رَقِيب
Anomaly detection in surveillance videos based on deep learning

Deema Alhawsa, Mona Alrougi, Norah Almutiri, Rahaf Alrouqi, Sara Alamer
Data Science and Artificial Intelligence (T5) Bootcamp,
Saudi Data and AI Authority (SDAIA),
Riyadh, Saudi Arabia

***Abstract*—Surveillance videos offer a rich source of realistic anomalies. This project presents the development of a real-time surveillance video anomaly detection system using the ResNet model capable of identifying various unusual activities, such as fighting and theft. By automatically detecting anomalies, the system dramatically improves monitoring efficiency and response times, contributing to safer public spaces and safer communities. This aligns with and supports the objectives of Saudi Vision 2030, particularly in enhancing public safety and security. Our approach treats normal and anomalous videos as bags and segments within videos as instances in multiple-instance learning (MIL). Our findings confirm the system's effectiveness in accurately identifying anomalies with an (AUC) value of 85%.**

I. Introduction

With the growing need for enhanced security to safeguard lives and public property, surveillance cameras are increasingly deployed in various public spaces, including markets, shopping malls, hospitals, banks, streets, and educational institutions. The primary goal of this task is to monitor daily activities and detect anomalous events early. Anomalies in videos refer to events or behaviors that are out of the ordinary and indicate abnormal behavior [1], such as fights, car accidents, crimes, or illegal activity.

Detecting anomalies in video is a critical task in many cases where human intervention is necessary to prevent crime. Nonetheless, this process demands human effort and constant monitoring, which is a tedious process, as abnormal events occur only 0.01% of the time, resulting in 99.9% of surveillance time being wasted [2]. Additionally, surveillance systems generate a large amount of redundant video data, requiring unnecessary storage space. Therefore, to reduce the waste of labor and time, there is an urgent need to develop intelligent computer vision algorithms for automatic video anomaly detection.

Recently, this problem has garnered significant attention in computer vision research. Numerous researchers have sought to determine the best method for accurately detecting anomalies in video streams while minimizing false alarms. The outcomes demonstrated that deep learning-based methods provide highly intriguing outcomes in this field. Therefore, in this work, we proposed a real-time video anomaly detection model leveraging deep learning techniques, specifically employing ResNet.

A. *Problem Statement*

Surveillance cameras are everywhere, but there aren't enough human monitors to watch all the footage effectively. This makes it hard to catch unusual events like accidents or crimes. Current methods to detect these events usually depend on knowing what normal behavior looks like, but it's tough to define normal behavior in all situations. These methods also tend to give a lot of false alarms. So, there's a need for developing intelligent, minimally supervised computer vision algorithms that can automatically and accurately detect unusual events in video footage with little human help.

B. *Project Scope and Objectives*

The scope of our project will focus on:

- Using training videos that are weakly labeled, where only the overall video is labeled as either normal or containing an anomaly, but the specific location of the anomaly within the video is not known.

- Implementing a multiple instance learning (MIL) framework to process these weakly labeled videos.
- Designing a deep learning model that can learn to identify and rank anomalous segments within a video.
- Ensuring the system can handle diverse and changing environments captured by surveillance cameras, reducing false alarm rates.

The objectives of our project include:

- Develop a deep learning framework to detect anomalies in surveillance videos using weakly labeled data.
- Enhance the system's ability to accurately identify anomalies while reducing false positives.
- Design the system to detect a wide range of anomalous events without needing specific models for each type of event.

II. RELATED WORK

In recent studies, several methods have been presented for video anomaly detection, which use deep learning techniques to improve the detection accuracy. This literature review examines the application of deep learning techniques to video anomaly detection, specifically focusing on the UCF dataset, a widely used benchmark in this field.

Zaheer et al. [3] introduced a weakly supervised learning method, using video-level labels, for anomaly detection. Employing batch-based training with randomly selected segments improved performance by reducing inter-batch correlations. Their approach included a normality suppression technique to emphasize anomalies by training the network to suppress features of normal segments. Furthermore, they introduced a clustering distance-based loss to enhance the network's representation of both normal and abnormal events. The authors tested this method on the UCF-Crime and ShanghaiTech databases with an AUC value of 83.03% for the UCF crime database.

Similarly, Zhong et al. [4] introduced a new approach to weakly supervised anomaly detection by framing it as a supervised learning task with noisy labels. They employed a graph convolutional network (GCN) to clean the labels, which were then used to train an action classifier. This method was tested on the UCF-Crime, ShanghaiTech, and UCSD-Peds2 databases, achieving an AUC value of 82.12% on the UCF-Crime dataset.

Hao et al. [5] proposed a two-stream convolutional network model that integrates flow and RGB (red-green-blue) networks. The final anomaly activity recognition score is derived from the combined scores of these two streams. Anomaly detection is treated as a regression problem. Their experiments on the UCF-Crime dataset resulted in an AUC value of 81.22%. Similarly, Dubey et al. [6] introduced a deep network with multiple ranking methods (DMRMs) for detecting anomalies. They also treated anomaly detection as a regression problem, using 3D ResNet-34 for this task. Their method, tested on the UCF-Crime dataset, achieved an AUC value of 81.91%.

Majhi et al.[7] proposed a method using a weakly supervised learning model to handle abnormality detection and classification in a unified model. They implemented an I3D many to many LSTM approach for outlier detection. This model was evaluated on the UCF Crime dataset, achieving an AUC value of 82.12%. Zaheer et al. [8] offered a weakly supervised technique for abnormality identification that uses video level labels to train feature extractor models such as Convolutional 3D (C3D), k-means, or fully connected networks. This approach was evaluated on the UCF Crime dataset and achieved an AUC of 78.27%.

III. Data Description and Structure

In this project, we used a large-scale dataset called the UCF-Crime dataset, specifically constructed to evaluate anomaly detection methods in surveillance videos provided by Soltani et al. [2]. The dataset includes 13 types of real-world anomalies, selected for their significant impact on public safety: Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. Videos were sourced from YouTube and LiveLeak using text search queries. The dataset consists of 1900 videos in total. Among these videos, 950 contain clear anomalies videos, while the rest are considered normal. Figure 1 depicts a sample of anomalies from the UCF dataset.

As for the challenges we faced while dealing with this dataset, we didn't encounter any issues. However, during the download process, it took a lot of time due to the large size of the videos, which were long and untrimmed.

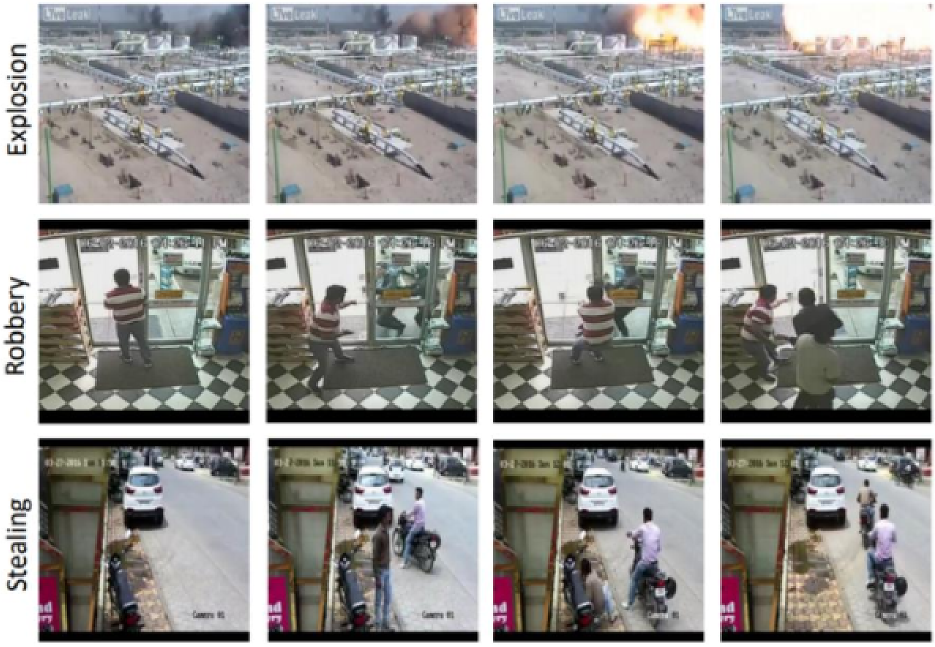

Fig. 1: Samples of anomalies from the UCF dataset

IV. Proposed Anomaly Detection Methodology

In this section, we present the methodology used in this work. As mentioned previously, this work aims to develop a real-time video anomaly detection system capable of identifying unusual activities, such as fighting and stealing, and issuing alerts to inform the relevant authorities. The details of the methodology are illustrated below.

A. *Dataset*

The dataset used in this project is detailed in the previous Section III.

1) Dataset Preprocessing: This phase aims to prepare our dataset for anomaly detection, we implemented a comprehensive preprocessing pipeline. Initially, each video was converted into a series of static frames at a fixed frame rate to facilitate frame-by-frame analysis and ensure essential event details were captured. These frames were then resized to uniform dimensions (320x240 pixels) to facilitate consistent input for the neural network. After that, feature extraction techniques were applied to transform the frames into numerical representations and tensors, capturing significant visual characteristics (colors, edges, textures) and landmarks (behavior). Finally, frames from each video were aggregated to maintain temporal context and sequence of events, ensuring the dataset was both consistent and rich in informative features for the anomaly detection model.

B. *Model Selection*

We used a ResNeXtBottleneck pre-trained model combined with a custom classifier (Learner model) to detects anomaly in videos. ResNeXtBottleneck is pre-trained on UCF-Crime datasets , which allows it to learn rich feature representations. This pre-training helps the model extract meaningful features from new datasets with fewer training samples. Additionally, ResNeXtBottleneck's deep and wide architecture, which includes multiple paths for feature extraction, enables it to capture a variety of features at different levels of abstraction. This flexibility means the output features from ResNeXtBottleneck can be easily adapted to various downstream tasks without retraining the entire network. ResNeXtBottleneck has also demonstrated strong performance on various benchmarks, making it a reliable choice for feature extraction.

The custom Learner model, a simple feedforward neural network, takes the features extracted by ResNeXtBottleneck and performs the final classification. Its simplicity makes it easy to customize and tune for specific tasks. Including a dropout layer helps regularize the model, reducing the risk of overfitting, especially with limited data. By training only the Learner model and keeping the feature extractor frozen (or fine-tuning only the last few layers), we can achieve good performance with reduced computational resources and training time. This combination leverages the strengths of both models, providing an efficient solution for video anomaly detection.

C. Evaluation Metric

To evaluate the effectiveness of our approach, we use frame-based receiver operating characteristic (ROC) curves and associated area under the curve (AUC) based on previous work in anomaly detection. Also, we use the F1 score to check the effectiveness of our model.

V. Experiments

In this section, we will explain the details of our experiments to build a video anomaly detection system. This section consists of two subsections. Each subsection discusses our different experiments.

A. First Experiment

This experiment was the first attempt to build an anomaly detection and classification system for real-world scenarios. The model used weakly supervised learning and relied solely on video-level labels rather than dense temporal annotations for the learning process.

1) Data Loader: There are two data loaders responsible for loading training data for a Multiple Instance Learning (MIL) framework: DataLoader_MIL_train and DataLoader_test_detect. These classes take annotation files that contain the video name, the actions that the video represents, and numbers specifying when the action is represented in a segment of the video. For example:

Abuse028_x264.mp4 Abuse 165 240 -1 -1

In this example, '-1' refers to a normal event. The data loaders extract both the video name and the action. In the case of DataLoader_test_detect, it determines the number of batches by dividing the total data length by the segment_size, which are used in the model.

2) Model Architectures:

- First Model: model_LSTM_RGB: This function defines an LSTM-based model with residual connections for processing RGB data sequences.
- Second Model: model_attn_RGB: This function defines a model with an attention mechanism for processing RGB data sequences.
- Third Model: Joint_model: This function defines a joint detection and classification model and combines the results.

The final result is a model with inputs from both detection and attention parts and outputs from both detection and classification parts. The model is compiled with binary cross-entropy loss and accuracy metrics.

Issues Faced: The batch size of the architecture caused problems for the model during testing. The model failed to recognize different video batches (segments), which varied according to each video and its range. A solution was tested by making the segments of the video static, which enabled the model to work but resulted in overfitting, with both detection part accuracy and classification part accuracy reaching 100% from the first epochs.

We attempted to optimize the model by using dropout layers, changing the segment sizes to different ranges, and increasing the training data. However, due to lack of time, we decided to stop fixing the code and shifted the focus to only anomaly detection.

B. Second Experiment

In the second experiment, we will discuss the data loader and multiple instance learning (MIL) as well as the details of the learner model, describing the training stage and evaluation.

1) Data Loading: The Normal_Loader and Anomaly_Loader classes are designed to manage the loading and preprocessing of video data categorized as normal and anomaly, respectively. During initialization, these classes load a list of video files based on a specified flag indicating whether it's for training or testing purposes. For testing, the list is shuffled and truncated as needed for validation purposes. When loading data, these loaders retrieve RGB and flow features for each video and concatenate them appropriately, ensuring that the data is prepared for further processing and analysis. These classes streamline the handling of different types of video data, facilitating efficient training and evaluation of models for anomaly detection tasks.

2) Multiple Instance Learning (MIL): Multiple Instance Learning (MIL) Loss Function: The MIL function implements the Multiple Instance Learning loss calculation. This loss function is designed to distinguishes between anomaly and normal instances within each batch and calculates a loss that encourages the model to separate anomaly scores from normal scores. To compute the MIL for anomaly detection, we distinguish between anomaly and normal instances within each batch. The loss function encourages the model to separate anomaly scores from normal scores effectively. The initial loss is calculated using the maximum scores for both anomaly and normal instances, with sparsity and smoothness penalties added. Finally, the average loss across the batch is computed. The final equation for the average loss is:

$$L = \frac{1}{N} \sum_{i=1}^{N} (\max\big(0, M_{a,i} - M_{n,i}\big) + P_{s,i} + P_{m,i})$$

3) Learner Model: Training and Evaluation Details: The Learner class is responsible for implementing a neural network classifier with a sequential architecture comprising linear layers, ReLU activations, dropout regularization, and a final sigmoid activation. During initialization, it sets up the classifier architecture and initializes the weights using Xavier normal initialization, ensuring a balanced initialization for effective training. The class collects the parameters of the classifier within self.vars for easy management and access during training. In the forward pass, the Learner class defines the sequence of operations using the collected parameters, sequentially applying linear transformations, ReLU activations, and dropout regularization. Finally, it returns the output after passing through the sigmoid activation function, providing a streamlined approach to neural network classification tasks.

During training iterations, the process begins with mini-batch training, where the dataset is divided into smaller batches containing bags with their respective instance features and bag-level labels, facilitating efficient training. Detailed steps include input preparation by loading preprocessed features and labels, creating mini-batches, forwarding instance features through the neural network model to obtain predictions at both instance and bag levels, calculating the MIL loss function based on these predictions to guide the model towards accurate bag-level predictions and anomaly identification within bags. Backpropagation propagates gradients to update the model's parameters, and training proceeds through multiple epochs, with monitoring of metrics like loss and accuracy for convergence assessment. Validation on a separate set helps prevent overfitting, while loading a pre-trained model checkpoint and transitioning models to evaluation mode ensure accurate inference. Image processing through transforms, model prediction, and metric calculation like FPS and anomaly counts further refine the training process.

During the evaluation process, several steps were taken to evaluate the performance of our model. The initial step involves visualizing the frames along with frames per second (FPS) and prediction details, saving these frames for further analysis, and generating an output video. The percentage of abnormal frames is calculated to determine the overall classification as "Abnormal" or "Normal. In addition, we used the receiver operating characteristic (ROC) curve to evaluate our model's performance. The ROC curve illustrates the trade-off between a true positive rate (sensitivity) and a false positive rate (1 - specificity) across different thresholds. A higher area under the ROC curve (AUC) generally indicates better class discrimination. A well-separated ROC curve with a high AUC suggests the model's effective differentiation between normal and abnormal frames. However, to comprehensively evaluate the model, we used other metrics like F1-score to check the effectiveness of our anomaly detection system.

VI. Results and Discussion

This section presents the project's results, comparing them with previous research, and discussing their implications and significance, particularly in achieving Saudi Vision 2030 objectives.

We can see that our approach showed an accuracy of 84% for the UCF-Crime dataset, as indicated by the receiver operating characteristic (ROC) curve in Figure 2. This represents a significant improvement over previous studies in the field of anomaly detection. The 84% accuracy highlights the effectiveness of our model in classifying events. Additionally, our model achieved an F1 score of

85%, further confirming its reliability and precision in detecting unusual activities, as shown in Figure 3.

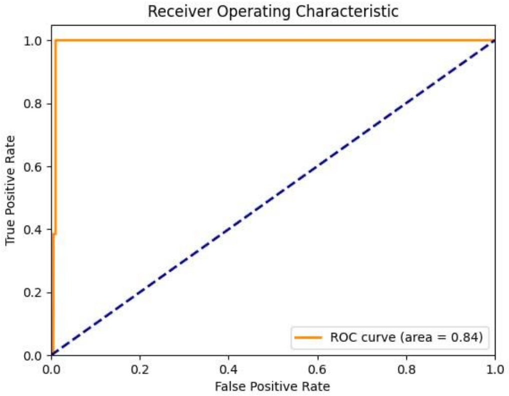

Fig. 2: ROC-curve

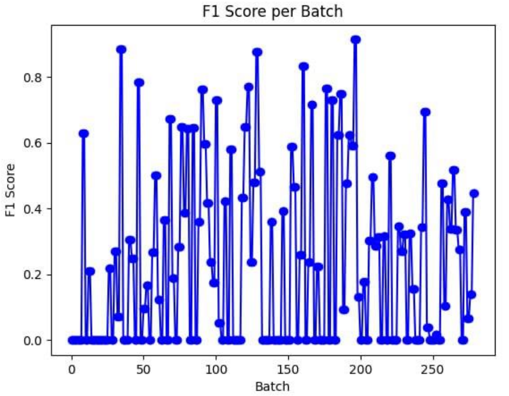

Fig. 3: F1-Score

The significant improvement in accuracy and F1 score shows that our model is robust and can be used in real-world surveillance systems. These results indicate that the system can effectively enhance public safety and security by accurately and promptly detecting anomalous activities, thus supporting the goals of Saudi Vision 2030 to create safer communities.

VII. Conclusion AND Future work

In this project, we successfully developed a real-time surveillance video anomaly detection system capable of identifying unusual activities such as fighting and theft. The system's ability to immediately detect these anomalies and alert the relevant authorities has significant implications for enhancing security and safety in various environments. Our findings demonstrate the effectiveness of the applied techniques and ResNet models in accurately detecting anomalies. The practical applications of this system are wide-ranging, including deployment in public places, transportation hubs, and schools to improve monitoring and response times to critical incidents. By automating the detection process, the system can help security personnel monitor large areas more efficiently and effectively, reducing reliance on manual monitoring and potentially increasing the rate of incident prevention and intervention.

Future work could further enhance the system by incorporating deep reinforcement learning techniques and expanding the range of detectable anomalous behaviors. Additionally, integrating the system with real-time alert mechanisms, such as linking to authorities' Gmail accounts, would further augment its practical utility and impact. Furthermore, the system can be enhanced to recognize each of the 13 types of anomalous activities.

References

[1] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *CVPR 2011*, pp. 3313–3320, IEEE, 2011.

[2] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6479–6488, 2018.

[3] M. Z. Zaheer, A. Mahmood, M. Astrid, and S.-I. Lee, "Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pp. 358–376, Springer, 2020.

[4] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1237–1246, 2019.

[5] W. Hao, R. Zhang, S. Li, J. Li, F. Li, S. Zhao, W. Zhang, *et al.*, "Anomaly event detection in security surveillance using two-stream based model," *Security and Communication Networks*, vol. 2020, 2020.

[6] S. Dubey, A. Boragule, J. Gwak, and M. Jeon, "Anomalous event recognition in videos based on joint learning of motion and appearance with multiple ranking measures," *Applied Sciences*, vol. 11, no. 3, p. 1344, 2021.

[7] S. Majhi, S. Das, F. Brémond, R. Dash, and P. K. Sa, "Weakly-supervised joint anomaly detection and classification," in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pp. 1–7, IEEE, 2021.

[8] M. Z. Zaheer, J.-h. Lee, M. Astrid, A. Mahmood, and S.-I. Lee, "Cleaning label noise with clusters for min-

imally supervised anomaly detection," *arXiv preprint arXiv:2104.14770*, 2021.