Deep Feature-based Anomaly Detection for Video Surveillance

Parth Lalitkumar Bhatt
Dept. of Computer Science
Lakehead University
Thunder Bay, Canada
bhattp@lakeheadu.ca

Dhruva Shah
Dept. of Computer Science
Lakehead University
Thunder Bay, Canada
dshah33@lakeheadu.ca

***Abstract*—Detecting anomalies in video surveillance is a challenging task that requires distinguishing between normal and abnormal behaviour. Video surveillance systems not only face challenges in identifying and monitoring unusual human actions but also in differentiating normal from anomalous actions due to a large amount of data in video format. This study consists of a proposal for an intelligent video surveillance system that utilizes deep feature-based anomaly detection to identify anomalous events in a video stream. Our approach uses a two-stream deep learning model with I3D as the feature extraction component, which has demonstrated effectiveness in action recognition and detection tasks. Thus evaluated proposed system on the UCF Crime dataset, consisting of videos of normal and abnormal occurrences, and achieve an AUC of 87.52%. Our results demonstrate the efficacy of the proposed method in identifying anomalous events and show significant improvement over state-of-the-art methods.**

***Index Terms*—Anomaly detection, Deep learning, Multiple Instance Learning, Video surveillance.**

I. Introduction

Detecting anomalies in videos is a challenging task due to the ambiguous nature of what constitutes an anomaly, and the significant amount of data in video format. However, with an upsurge in crime, there is a growing need for smart surveillance systems that can detect and respond to anomalous events in real-time. Delays in reaction from relevant authorities increase the amount of loss of life and property, emphasizing the importance of early detection and response. The video analytics market is expected to grow at a Compound Annual Growth Rate (CAGR) of 21.5%, from USD 3.23 billion in 2018 to USD 8.55 billion by 2023[1]. In recent years, deep learning-based video anomaly detection has emerged as a promising solution to this problem. However, there are various technical challenges, such as ambiguous data quality and quantity, model selection, computational complexity, and real-time processing, that make it difficult to accurately detect anomalies using models. Therefore, deep learning models for video anomaly detection should be interpretable, robust, generalizable, and designed with privacy in mind. To overcome these challenges, innovative approaches such as data augmentation, transfer learning, and advanced model architectures are necessary to develop accurate and robust models that can generalize to novel scenarios and environments, enabling smart surveillance systems to enhance public safety and security in various domains.

The objective of this research is to enhance the detection of anomalies in video surveillance by implementing a deep feature-based methodology. Our approach uses a two-stream deep learning model with Inflated 3D Convolutional Neural Networks (I3D) [1] as the feature extraction part and the UCF Crime dataset for training. Specifically, the I3D model is pre-trained on large-scale action recognition datasets and fine-tuned on the UCF Crime dataset, which contains normal and abnormal activities. The project consists of the use of Multiple Instance Learning (MIL) loss function [2] with a threshold to classify video segments as normal or abnormal. Our proposed method achieves an Area Under the Curve (AUC) of 87.52% and demonstrates the effectiveness of the I3D two-stream architecture for video anomaly detection. And also visualize the output using a heat map and a graphical representation of the temporal analysis of anomaly prediction on videos. Our experimental results show a significant improvement in anomaly detection performance compared to state-of-the-art methods. Our abstract level flow diagram is shown in Fig. 1

A. Organization

The content is organized into four main sections: Introduction, Related Work, Proposed Method, and Experimental Analysis. The Introduction section provides an overview of the problem statement and motivation, the Literature Review section presents background information about all existing research and the segregation of different types, the Proposed Method section outlines the methodology of the research, and the Experimental Analysis section provides the results of the experimental study.

II. Literature Review

Traditional anomaly detection systems define the problem as modelling normalcy given a large number of normal samples and declaring anomalies based on deviations from normality. Early research attempts to learn a discriminative decision boundary using hand-crafted characteristics. The main distinction between these existing methods is how anomalies

[1]https://www.marketsandmarkets.com/Market-Reports/intelligent-video-analytics-market-778.html

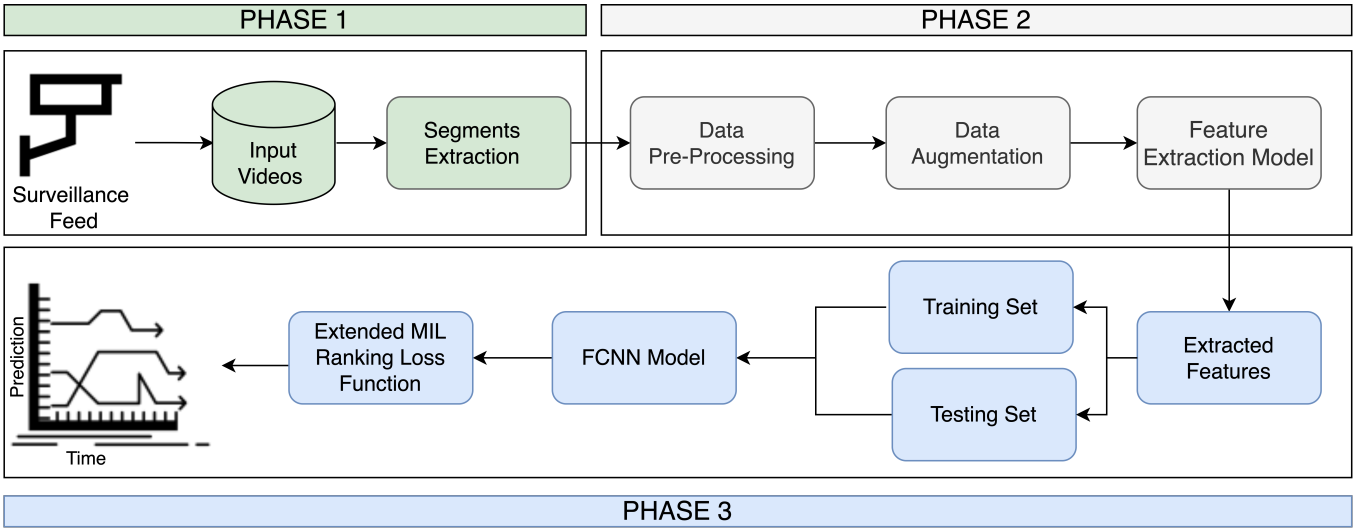


Fig. 1: Complete Flow Diagram of Proposed Solution. It consists of three phases, **Phase 1**: Input, **Phase 2**: Feature Extraction and **Phase 3**: Classification.

are distinguished from normal scenarios. When an element or event displayed in a video or video frame differs considerably from those learned from the training set, it is often treated as an outlier.

Here, provide an extensive literature review, which is categorized into 7 different learning techniques, to comprehensively cover the state-of-the-art in the field and they are as follows:

1) ***Transfer Learning****:* Waqas Sultani *et al.*[2] created a method for classifying anomalous and normal occurrences by leveraging the usage of poorly annotated videos and Multiple Instance Learning algorithms (MIL). It creates bundles of videos from the same class. The videos have been divided into instances. During training, an anomaly ranking model is created, and instances in the video are assigned a rank based on the abnormality. Anomaly is found because abnormal segments are assigned a higher score. Nazare *et al*[3] investigated the utility of pre-trained CNNs in anomaly detection, including VGG-16, ResNet-50, Xception, and DenseNet-121.It looked at the importance of pre-trained image classifiers in feature extraction to tackle it. Nguyen and Meunier [4]use a Conv-AE with an Inception Module to create a deep autoencoder that recognises appearance and motion characteristics from videos. The model's decoder contains two units dedicated to motion and appearance.

2) ***Ensemble Learning****:* Other models under consideration include random examples of ensemble learning, which mixes many learning algorithms to achieve higher prediction performance than the constituent learning algorithms alone. For example, Zahid *et al.*[5] is a classic example of ensemble and transfer learning. The model incorporates a 3D convolutional network as well as a Fully Connected (FC) Network. Another example of ensemble learning is Vu*et al.* [6], which integrates Conditional Generative Adversarial Networks, R-CNN, and Support Vector Machines (SVM).

3) ***Continual Learning****:* Continual Learning is a never-ending learning system that gradually reinforces previously learned knowledge. To address the catastrophic forgetting problem, a strategy to regularise the whole network was presented. Zhou *et al.*[7] use a feature learning subnetwork to combine motion and appearance features, while Xu *et al.*[8]use AICN for anomaly detection and localization. Doshi *et al.*[9] use a feature extraction module and a statistical decision-making module to incrementally update the learned model. Yu *et al.*[10] collect the hard normal instances from previous training epochs in order to perform cross-epoch learning, with the validation loss proposed to suppress the anomaly scores of these hard normal instances. Furthermore, it proposes a dynamic margin loss function to realize the anomaly score margin between hard normal instances and the most deviant instances in abnormal bags, with the margin increasing progressively during the training process.

4) ***Reinforcement Learning****:* Reinforcement Learning is a sequential decision-making system that uses an agent to make choices. To resolve this, Sabzailan *et al.*[11]combined the spatiotemporal convolution neural network (CNN) with handcrafted feature sets such as histograms of optical flow (HOF) and histograms of oriented gradients (HOG). Sultani *et al.*[2] proposed Multiple Instance Learning, which assigns a rating score for normal and abnormal examples using a C3D pre-trained model mixed with a light classifier. Aberkane and Elarbi *et al.*[12] identified abnormalities in videos using a Deep Q Learning Network (DQN), which is made up of a completely linked layer that estimates the probability of each video clip in the anomalous and normal bags.

5) ***Representative Learning****:* Representation learning is a collection of techniques that allow a system to automatically identify the representations required for feature detection or classification from raw data. Hu *et al*[13]. proposed a deep incremental slow feature analysis (D-IncSFA) network

which is applied directly in learning progressively abstract and global high-level representations from raw data sequence. Kavikul [14] leveraging proposes a CNN architecture to learn powerful features from weakly labelled data and the connectivity in architecture is able to learn spatial features. The main merit of the D-IncSFA network is that it has feature extractor and anomaly detector functionality, allowing anomaly detection in a single step, but it fails to identify local anomalies.

Zhu *et al.*[15]presented temporal enhanced MIL ranking loss that takes into account the temporal context via an attention method. To capture long-range relationships in reliable anomaly identification, Wu *et al.*[16] proposed a causal convolution for feature extraction. Despite convincing findings, their effectiveness is restricted by inaccurate classification scores resulting from the weak supervisory signal.

To enhance the MIL framework, Gong *et al.* [17] suggest a Multi-scale Continuity-aware Refinement network (MCR). MCR calculates anomaly score vectors for cases at multiple temporal scales using varying window widths, capturing continuity through time. These vectors are then merged based on attention weights gained throughout the training procedure.

6) ***Generative Learning****:* Generative models are probabilistic models that specify how a dataset is formed and can produce fresh data by sampling from it. Auto-encoders are used to identify abnormalities in videos, and M. Hasan *et al.*[18] uses a 2D convolutional autoencoder to model regular frames. Medel proposed an End-to-end trainable composite Convolutional Long Short-Term Memory (Conv-LSTM) network to predict the evolution of a video sequence from a small number of the input frame. However, this composite model suffers from a few challenges, such as not adapting to new (unusual) movements and overtime errors in object prediction.

The Convolutional Autoencoder (CAE) is an interesting choice for anomaly detection, as it captures the 2D structure in image sequences during the learning process. Ionescu *et al.*[19] developed an unsupervised feature learning framework based on object-centric convolutional auto-encoders to encode both motion and appearance information. The Spatial-temporal autoencoder by Chong and Tay [20] is different due to its building constructs. Sabokrou *et al.*[21] proposed a two-stage cubic-patch-based cascade classifier to perform anomaly detection and localization of anomalous regions, with these two stages built on analyzing the Reconstruction error (RE) and Sparsity Value (SV). This method is a quick, simple, and accurate method for locating abnormal events in the video.

The autoencoder developed by Duman and Erdem [22] is made up of Convolutional Autoencoder and Convolutional LSTM, which extracts speed and trajectory characteristics from movies using Optical Flow. Ramchandran and Sangaiah's [23] unsupervised solution for anomaly detection in crowded scenes uses Conv-LSTM, Bhakat and Ramakrishnan *et al.*[24] uses a spatial-temporal autoencoder, Pawar *et al.*[25] proposes an unsupervised approach learning approach based on deep learning and one class learning paradigm for the detection of global anomalies from crowd surveillance videos, Nguyen [4] proposes a novel method based on the GAN approach to detect abnormal events in traffic surveillance videos, and Sabokrou *et al.*[26] proposes a cubic-patch-based cascade classifier to improve the computational time required for anomaly detection.

7) ***GNN Models****:* Despite convincing conclusions, their effectiveness is hampered by untrustworthy classification scores resulting from the weak supervisory signal. To address this issue, Zhong *et al.*[27] proposed employing graph convolutional neural networks to improve the noisy prediction using video-level labels.

The current research on video anomaly detection has several limitations. Firstly, some approaches either ignore valuable optical flow information or use datasets with limited number of anomalies and data. Secondly, the employed loss functions are not suitable for perfect distinction between normal and anomalous events. Thirdly, many existing methodologies include models with high time complexity, which is unsuitable for real-time video anomaly detection. To overcome these limitations, new methods should be developed that incorporate optical flow information, use large datasets for training, and develop novel loss functions or modify existing one. Additionally, time-efficient models should be used. Addressing these limitations has the potential to significantly improve video anomaly detection.

III. Methodology

The proposed approach is divided into 3 phases as described in Fig. 2 and the discussion regarding each phase is as follows:

A. *Phase 1*

1) ***Input****:* UCF-Crime [2] is a large-scale dataset with 1900 untrimmed surveillance videos, each lasting 128 hours, from indoor and outdoor cameras. The dataset contains an equal number of normal and anomalous videos in both the training and testing sets. The dataset includes 13 different anomalies such as abuse, arrest, arson, assault, road accident, burglary, explosion, fighting, robbery, shooting, stealing, shoplifting, and vandalism. Sultani *et al.* [2] proposed this dataset to overcome the limitations of previous datasets used for video anomaly detection, which lacked diversity in the types of anomalies and the number of videos.

The dataset contains only frame-level annotations gathered by multiple annotators, and only the testing set has these annotations. The training set has 800 normal and 810 anomalous videos, while the testing set has 150 normal and 140 anomalous videos, with all 13 anomalies occurring in various temporal locations, and some videos containing multiple anomalies.

B. *Phase 2*

1) Pre-Processing:

- **Resizing**: The first step is to resize the input video frames to a fixed size to ensure uniformity across all frames.
- **Center and Random Cropping**: The second step is to perform cropping on the video frame, where in center

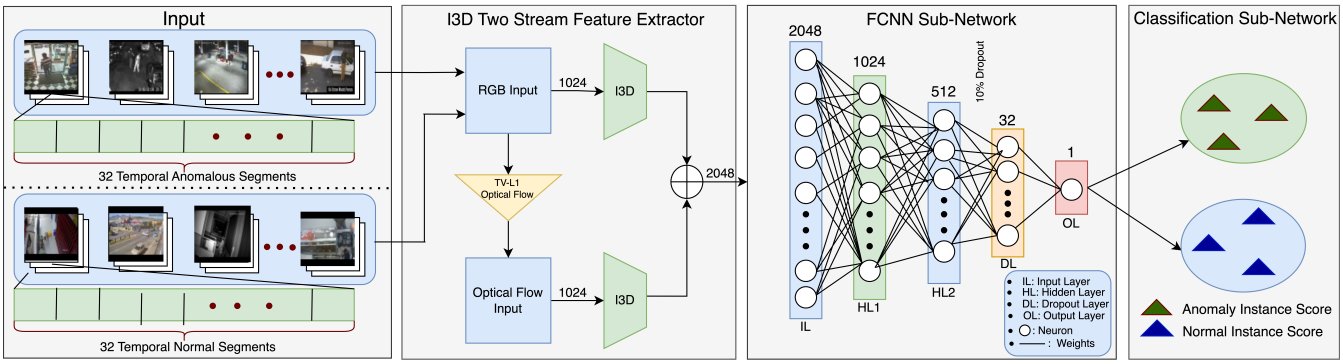


Fig. 2: A complete pipeline of the proposed solution in a step-wise manner, Phase 1 represents input sub-network, where inputs from normal and anomalous videos are divided into 32 segments and are provided as an input to Phase 2, which is a two-stream I3D feature extractor, where both RGB frames and optical flow information of the video is utilized to extract features and provided to the 5 layered fully connected neural network, which uses modified MIL ranking loss function in Phase 3 perform classification.

cropping the center of the video frame is cropped out, and the remaining part is used for processing and random cropping involves randomly selecting a section of the image or video and cropping it out to create a new image or video sample

- **Normalization**: The next step is to normalize the pixel values of the cropped video frames. This is done to ensure that the data has zero mean and unit variance, which helps in better convergence during training.
- **RGB to BGR Conversion**: In the two-stream I3D model, the input frames are expected to be in the BGR format, which is different from the RGB format used by most video codecs. Therefore, the RGB frames are converted to BGR format before feeding into the model.
- **Optical Flow Computation**: For the second stream of the two-stream I3D model, optical flow computation is performed on the input video frames. Optical flow is the pattern of apparent motion of objects in a video, which is computed by comparing adjacent frames. It helps in capturing the temporal information of the video.
- **Optical Flow Normalization**: Finally, the optical flow values are normalized to have zero mean and unit variance, similar to the RGB frames, to ensure consistency between the two streams.

2) Feature Extraction: Raw videos can't be used for anomaly prediction, and hence the important features need to be extracted from them. For this purpose, the Two-Stream Inflated 3D ConvNet (I3D) [1] has been utilized, which is based on 2D ConvNet inflation. I3D allows the learning of spatiotemporal features from video by extending successful ImageNet architecture designs and parameters. The I3D model employs a two-stream architecture, where one stream handles RGB frames and the other handles optical flow frames, as shown in Fig. 3. The model has 25 layers, including 9 Inception modules, which are used for efficient feature extraction by concatenating multiple filter sizes in parallel. The I3D model has been pre-trained on Kinetics and outperforms the state-of-the-art methods in action classification. The novel part of the approach is that the modified two-stream I3D model has been used for Feature extraction for video anomaly detection on the UCF-Crime dataset. The original implementation was done in TensorFlow [2], while there has been modification and implementation in PyTorch [3].

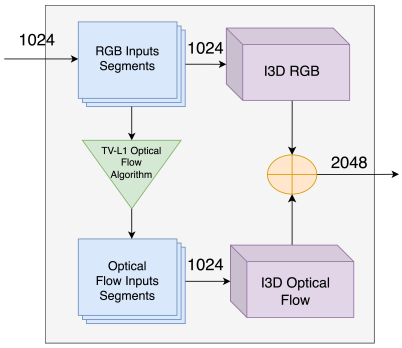


Fig. 3: A brief overview of two stream feature extraction process, where the RGB frames of 1024 dimension plus the optical flow information of also 1024 dimension, which is computed using the TV-L1 optical flow algorithm[28], where RGB frames are provided as input. At last, both of them are concatenated and provided as input of dimension 2048 to the FCNN.

C. *Phase 3*

1) Model Architecture: A video is divided into 32 segments and treated as bag instances just like Sultani et. al. [2]. Overlapping temporal segments of different scales were ex-

[2]https://www.tensorflow.org
[3]https://pytorch.org/

perimented with, but they did not enhance detection accuracy. A minibatch of 30 positive and 30 negative bags is randomly selected.

As clearly described in Fig. 4, extracted features of 2048-Dimension are provided as input to a 5 Layer Fully connected neural network. The first 4 layers are followed by the ReLU Activation function [4], as described in Eq. (1) to introduce non-linearity and avoid the vanishing gradient problem. The last layer is followed by the Sigmoid Activation Function, as described in Eq. (2) to indicate normal and abnormal inputs, with a smooth gradient that makes it suitable for backpropagation algorithms. 10% Dropout Regularization is used in the 4^{th} layer to prevent overfitting.

$$f(x) = \max(0, x), \tag{1}$$

where x is the input to the activation function and f(x) is the output.

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2}$$

where x is the input to the function and it can take any real value, and the function will output a value between 0 and 1.

The Xavier initialization method is used to initialize the weights of the linear layers. It helps to keep the variance of activations and gradients consistent across layers, preventing issues like vanishing and exploding gradients which can lead to unstable training. The Xavier initialization formula, represented in Eq. (3) ensures that the weights are initialized to values that have a variance of $1/n$ and $1/m$ for the input and output layers, respectively

The Xavier initialization formula for a layer with n inputs and m outputs is:

$$W \sim U\left(-\frac{\sqrt{6}}{\sqrt{n+m}}, \frac{\sqrt{6}}{\sqrt{n+m}}\right), \tag{3}$$

where W is the weight matrix and U is a uniform distribution. The formula ensures that the weights are initialized to values that have a variance of $1/n$ and $1/m$ for the input and output layers, respectively. This helps to prevent the gradients from vanishing or exploding during training.

TABLE I: Hyperparameter Table

Hyperparameter	Value
Optimizer	Adam
Batch size	30
Loss function	MIL
Learning rate	0.001
Weight decay	1×10^{-12}
Number of epochs	100

2) Model Training and Hyperparameters: Table I shows various hyperparameters used in our approach and their significance is as follows:-

[4]https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html

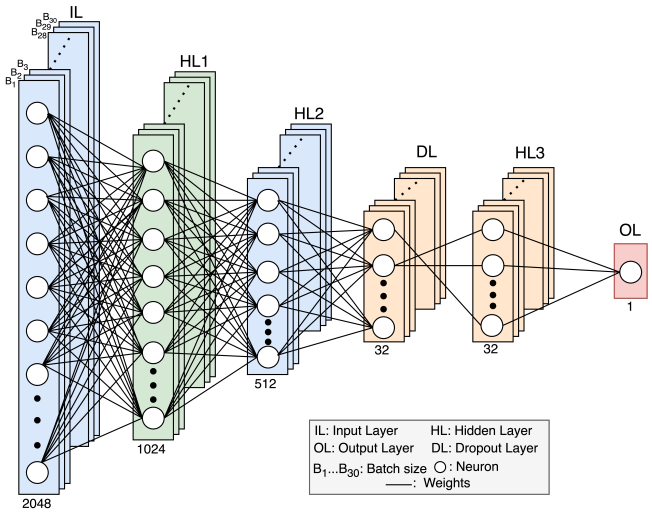


Fig. 4: Detailed layer-wise schematic of proposed 5-layered fully connected neural network model, which consists of one input layer represented by IL, three hidden layers represented by HL, of which HL3 is also a Dropout layer represented by DL and an output layer represented by OL. All layers is followed by ReLU Activation, whereas OL is followed by Sigmoid Activation.

1) **Optimizer**: Adam Optimizer [5], is utilized as described in Equation (4) for our final model training, as it helps the model, learn to make better predictions, adjust the weights of the model during training, and speed up convergence on our large dataset.
The Adam optimization algorithm updates the parameters of a neural network based on the first and second moments of the gradients. The formula for Adam is:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}t &= \frac{v_t}{1-\beta_2^t} \\ \theta_t &= \theta t - 1 - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t, \end{aligned} \tag{4}$$

where m_t and v_t are the first and second moments of the gradients at time step t, g_t is the gradient at time step t, β_1 and β_2 are the exponential decay rates for the first and second moments, respectively, $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first and second moments, α is the learning rate, and ϵ is a small constant used for numerical stability.

2) **Batch Size**: The batch size of 30 which means that the model will be updated after processing 30 samples at a time.

[5]https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

3) **Loss Function**: There is usage of MIL(Multiple Instance Learning) Ranking Loss function proposed in [2] and modified it for better performance. The updated version of MIL is shown via Eq. 5.

$$\begin{aligned}\text{loss} = &\frac{1}{N}\left(\sum_{i=1}^{N}\text{ReLU}\Big(1-y_{\text{am}}+y_{\text{nm}}\Big)\right.\\&+\sum_{i=1}^{N}\left(\sum_{j=1}^{T}y_{\text{anomaly}},j\times\lambda\right)\\&\left.+\sum_{j=1}^{31}\Big(y_{\text{pred}},i,j-y_{\text{pred}},i,j+1\Big)^{2}\times\lambda\right),\end{aligned}\tag{5}$$

where N is the batch size, T is the sequence length, y_{am} is the maximum value in the y_{anomaly} tensor, y_{nm} is the maximum value in the y_{normal} tensor, y_{anomaly}, j is the anomaly score for the j-th data point in the batch, y_{pred}, i, j is the predicted value for the j-th time step of the i-th data point in the batch, $y_{\text{pred}}, i, j+1$ is the predicted value for the $j+1$-th time step of the i-th data point in the batch, and λ is a hyperparameter that controls the strength of regularization and for our experiment, it is set to 8×10^{-5}.

4) **Learning Rate**: The project consists of 0.001 as our initial learning rate and get updated at the milestone of 25 and 50 epochs. Usually, the learning rate controls the step size taken during optimization and it determines how quickly the model parameters are updated in response to the estimated error gradient.
5) **Weight Decay**: Weight decay is a regularization technique used to prevent overfitting in models by adding a penalty term to the loss function that discourages large weights.A weight decay of 1×10^{-12}, which means that the penalty term will be proportional to the square of the weights and scaled by this factor
6) **No. of Epochs**: An epoch refers to one full iteration through the entire training dataset. The number of epochs is the number of times the model will see the complete training dataset during the training process. For our approach, the value is set to 100, which means that the model will observe the training data 100 times during the training phase.

IV. Experimental Results

A. Evaluation Metrics

The Area Under the Curve (AUC) is a commonly used evaluation metric for binary classification models. It is calculated as the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds.

To evaluate a binary classification model using AUC as a metric, the first step is to compute the ROC curve using the predicted probabilities and true class labels. Once the ROC curve is obtained, the AUC can be calculated using the integral formula, as shown in Eq. (6).

$$AUC=\int_{-\infty}^{\infty}ROC(x)dx\tag{6}$$

where $ROC(x)$ represents the ROC curve at a given threshold value x, and dx denotes the differential element in the integral. The resulting value ranges from 0.5 to 1.0, where 0.5 indicates a random classification and 1.0 indicates a perfect classification.

B. Performance Analysis

TABLE II: Performance Comparison of Different Methods

Method	Features	AUC(%)	% Gain
Sultani *et al.*[2]	C3D-RGB	75.41	↓2.5
Sultani *et al.*[2] [**B**]	I3D-RGB	77.92	-
RTFM [29]	I3D-RGB	84.30	↑6.38
MGFN [30]	I3D-RGB	86.98	↑9.06
Ours	**I3D-RGB + I3D-OF**	**87.52**	**↑9.6**

B=Baseline

1) Quantitative Analysis: As shown in Table II, The project consists of comparision with three other methods: [2], [29], and [30] as shown in TABLE II. [2] is our baseline method that uses the UCF-Crime Dataset. While [29] proposes an improvement on [2], [30] shows the best performance on the UCF-Crime Dataset. The approach differs from these methods because there is the usage of the two-stream model that integrates both RGB and optical flow information to capture both appearance and motion information. Our baseline approach [2] uses both C3D and I3D to extract features from only RGB frames, but our two-stream I3D model with RGB and optical flow streams improves performance. And modified the MIL loss function and achieved a performance gain of 9.6%. Also tried to incorporate self-attention mechanisms, but it only achieved 61.67% AUC. Inspired by [30], Used only temporal annotations for testing videos and achieved a 0.54% improvement to 87.52% AUC using a simple 5-layer fully connected neural network with our two-stream I3D feature extractor.

2) Qualitative Analysis: The training progress for the proposed solution on the UCF-Crime dataset is shown in Fig. 5. It shows the training loss progression with each epoch, which is computed using the modified version of the MIL Ranking loss function. It is clearly observed that the magnitude of loss is drastically decreasing till the 30^{th} epoch after that it remains stable, which few ups and down till the last epoch. During the training process, to avoid the issue of overfitting, early stopping regularization[6] is employed with the patience of 10 and a minimum delta value of 0.001, with training loss set as a criterion.

The testing progression is clearly visible in Fig. 6. Here along with loss progress, AUC progression with each epoch is also displayed.

[6]https://pytorch.org/ignite/generated/ignite.handlers.early_stopping.EarlyStopping.html

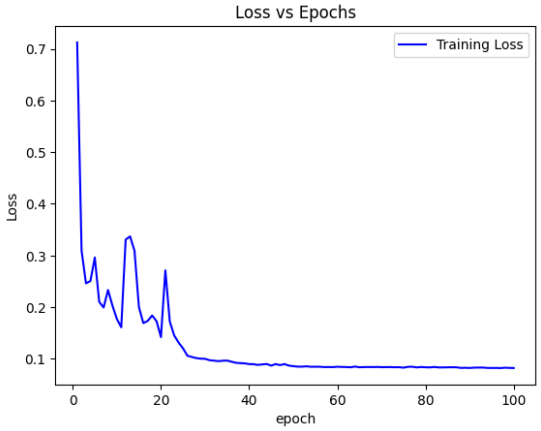


Fig. 5: Training Progress

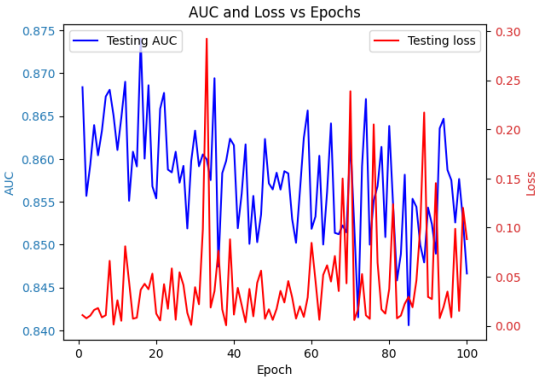
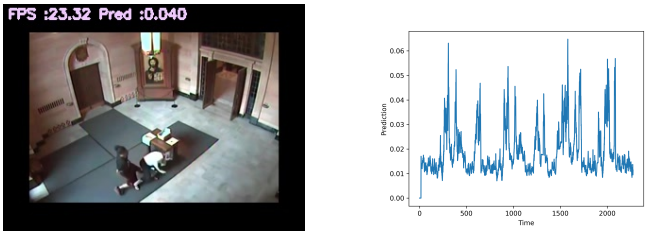


Fig. 6: Testing Progress

Fig. 7a shows the video level snippets from the abuse video set from 13 different anomalies video set present in the UCF-Crime dataset. It also contains the FPS value and real-time prediction score for each frame. While Fig. 7b shows temporal analysis of anomaly prediction on videos used in Fig. 7a. Here higher the prediction value, the higher the chances of that event being anomalous.



(a) (b)

Fig. 7: Analyzing Anomaly Detection Results: Video and Graph Perspective

V. Conclusion and Future Work

The task of video anomaly detection remains an ongoing challenge, requiring existing research to achieve a better surveillance system. Existing approaches have made significant strides in Transfer Learning and Generative models, with features playing a crucial role in improving classification. However, most approaches only consider RGB frames and do not take advantage of valuable optical flow information, which could further enhance performance. In this study, the method proposed uses the two streams of the I3D model as feature extractors, combined with a basic 5-layer fully connected neural network for classification. Also, modifications have been made to the existing MIL ranking loss function to achieve a performance improvement of 12.11% over the Baseline [2] and 0.54% over the state of the art [30]. However, the issue of better separability and identifying temporal-level anomalies remains a challenge. In the future, experiment with our model on various datasets and add more anomalies to the existing dataset in the list. Also, a plan to explore other methodologies to achieve improved performance is also scheduled. Overall, the work highlights the potential of deep feature-based anomaly detection for video surveillance and provides a promising direction for future research in this field.

VI. Acknowledgement

This research was enabled in part by support provided by the Digital Research Alliance of Canada

References

[1] Joao Carreira and Andrew Zisserman. *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*. 2018. arXiv: 1705.07750 [cs.CV].

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

[3] Tiago S Nazare, Rodrigo F de Mello, and Moacir A Ponti. "Are pre-trained CNNs good feature extractors for anomaly detection in surveillance videos?" In: *arXiv preprint arXiv:1811.08495* (2018).

[4] Khac-Tuan Nguyen et al. "Anomaly detection in traffic surveillance videos with gan-based future frame prediction". In: *Proceedings of the 2020 International Conference on Multimedia Retrieval*. 2020, pp. 457–463.

[5] Yumna Zahid, Muhammad Atif Tahir, and Muhammad Nouman Durrani. "Ensemble learning using bagging and inception-V3 for anomaly detection in surveillance videos". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 588–592.

[6] Tuan-Hung Vu et al. "Multi-Channel Generative Framework and Supervised Learning for Anomaly Detection in Surveillance Videos". In: *Sensors* 21.9 (2021), p. 3179.

[7] Joey Tianyi Zhou et al. "Anomalynet: An anomaly detection network for video surveillance". In: *IEEE Transactions on Information Forensics and Security* 14.10 (2019), pp. 2537–2550.

[8] Ke Xu, Tanfeng Sun, and Xinghao Jiang. "Video anomaly detection and localization based on an adaptive intra-frame classification network". In: *IEEE Transactions on Multimedia* 22.2 (2019), pp. 394–406.

[9] Keval Doshi and Yasin Yilmaz. "Continual learning for anomaly detection in surveillance videos". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 254–255.

[10] Shenghao Yu et al. "Cross-Epoch Learning for Weakly Supervised Anomaly Detection in Surveillance Videos". In: *IEEE Signal Processing Letters* 28 (2021), 2137–2141. DOI: https://doi.org/10.1109/lsp.2021.3117737. URL: https://ieeexplore.ieee.org/document/9560033.

[11] Behnam Sabzalian, Hossein Marvi, and Alireza Ahmadyfard. "Deep and sparse features for anomaly detection and localization in video". In: *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*. IEEE. 2019, pp. 173–178.

[12] Sabrina Aberkane and Mohamed Elarbi. "Deep reinforcement learning for real-world anomaly detection in surveillance videos". In: *2019 6th International Conference on Image and Signal Processing and their Applications (ISPA)*. IEEE. 2019, pp. 1–5.

[13] Xing Hu et al. "Video anomaly detection using deep incremental slow feature analysis network". In: *IET Computer Vision* 10.4 (2016), pp. 258–267.

[14] K Kavikuil and J Amudha. "Leveraging deep learning for anomaly detection in video surveillance". In: *First International Conference on Artificial Intelligence and Cognitive Computing*. Springer. 2019, pp. 239–247.

[15] Yi Zhu and Shawn Newsam. "Motion-aware feature for improved video anomaly detection". In: *arXiv preprint arXiv:1907.10211* (2019).

[16] Peng Wu and Jing Liu. "Learning Causal Temporal Relation and Feature Discrimination for Anomaly Detection". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 3513–3527. DOI: 10.1109/TIP.2021.3062192.

[17] Yiling Gong et al. "Multi-Scale Continuity-Aware Refinement Network for Weakly Supervised Video Anomaly Detection". In: *2022 IEEE International Conference on Multimedia and Expo (ICME)* (2022). DOI: https://doi.org/10.1109/icme52920.2022.9860012. URL: https://ieeexplore.ieee.org/document/9860012.

[18] Mahmudul Hasan et al. "Learning temporal regularity in video sequences". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 733–742.

[19] Radu Tudor Ionescu et al. "Object-centric auto-encoders and dummy anomalies for abnormal event detection in video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7842–7851.

[20] Yong Shean Chong and Yong Haur Tay. "Abnormal event detection in videos using spatiotemporal autoencoder". In: *International symposium on neural networks*. Springer. 2017, pp. 189–196.

[21] Mohammad Sabokrou, Mahmood Fathy, and Mojtaba Hoseini. "Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder". In: *Electronics Letters* 52.13 (2016), pp. 1122–1124.

[22] Elvan Duman and Osman Ayhan Erdem. "Anomaly detection in videos using optical flow and convolutional autoencoder". In: *IEEE Access* 7 (2019), pp. 183914–183923.

[23] Anitha Ramchandran and Arun Kumar Sangaiah. "Unsupervised deep learning system for local anomaly event detection in crowded scenes". In: *Multimedia Tools and Applications* 79.47 (2020), pp. 35275–35295.

[24] Sukalyan Bhakat and Ganesh Ramakrishnan. "Anomaly detection in surveillance videos". In: *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. 2019, pp. 252–255.

[25] Karishma Pawar and Vahida Attar. "Application of deep learning for crowd anomaly detection from surveillance videos". In: *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE. 2021, pp. 506–511.

[26] Mohammad Sabokrou et al. "Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes". In: *IEEE Transactions on Image Processing* 26.4 (2017), pp. 1992–2004.

[27] Jia-Xing Zhong et al. "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*. 2019, pp. 1237–1246.

[28] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. *TV-L1 optical flow estimation*. 2013. URL: https://doi.org/10.5201/ipol.2013.26.

[29] Yu Tian et al. *Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning*. 2021. arXiv: 2101.10030 [cs.CV].

[30] Yingxian Chen et al. *MGFN: Magnitude-Contrastive Glance-and-Focus Network for Weakly-Supervised Video Anomaly Detection*. 2022. arXiv: 2211.15098 [cs.CV].