constructed by grouping 100 consecutive feature vectors, which were then provided as inputs to the temporal classification model.
- **Model Training:** The ABiGRU network was trained using the Adam optimizer with an initial learning rate of 0.0005 [22]. Training was conducted for 20 epochs with a batch size of 32 sequences. To address the pronounced class imbalance between normal and anomalous samples, a weighted cross-entropy loss function was employed [22]. In this formulation, a higher penalty was assigned to misclassifications of the normal class, which helps preserve accuracy on normal behavior while reducing bias toward predicting anomalous events.

  The dataset was split into an 80 percent training set and a 20 percent validation set to assess the model's ability to generalize. The training process was closely monitored by observing the loss values for both the training and validation sets across epochs, and these trends are illustrated in Fig. 4.

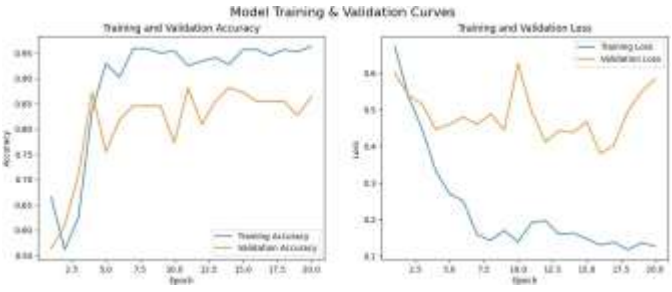

Fig. 4. Training and validation loss values observed over the training epochs.

C. Evaluation Metrics

The performance of the trained ABiGRU model was quantitatively evaluated on a held-out validation set using standard metrics for binary classification:

- **Accuracy:** The proportion of sequences that are correctly classified.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

- **Precision:** The proportion of sequences predicted as anomalous that were actually anomalous. High precision indicates fewer false alarms.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

- **Recall (Sensitivity):** The proportion of actual anomalous sequences that were correctly detected. High recall indicates fewer missed anomalies.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

- **F1-Score:** The harmonic mean of Precision and Recall, providing a single balanced measure of detection performance.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Here, TP (True Positives) are anomalies correctly identified, TN (True Negatives) are normal sequences correctly identified, FP (False Positives) are normal sequences incorrectly flagged as anomalous (false alarms), and FN (False Negatives) are anomalous sequences that the model fails to detect. These metrics were computed separately for each class and also as a weighted average according to class support..

V. Results and Discussion

A. Quantitative Performance

The performance of the core anomaly detection module (EfficientNet feature extractor followed by the ABiGRU classifier) on the validation subset of the preprocessed UCF-Crime dataset [13] was quantitatively very robust. The model achieved a final overall validation accuracy of 86.4%. Detailed class-wise performance metrics—in terms of Precision, Recall, F1-Score, and the number of samples (Support) for each class in the validation set—are presented in Table I.

TABLE I
Classification Performance on the Validation Set

Class	Precision	Recall	F1-Score	Support
Normal	0.75	0.72	0.74	29
Anomaly	0.90	0.91	0.91	81
Accuracy		0.86		
Macro Avg	0.83	0.82	0.82	110
Weighted Avg	0.86	0.86	0.86	110

The model exhibits strong performance in detecting anomalous events. For the *Anomaly* class, a precision of 0.90 and a recall of 0.91 are achieved, resulting in an F1-score of 0.91. This performance is well suited to surveillance applications, as it represents a balanced trade-off between effective threat detection and the reduction of false alarms. High recall contributes to minimizing missed anomalous events, while high precision helps limit unnecessary alerts.

Performance on the *Normal* class is comparatively lower, with an F1-score of 0.74. This outcome is expected, given that normal samples constitute a smaller portion of the dataset, even when a weighted loss function is applied during training. The weighted average F1-score further indicates that the model maintains consistent performance across both classes.

Additional insight into the classification behavior is provided by the confusion matrix presented in Table II, which reports the distribution of true positives, true negatives, false positives, and false negatives.

TABLE II
Confusion Matrix

Actual Label	Predicted Label: Normal	Predicted Label: Anomaly
Normal	21 (TN)	8 (FP)
Anomaly	7 (FN)	74 (TP)

Among the 29 sequences belonging to the *Normal* class, 21 were correctly classified as normal, corresponding to true negatives, while 8 were misclassified as anomalous, corresponding to false positives. For the *Anomaly* class, 74 out of 81 sequences were correctly identified as anomalous, representing true positives, whereas 7 sequences were incorrectly classified, resulting in false negatives. These findings indicate that the proposed model effectively detects anomalous behavior while maintaining a relatively low false-positive rate.

B. Discussion and Limitations

The experimental results demonstrate the effectiveness of the proposed multistage hybrid architecture. The initial perception layer, implemented using YOLOv8, reliably detects key objects within the scene, including individuals and potential weapons, and provides meaningful contextual information for subsequent processing stages [5], [24].

The selection of EfficientNet-B0 as the spatial feature extractor further contributes to system performance by achieving a favorable balance between representational capability and computational efficiency [6]. The ABiGRU-based temporal modeling component performs effectively and highlights the benefits of combining bidirectional recurrent processing with an attention mechanism for sequential analysis [8], [10]–[12].

Qualitative evaluation, particularly the analysis of attention weight distributions across selected sequences, indicates that the model learns to emphasize the most informative segments within the 100-frame temporal window. Frames associated with abnormal behavior receive higher attention, while those corresponding to normal or redundant activity are assigned lower importance. Additionally, the bidirectional GRU structure ensures that contextual information from both preceding and succeeding frames contributes to the classification outcome at each temporal step [12].

Although the results are promising, the system still has certain inherent limitations. Performance may decrease in environmental conditions that are not well represented in the training data, such as very low illumination, heavy rain

or fog, or significant occlusions in the camera view [14], [15]. The accuracy of the model depends directly on the quality, diversity, and representativeness of the datasets used for training, including UCF-Crime and the weapon dataset [13].

Consequently, it may inherit biases present in these datasets and could struggle with entirely novel categories of anomalies or weapons not encountered during training. Current weapon detection is restricted to overt, visible firearms and is likely to miss concealed weapons or other weapon types, such as knives or explosives. Furthermore, the fixed sequence length of 100 frames may be inadequate for anomalies that unfold over longer timescales, or unnecessarily long for very short anomalous events, thus introducing noise into the temporal representation.

Some possible directions for future work directions include the following:

- **Robustness Enhancement:** Exploring domain adaptation and generalization techniques such as adversarial training and data augmentation that recreate challenging environmental conditions. The aim is to improve performance under varying lighting conditions, changing weather, and different camera viewpoints.
- **Alternative Architectures:** Exploring end-to-end 3D convolutional neural networks and video transformer-based models as possible alternatives or complementary solutions to the current architecture, which uses EfficientNet for spatial feature extraction and ABiGRU for temporal analysis.
- **Explainability:** Incorporating explainable artificial intelligence techniques to visualize attention weights or assess feature importance. Such methods can provide insight into the model's decision-making process and enhance interpretability, user trust, and ease of debugging.
- **Extended Threat Detection:** Extending the weapon detection component to recognize additional object categories, such as knives or unattended packages. The integration of audio signals may further support a more comprehensive multimodal threat detection framework.
- **Adaptive Sequence Length:** Developing adaptive strategies that allow the system to dynamically adjust the temporal window length based on observed activity. This capability would enable more effective handling of both short-duration incidents and events that evolve over longer time spans.

VI. Real-Time Deployment

To bridge the gap between offline model development and real-world deployment, a prototype system for real-time anomaly and weapon detection was implemented using the Flask micro web framework. Flask was selected for its lightweight design, flexibility, and straightforward integration with Python-based machine learning libraries.

The system includes a user-oriented dashboard accessible through a standard web browser, which serves as the primary monitoring and interaction interface.

Using this dashboard, users are able to select video input sources, view the processed video streams, and receive notifications generated by the detection pipeline. The supported input sources include:

- Local webcam feeds
- Network IP camera streams provided through RTSP or similar protocols
- Pre-recorded video files supplied by the user

The Flask back-end manages the video input stream and executes the full **three-stage detection pipeline** in near real time. For each frame, or for frames sampled according to processing constraints, the system performs **YOLOv8**-based detection followed by **EfficientNet** feature extraction when required. The **ABiGRU** model then processes the resulting sequence of feature vectors.

The output of this pipeline is a processed video stream that includes bounding box overlays for detected persons and weapons and a status indicator showing the current behavioral classification, either **"Normal"** or **"Anomaly"**. This augmented stream is transmitted back to the user's browser through Motion JPEG streaming, WebSockets, or comparable techniques.

The core functionality of the system relies on an alerting and logging mechanism. When the ABiGRU model flags a sequence as an **"Anomaly"**, or when **YOLOv8** identifies a weapon with sufficient confidence, the system performs two actions:

- A clear visual alert is shown on the dashboard, for example a flashing banner or a change in color tone, to immediately capture the operator's attention.
- The event is recorded in durable storage, such as a database or log file. Each entry includes the event type (anomaly or weapon), the exact timestamp, and optionally a short video clip or sequence of frames for later review and analysis.

This prototype demonstrates the integration and practical viability of the proposed framework in a real-time application, although further tuning may be required for large-scale deployments, such as introducing GPU acceleration or distributed processing. A screenshot of the web application interface is shown in Fig. 5.

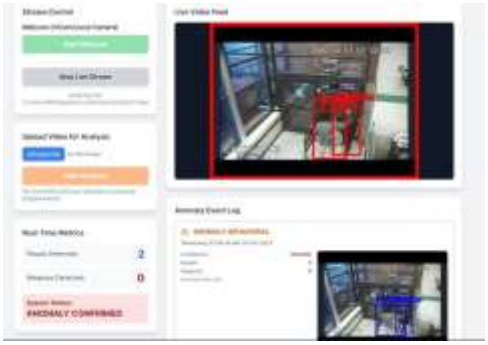

Fig. 5. Dashboard of the real-time Flask web application displaying the monitored video stream with detection overlays and behavioral status indicators.

VII. Conclusion

This paper presents a modular deep learning approach for real-time surveillance that integrates YOLOv8, EfficientNet-B0, and an attention-enhanced bidirectional GRU to detect both crowd anomalies and visible weapons. The proposed architecture achieved a validation accuracy of **86.4%** and an F1-score of **0.91** on the UCF-Crime dataset, demonstrating its effectiveness and highlighting the contribution of the attention mechanism in the temporal modeling stage. A prototype real-time system was also deployed using a Flask-based application with monitoring, alerting, and logging functions, confirming the practical feasibility of the method.

Although certain limitations remain, particularly with respect to performance under challenging environmental conditions and the recognition of threat types not represented in the training data, the proposed approach establishes a solid foundation for automated surveillance systems that support public safety. Future work will focus on improving robustness in adverse scenarios, expanding the range of detectable threats, and integrating explainability techniques to facilitate a better understanding of model decisions.

Overall, this study presents a scalable framework for proactive security applications through the integration of object detection and sequential behavioral analysis. By enhancing situational awareness, the proposed approach contributes toward the development of more intelligent and responsive surveillance infrastructures for urban environments.

References

[1] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[4] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7473.

[5] Ultralytics, "YOLOv8," *https://github.com/ultralytics/ultralytics*, 2023.

[6] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[8] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv preprint arXiv:1409.0473*, 2014.

[11] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

[12] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[13] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6479–6488.

[14] G. Pang, C. Shen, L. van den Hengel, and A. R. Dick, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[15] R. Nayak, U. C. Pati, and S. K. Das, "A comprehensive review on deep learning-based methods for video anomaly detection," *Image and Vision Computing*, vol. 106, p. 104094, 2021.

[16] V. Mahadevan *et al.*, "Anomaly detection in crowded scenes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1975–1981.

[17] M. Hasan *et al.*, "Learning temporal regularity in video sequences," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 733–742.

[18] C. Lu, J. Shi, and J. Jia, "Abnormal Event Detection at 150 FPS in Matlab," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2013.

[19] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[20] R. Girshick *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[21] C. Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," *Distill*, 2017.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[23] F. P. Garcia *et al.*, "A survey on attention mechanisms for medical image analysis," *Computers in Biology and Medicine*, vol. 138, p. 104894, 2021.

[24] C. A. Ververidis, G. S. Andreadis, and N. S. Tselios, "A Comprehensive Review of YOLO Architectures in Computer Vision," *Electronics*, vol. 12, no. 17, p. 3676, 2023.

[25] D. Tran *et al.*, "Learning Spatiotemporal Features with 3D Convolutional Networks," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2015.

[26] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.