2024 IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) | 979-8-3503-7428-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/AVSS61716.2024.10672568

Dual Deep Learning Network for Abnormal Action Detection

Fath U Min Ullah
School of Engineering and Computing, UCLan,
Preston, United Kingdom.
fath@ieee.org

Zulfiqar Ahmad Khan
Sejong University,
Seoul, Republic of Korea.
zulfiqar@sju.ac.kr

Sung Wook Baik
Sejong University,
Seoul, Republic of Korea.
sbaik@sejong.ac.kr

Estefania Talavera
University of Twente,
Netherlands.
e.talaveramartinez@utwente.nl

Saeed Anwar
King Fahd University of Petroleum and Minerals,
Dhahran 31261, Saudi Arabia.
saeed.anwar@kfupm.edu.sa

Khan Muhammad
Sungkyunkwan University,
Seoul 03063, Republic of Korea.
khan.muhammad@ieee.org

Abstract

Neural networks have demonstrated remarkable effectiveness in solving distinct real-world vision problems pertaining to activity recognition and violence detection in surveillance scenarios. The broad reliance on practicing a single network for spatial and motion information collection has made them less effective for long-term dependency analysis in video snippets. Our work solves this issue through a multi-network fusion strategy suitable for real-world surveillance. Initially, the spatial information is accessed from a compound coefficient strategy inspired by a robust convolutional neural network (ConvNet). Next, the pyramidal convolutional features from two consecutive frames are obtained through LiteFlowNet. The output from both the networks (ConvNet and LiteFlowNet) is separately passed into a deep-gated recurrent Unit (GRU) that is assembled for a skip connection. The latter obtained from each GRU is fused and further propagated to the dense layer for final decision. The results on the datasets and the ablation study confirm our method's efficiency, outperforming the state-of-the-art methods. (Code: GitHub)

1. Introduction

Billions of daily captured videos during surveillance are stored on hard drives for purposeful analysis. Their examinations may reveal unusual events or abnormal activities. These manual settings make the process tedious in recognizing certain activities in massive data. An intelligent system is highly demanded to combat these unusual events for safety because their automatic detection with significant accuracy to eliminate human efforts and tiresome monitoring is challenging. Therefore, a machine-based intelligent method is obligatory for detecting abnormal events, such as violence detection (VD), for safeguarding. Violence is an abnormal or aggressive action that harms or affects an object's state, human being, or pet through physical force. Its detection from videos is mandatory for surveillance systems to ensure safety.

Over the past decade, improvements in computer vision algorithms have enabled the execution of diverse vision tasks, including human activity recognition for security [26], robotics, human-computer interaction, etc. Large-scale action recognition has improved exponentially owing to the availability of large-scale datasets. The primary purpose of these applications is to focus on specific sub-tasks, such as anomaly detection, VD, egocentric activity recognition, etc. The most critical action recognition subset is VD, which is overwhelmingly growing because of its applications in security, safety, crime prevention, etc. A wide variety of automatic VD can be performed through deep learning (DL) algorithms to secure surveillance. An imperative aspect of VD in surveillance is considering both the spatial and motion information in video frames to identify violent scenes. Moreover, it is crucial to efficiently process video frames to reduce computational complexity while ensuring accuracy. In this regard, existing techniques have several challenges, making them incapable of real-time processing.

979-8-3503-7428-5/24/$31.00 © 2024 IEEE

Most VD works are based on a single end-to-end trainable network for both appearance and motion information collection, which is less effective for extracting the most discriminative features. Similarly, most methods are functional in non-surveillance systems; that is, they can only operate on traditional camera data. Therefore, to efficiently address all these challenges, we propose a novel multi-stream network fusion framework to conduct VD on surveillance videos, which includes the following contributions.

1. Existing ConvNet models apply a single scaling strategy (depth, width, or resolution) to extract fine-grained features. Their arbitrary scaling based on manual tuning results in suboptimal accuracy and efficiency. We strengthen this by employing EfficientNet ConvNet, which applies the compound coefficient scaling to all required model dimensions for fine-grained information collection.

2. Feature extraction from sequential video data is considerably time-consuming for motion information owing to its high dimensionality and computational complexity. The proposed method uses LiteFlowNet, a novel procedure developed initially for motion estimation between two consecutive frames. To capture the motion, we extract pyramidal ConvNet features from the intermediate layers of LiteFlowNet, which is comparatively smaller and faster than the state-of-the-art (SOTA) optical flow ConvNet models.

3. Sequential learning algorithms, such as long-short-term memory (LSTM) and its variants, are popular learning mechanisms that can learn long-term information. However, their complex gated architecture and storage units render real-time processing difficult using such methods. To solve this issue, a skip connection is established in its alternative gated recurrent unit (GRU), which is simpler and faster than the conventional LSTM.

4. After conducting the experiments, we empirically proved that the proposed method outperformed SOTA using standard surveillance benchmarks. The ablation study on its variants further verified its performance in real-time surveillance systems.

2. Proposed Method

The functional details of our VD framework are divided into three sections and are depicted in Fig. 1. First, the data acquisition is explained, followed by our feature extraction strategy. Section 2.3 discusses the learning procedure adopted in the activity sequence.

Layers	Channels	Resolution	Operation
1	32	224 × 224	Conv3×3
1	16	112 × 112	MBConv1, K3×3
2	24	112 × 112	MBConv6, K3×3
2	40	56 × 56	MBConv6, K5×5
3	80	28 × 28	MBConv6, K3×3
3	112	14 × 14	MBConv6, K5×5
4	192	14 × 14	MBConv6, K5×5
1	320	7 × 7	MBConv6, K3×3
1	1280	7 × 7	Conv1×1 & Pooling & FC

Table 1. EfficientNet: Architectural details of baseline networks.

2.1. Data Acquisition

The data are acquired using surveillance cameras installed to monitor human activities. The input frames from the video data are passed to a feature extraction step, which collects concrete information using two networks for spatial and pyramidal feature extraction. The length of the input frames prior to the feature extraction depends on the nature of the network. The first network obtains a single frame to extract the spatial features, whereas the second network receives two consecutive frames to estimate the motion flow between the two frames. These steps are performed online, whereas the training (offline) uses publicly available surveillance benchmarks collected in real-world scenes (details are given in Section 3. Next, each dataset is divided using a standard data-splitting procedure (i.e., training: 70%, testing: 20%, and validation: 10%) for experiments. Further details of the data settings for VD training are given in 3.

2.2. Feature Extraction Timeline

The feature extraction timeline consists of dual networks in charge of spatial and motion feature collection to learn the activity sequence. The details of each network are provided below:

2.2.1 Spatial Feature Collection

ConvNets are developed for fixed resource purposes, and their performance is boosted via different scaling strategies. To extract spatial features, we employ EfficientNet, which is more accurate and efficient than existing ConvNets. Using various resources, there are several ways to scale up networks to improve and boost their performance. For instance, ResNet-18 can be scaled up to ResNet-200 through layer (depth) adjustments, whereas MobileNets [13] and WiderResNet [31] are scaled through network width (channels). Existing methods demonstrate that network width and depth are essential for improving ConvNet performance via effective scaling for better accuracy. Accordingly, EfficientNet uses a compound scaling method to scale up all width, depth, and resolution dimensions of MobileNets and

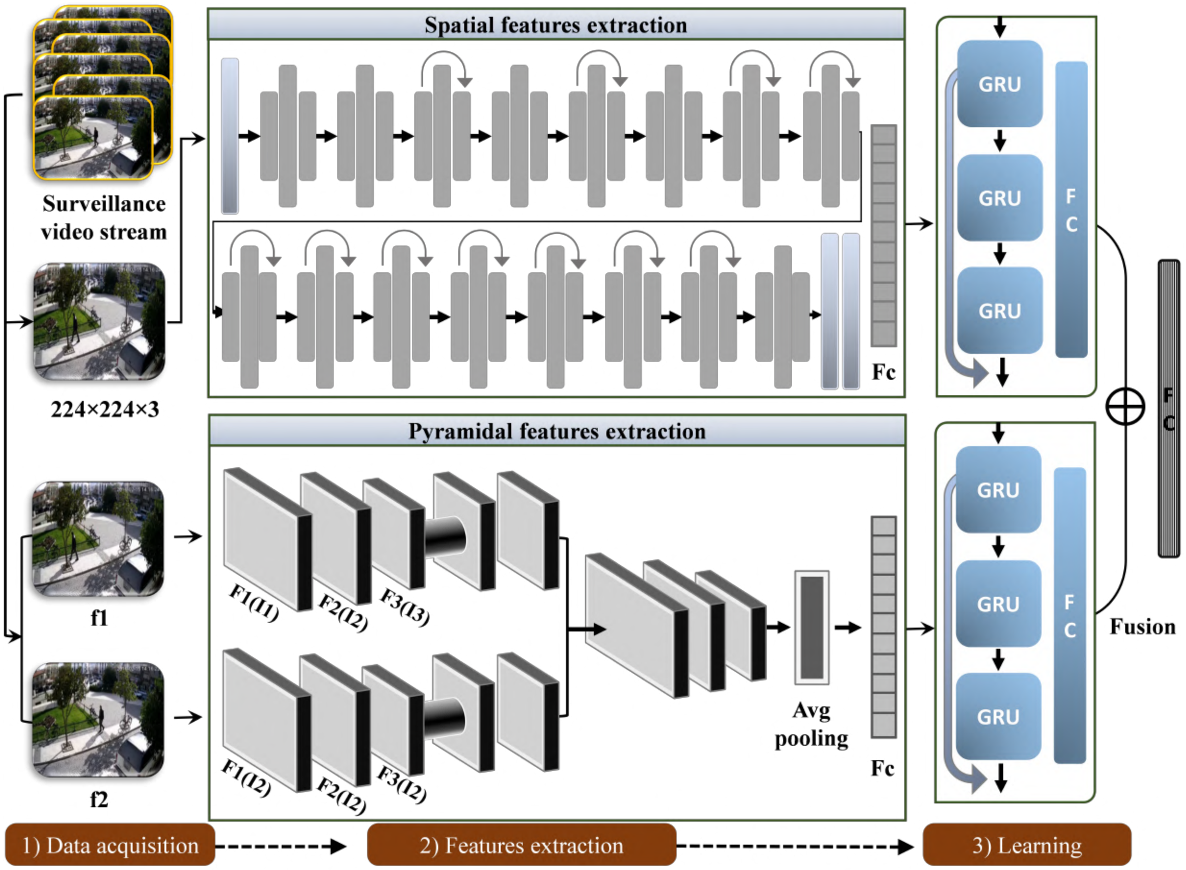


Figure 1. Overview of the proposed VD framework in video surveillance. 1) Data acquisition, showing videos acquired from the surveillance cameras. 2) Two networks responsible for spatiotemporal and pyramidal feature extraction. 3) The deep GRU learns the features from both networks in the learning phase. The features obtained from both networks are fused in the FC layer for the final VD output.

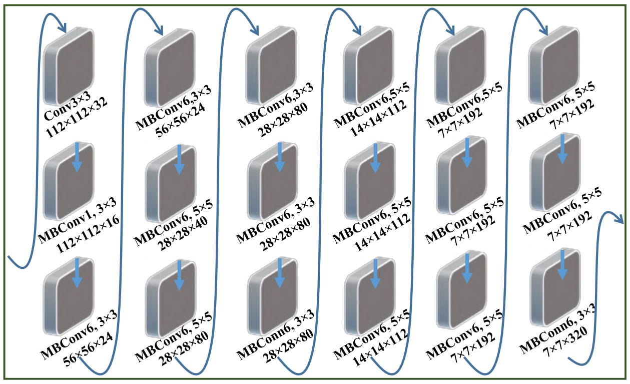


Figure 2. Detailed architecture of the EfficientNet that extracts spatiotemporal features.

ResNet. The compound scaling method uses a compound coefficient ∅ that uniformly scales the width, resolution, and depth of the network: depth: $D = \alpha^{\emptyset}$, width $W = \beta^{\emptyset}$, resolution $R = \gamma^{\emptyset}$, s.t. $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$, $\alpha \geq 1$, $\beta \geq 1$, $\gamma \geq 1$. Here, D, W, and R are constants determined by the hyperparameter tuning technique known as grid search. Similarly, ∅, which manages the available resources for scaling, is specified by the user. The constants indicate how additional resources can be assigned to the network dimensions.

Usually, the FLOPS of a convolution is proportional to D, W^2, and R^2, which means that doubling the value of the network depth will also increase the FLOPS by four times. However, convolutions usually dominate the computational cost of ConvNets. The baseline network (EfficientNet-B0) consists of 18 layers with a defined number of channels and resolutions. The features obtained from the final layer are forwarded into a deep GRU, where they are analyzed for long-term dependencies in the sequence. A visual representation of EfficientNet is shown in Fig. 2, and its architectural details are reported in Table 1.

2.2.2 Pyramidal Features Extraction

ConvNet has attained significant achievements in vision-related tasks, including action/activity recognition. Activity recognition involves pattern collection regarding motion and visual information that changes in a video sequence. Visual pattern representation in still images has attained SOTA performance since the introduction of AlexNet. However, the content representation in video frames remains a highly challenging task. To deal with this problem, researchers have proposed obtaining features from the space volume, 3D ConvNet, spatiotemporal attributes [12], etc. However, reasonable results are still lacking. Therefore, along with visual content, motion is a significant parameter in a video sequence that can precisely explain a particular activity. The most commonly used approach is optical flow, which can effectively capture the motion details. However, these approaches fail to capture the smallest movements and extract the precise flow contained in the frame sequence. To address this challenge, we exploit a ConvNet-assisted optical flow-based model known as LiteFlowNet [14], a modified version of FlowNet2. This network can be executed quickly and can extract slight motions, helping to represent activity/violent actions in videos. Therefore, the pyramidal ConvNet features obtained from LiteFlowNet can be represented as activity/violent action received in the video data.

Two lightweight networks specialized in pyramidal feature extraction to estimate the optical flow from LiteFlowNet, an end-to-end DL architecture, were trained on two labeled optical flow images via supervised learning to generate a flow within two consecutive frames. LiteFlowNet has a processing pipeline similar to that of FlowNet2. Because the spatial dimension of the feature vector contracts in the feature encoder and the flow field expands in the decoder, the two sub-networks are named NetC and NetE. NetC transforms an input image into two pyramids with high-dimensional features. NetE comprises the cascaded flow and regularization tools and estimates the flow fields from low to high spatial resolution. First, the entire network extracts semantic features from each image by processing pairs of images. The correlation layer depicts the change between the two images by combining their semantic representations. In the proposed technique, we extracted the pyramidal features from the intermediate layer of LiteFlowNet, capable of motion extraction from a pair of images provided after the correlation layer. This layer compares the patches between two feature maps obtained from the pyramidal pipeline. The feature maps obtained from this convolutional layer contain 128 channels with dimensions of 10 × 8. Each feature map is convoluted via average pooling; subsequently, representative features are obtained. Average pooling is used because it precisely reduces dimensionality by capturing all feature effects in the kernel as the mean value.

2.3. Fusion and Learning Phase

Recurrent neural networks (RNNs) are specifically designed for learning sequence data and are extended forms of conventional feedforward neural networks. RNNs utilize recurrent hidden states to process the sequence in which the activation of the hidden states at each time step depends on the activation value in the previous time steps [8]. Similarly, many studies underline the vanishing gradient problem of RNNs when there are long-term dependencies in the sequence [21]. To overcome these problems, researchers have introduced LSTM and GRU. LSTM contains gated recurrent units, such as input, output, forget gates, and memory cells, which make the LSTM structure complex. Consequently, LSTM networks require significant computations for sequence processing. However, the GRU includes only two gates, namely, reset and update gates, and contains one activation unit, which makes the GRU effective at processing sequential data in real-time. In the proposed method, skip connections are added instead of using a fully connected GRU. Usually, going deeper into the LSTM or GRU structure, the model suffers from the vanishing gradient problem, which is efficiently solved by adding skip connections in the GRU layers. Li et al. [19] visualized different DL structures to observe their internal structures. Accordingly, they reported that this connection helps to keep the loss function chaotic, which leads to convex loss and makes it easy to fall into a local minimum. GRU represents sequential dependencies in data and contains recurrent units similar to the LSTM, which propagate the pattern flow in the recurrent units. The mathematical representation of the GRU processing is stated in Eqs. 1 - 4.

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j \tag{1}$$

$$z_t^j = \sigma(W_z x_t + U_z h_t)^j \tag{2}$$

$$\tilde{h}_t^j = \tanh(W x_t + U(r_t \circ h_{t-1}))^j \tag{3}$$

$$r_t^j = \sigma(W_r x_t + U_r h_t)^j \tag{4}$$

The linear interpolation h_{t-1}^j computes the activation h_t^j in GRU j^{th} unit at the time step t, hidden state in the previous time step, and activation of the current hidden state h_{t-1}^j. In our case, $h_t^j = 1024$ features are obtained from the two consecutive frames at t. z_t^j is an update gate working as a specific kernel that decides how much each unit activation needs to be updated. GRU has several functionalities in LSTM, such as the linear sum of the current and previous hidden states. h_t^j is a hidden state computed at a certain time in the same way as in conventional RNNs. Similarly, r_t is the reset gate; a value close to zero indicates that the information of the previous state is forgotten.

3. Experimental Results

This section describes the overall experimental procedure, dataset details, results, and empirical assessment. We included details regarding system design and implementation settings. The method's effectiveness is confirmed on three benchmark datasets, namely, RWF-2000 [7] (denoted as **D1**), Surveillance Fight [2] (**D2**), and Hockey Fight [3] (**D3**) datasets; the achieved results are discussed in detail. Finally, the proposed method is compared against the baseline methods.

Dataset	Samples	Frame resolution	Video length (sec)	Sample in each class	Frame rate
D1	2000	Variable	5	1000	30
D2	300	Variable	2	150	20-30
D3	1000	360×240	1.6∼1.96	500	25

Table 2. Statistics of the VD datasets.

3.1. Platform Configuration

The system implementation includes data loading, processing for training, and other configuration settings. Our method was developed in Python version 3.6 using the Spyder integrated development environment. We used the well-known DL framework Keras with Tensorflow (version 2.4) as a backend to extract the spatial features. The system settings for this process utilized a Windows 10 operating system with Nvidia GeForce RTX 2060 GPU. However, pyramidal features were extracted using another popular DL, Caffe, with Ubuntu as the operating system. Other details include significant libraries, such as OpenCV, related to computer vision. The Windows 10 operating system was used with the previously explained settings to fuse both network features.

Technique	Dataset	ACC (%)	PRE	REC	F1-score
ConvNet	D1	84.60	0.8447	0.844	0.844
	D2	79.20	0.8357	0.807	0.789
	D3	88.62	0.8858	0.886	0.885
LiteFlowNet	D1	73.40	0.7344	0.725	0.727
	D2	53.13	0.2656	0.500	0.346
	D3	47.25	0.2362	0.500	0.320
Proposed Method	D1	**90.10**	**0.8998**	**0.900**	**0.900**
	D2	**96.69**	**0.9690**	**0.965**	**0.966**
	D3	**98.62**	**0.9860**	**0.986**	**0.986**

Table 3. Investigation of different networks and their comparative analysis against the proposed method using VD datasets.

3.2. Datasets for Experiments

We explored surveillance datasets for usage in the evaluation of the methods. These datasets consist of two classes: violent and non-violent actions. The statistical details of these datasets are presented in Table 2. Further information regarding each dataset is given as follows.

RWF-2000 (D1) is the most challenging dataset, recently introduced by Cheng et al. [7]. It contains videos of indoor and outdoor real-world surveillance environments against diverse backgrounds during the day and night. The captured scenes belong to only surveillance and are not modified into other forms. Another dataset is **Surveillance Fight (D2)** [2], which consists of indoor and outdoor real-world surveillance videos from day and nighttime. Most of the videos are captured in stores, outways, industries, entrances to buildings, shops, etc., making the dataset significantly challenging. The **Hockey Fight (D3)** [3] comprises two classes of videos taken from a hockey playground (the National Hockey League). The actions of violence and non-violence were recorded when players were colliding and hitting each other.

	D1		D2		D3	
	Violent	NV	Violent	NV	Violent	NV
Violent	1991	225	163	2	415	7
NV	260	2427	8	130	4	374

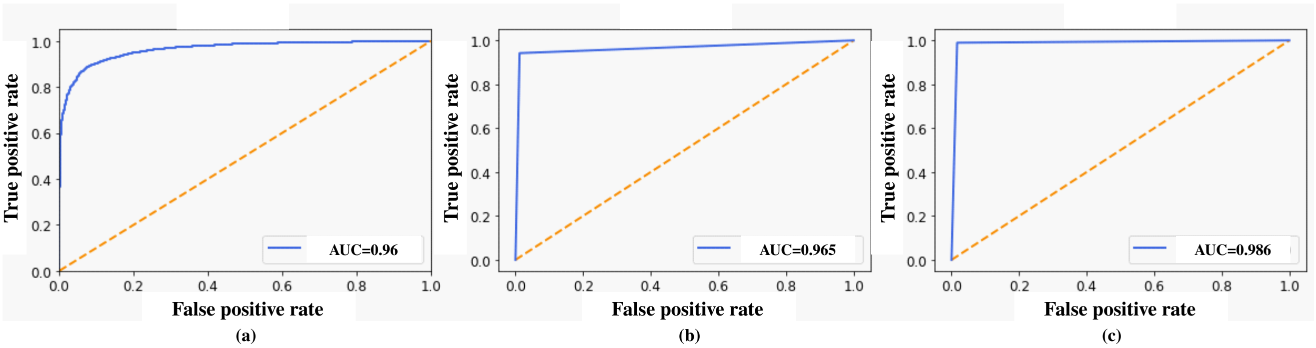
Table 4. Confusion matrix of our method over all three datasets. Non-violent is indicated as NV in this table.

3.3. Results and Discussion

This section explains the results and evaluation of our method using empirical analysis. An ablation study is also conducted to validate the effectiveness of our method. For a fair analysis, each network is connected by the deep GRU, which has a skip connection. The evaluation metrics are the area under the curve (AUC) (see Fig. 3), precision, recall, and accuracy, commonly used in VD tasks. Our method's competence is assessed using AUC values and the ROC curve. Similarly, the accuracy and corresponding loss are plotted to visualize the model's performance throughout the training session.

Initially, ConvNet was evaluated using D1, and the corresponding results are reported in Table 3. Spatial features obtained from the frames were extracted via ConvNet to identify violent and non-violent patterns. Similarly, the accuracy (ACC), precision, recall, and F1-score obtained using ConvNet on the D1 were 84.60%, 0.8447%, 0.8441%, and 0.8444%, respectively. From these results, we observe that ConvNet performed the best on D3. In ConvNet, a compound scaling strategy is justified by the logic that if an input image size is larger, the network needs more layers and channels to increase the receptive field and capture the detailed patterns. Accordingly, the same network acts as an end-to-end framework, where the features are extracted, and the respective output is designated as violent or non-violent.

Next, LiteFlowNet extracts pyramidal features and consists of a two-stream network containing filter weights shared across the streams. The streams act as feature descriptors, transferring the image into a pyramid of multi-scale high-dimensional features from the highest spatial resolution to the lowest spatial resolution. These pyramidal features are generated through stride convolutions, where a



(a) (b) (c)

Figure 3. Detailed analysis of the proposed method using ROC and AUC values. The ROC is drawn between the true positive and false positive rates. (a) ROC computed for D1 with AUC:0.96 (b) Computation for the D2 dataset. (c) ROC obtained for the D3 dataset.

Methods	Core operator	Accuracy (%)
Dipon et al. [10]	1-3D conv	86.00
Kunal et al. [6]	ConvLSTM	85.97
Qiming et al. [22]	RCN	87.50
Ming et al. [7]	FGN	87.20
Herwin et al. [15]	Spatial Motion Extractor	88.50
Yukun et al. [27]	SPIL	89.30
Piotr et al. [4]	Sliding window	93.70
Proposed method	Two-stream	**94.50**

Table 5. Comparative analysis of the proposed method against SOTA VD methods in terms of accuracy using the D1.

factor reduces the spatial resolution in the pyramids. Compared to ConvNet, this network performs slightly worse on D1, achieving 73.40% accuracy. However, its accuracy on D2 and D3 is 53.13% and 47.25%, respectively. Furthermore, the best performance was achieved by our method, where the features obtained from the two networks were fused to produce the desired results. A deep GRU follows each network with a skip connection, which is later propagated into a fully connected layer. In this manner, the outputs from each FC layer are fused using a concatenate function, and the final feature map with a size of 2024 is forwarded to the FC. For further inspection of our method, other metrics, such as confusion metrics, AUC, and ROC, were computed, showing the correct and incorrect VD. The ROC curves of our method for all three datasets are presented in Figs. 3 (a), (b), and (c), respectively. The qualitative results of our method using D1 are shown in Fig. 4, which verifies its effectiveness and exceptional performance. Similarly, the confusion metrics of our method for all three datasets are reported in Table 4.

Furthermore, the processing speed of the proposed model is 1.15 sequences per second, complemented by 33.03 MFlops (Million Floating Point Operations per Second), with a size of 5.27 MB and 455,098 parameters. These metrics underscore the model's performance and efficiency in processing single sequences, which is vital for real-time applications. The MFlops metric highlights the model's computational efficiency. Additionally, the small model size indicates its optimization for resource efficiency.

Methods	Core operator	Accuracy (%)
Şeymanur et al. [2]	LSTM	72.00
Fathu et al. [29]	ConvNet	74.00
Şeymanur et al. [1]	VIT	84.60
Min-Seok et al. [17]	Motion Saliency Map	92.00
Mustaqeem et al. [18]	Temporal Convolution	92.50
Proposed method	Two-stream	**96.69**

Table 6. Results assessment of the proposed method against SOTA VD methods in terms of accuracy using the D2.

Methods	Core operator	Accuracy (%)
Ismael et al. [24]	Elastic cuboid trajectories	82.50
Chen et al. [11]	Positive and unlabeled learning	85.20
Javad et al. [23]	HOMO, SVM	89.30
Herwin et al. [15]	Spatial Motion Extractor	98.20
Saba et al. [16]	Deep CNN	92.89
Ismael et al. [25]	Hybrid features	94.60
Jing et al. [30]	Kernel extreme learning machine	95.05
Febin et al. [9]	SIFT, Movement	96.50
S et al. [5]	DDE	98.27
Fathu et al. [28]	RNN	98.50
Ji et al. [20]	DenseNet	98.30
Proposed Method	Two-stream	**98.62**

Table 7. Results assessment of our method against SOTA VD methods using the D3.

Scalability to Other Violence Forms:
The proposed method is currently based on two classes (violent and non-violent), where ”violent” refers specifically to fights. However, there are other forms of violence, such as falling, road accidents, robbery, shoplifting, etc. Although these are different types of anomalies, our method can be extended and modified to consider them. For these anomalies, the UCF-Crime dataset, which includes these types of incidents, can be used.

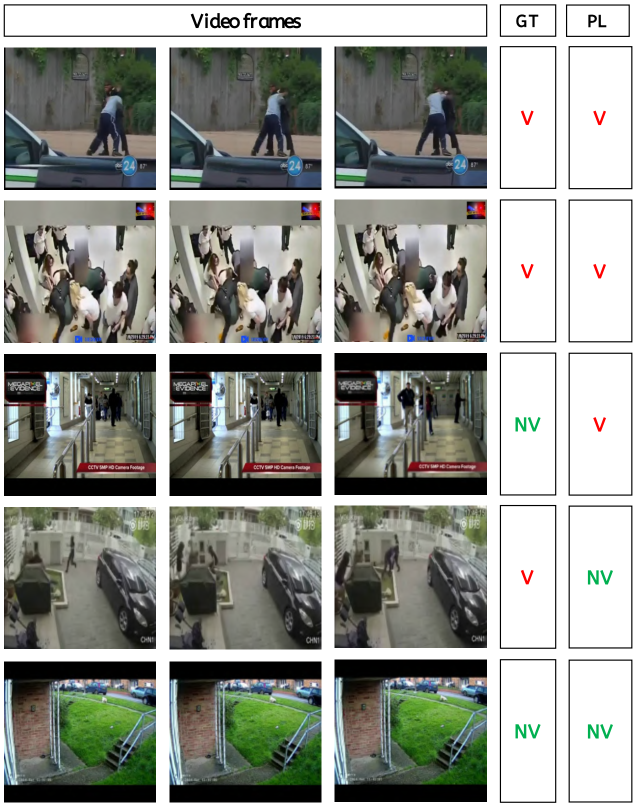


Figure 4. Qualitative results of the proposed method for a VD task conducted on D1 dataset, which is chosen owing to the variety of its content. The proposed method correctly predicts the output classes that can be observed from the 1^{st}, 2^{nd}, and last rows; whereas an incorrect prediction occurs in the 3^{rd} row owing to the speedy movement of several people. Next, violence is detected as non-violence in the 4^{th} row because it initially occurs behind the box. Terms such as V, NV, GT, and PL indicate Violent, Non-violent, Ground Truth, and Predicted Label, respectively.

3.4. Comparative Assessment of Our Method

This section explains the comparative analysis of our method through an assessment against SOTA. VD methods mainly assess their results based on accuracy and their core operators. We perform the assessments using these metrics to measure the effectiveness of our method. Assessing D1, 3D-human skeleton points were formulated from videos in [27] and applied to interaction learning on a 3D skeleton point. Furthermore, the authors of [27] proposed skeleton point interaction learning (SPIL) for modeling the interaction between skeleton points. Next, a dual spatiotemporal convolutional network was proposed in [10] which extracted spatial and temporal information from video frames using 1D, 2D, and 3D-ConvNet. Similarly, along with the attention mechanism, a long-term recurrent convolutional network (RCN) was proposed in [22]. The method [7] formed the D1 and implemented several networks, such as 3D-ConvNet, optical flow, and flow-gated network (FGN). The results for D1 are presented in Table 5. A wide range of surveillance-based VD methods have emerged that use the D2, such as the method in [2], which proposed different LSTM-based approaches to deal with surveillance VD. Authors in [29] proposed a method for precise VD using the analysis of video patterns in surveillance. In [1], different approaches were applied for VD and vision transformer (ViT). Another method in [17] combined 2D-ConvNets and applied a frame-grouping strategy to make 2D-ConvNet capable of learning spatiotemporal representations in videos. The results related to the D2 dataset are presented in Table 6.

Regarding D3, the method evaluation was done via an optical flow-based method [23], where a histogram of optical flow magnitude and orientation (HOMO) was introduced. Another method [24] proposed spatiotemporal elastic cuboid trajectories to model different motions specific to the fight. Authors in [30] presented a three-dimensional histogram (3D-HOG), combining feature pooling technology and bag-of-visual-words. 3D-HOG features are extracted at the block level from videos, where K-means clustering is applied for visual word generation. The method in [9] proposed a cascaded-based VD technique using movement filtering and motion boundary SIFT. A method in [11] introduced a multi-manifold positive and unlabeled algorithm. Similarly, a framework based on deep discriminative embedding (DDE) using residual spatiotemporal autoencoders was proposed that was learned via V2AnomalyVec embedding [5]. The method proposed in [25] advanced the VD using Hough Forest and 2D-ConvNet. The results for the D3 dataset are presented in Table 7.

4. Conclusion

This study addressed the challenges of working with spatial and motion features for VD from videos using a multi-stream approach. Spatial and pyramidal features were extracted via efficient ConvNet and LiteFlowNet models, respectively, fed into GRUs. Both GRUs were implemented with a skip connection approach, and their output was fused to provide the final feature map for VD. Our results surpassed SOTA by significant margins, showing superiority of our method. In the current study, VD practice was performed using single-view surveillance videos. In the future, we aim to develop a multi-view VD dataset and evaluate its performance using different variants of ConvNets and RNNs.

References

[1] Ş. Aktı, F. Ofli, M. Imran, and H. K. Ekenel. Fight detection from still images in the wild. In *WACV*, pages 550–559, 2022.

[2] Ş. Aktı, G. A. Tataroğlu, and H. K. Ekenel. Vision-based fight detection from surveillance cameras. In *IPTA*, pages 1–6. IEEE, 2019.

[3] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar. Violence detection in video using computer vision techniques. In *CAIP*, pages 332–339. Springer, 2011.

[4] P. Bilinski and F. Bremond. Human violence recognition and detection in surveillance videos. In *AVSS*, pages 30–36. IEEE, 2016.

[5] S. Chandrakala and L. Vignesh. V2anomalyvec: Deep discriminative embeddings for detecting anomalous activities in surveillance videos. *TCSS*, pages 1307–1316, 2021.

[6] K. Chaturvedi, C. Dhiman, and D. K. Vishwakarma. Fight detection with spatial and channel wise attention-based convlstm model. *Expert Systems*, page e13474, 2024.

[7] M. Cheng, K. Cai, and M. RWF. An open large scale video database for violence detection. In *ICPR*, pages 4183–4190, 2000.

[8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[9] I. Febin, K. Jayasree, and P. T. Joy. Violence detection in videos for an intelligent surveillance system using mobsift and movement filtering algorithm. *PAA*, pages 611–623, 2020.

[10] D. K. Ghosh, A. Chakrabarty, N. Mansoor, D. Y. Suh, and M. J. Piran. Learning-driven spatio-temporal feature extraction for violence detection in iot environments. In *ICTC*, pages 1807–1812. IEEE, 2021.

[11] C. Gong, H. Shi, J. Yang, and J. Yang. Multi-manifold positive and unlabeled learning for visual analysis. *TCSVT*, pages 1396–1409, 2019.

[12] K. Hara, H. Kataoka, and Y. Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCV workshops*, pages 3154–3160, 2017.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[14] T.-W. Hui, X. Tang, and C. C. Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, pages 8981–8989, 2018.

[15] H. A. Huillcen Baca, F. d. L. Palomino Valdivia, and J. C. Gutierrez Caceres. Efficient human violence recognition for surveillance in real time. *Sensors*, page 668, 2024.

[16] S. A. Jebur, K. A. Hussein, H. K. Hoomod, and L. Alzubaidi. Novel deep feature fusion framework for multi-scenario violence detection. *Computers*, page 175, 2023.

[17] M.-S. Kang, R.-H. Park, and H.-M. Park. Efficient spatio-temporal modeling methods for real-time violence recognition. *Access*, pages 76270–76285, 2021.

[18] M. Khan, A. El Saddik, W. Gueaieb, G. De Masi, and F. Karray. Vd-net: An edge vision-based surveillance system for violence detection. *Access*, pages 43796–43808, 2024.

[19] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. *NIPS*, 31, 2018.

[20] J. Li, X. Jiang, T. Sun, and K. Xu. Efficient violence detection using 3d convolutional neural networks. In *AVSS*, pages 1–8. IEEE, 2019.

[21] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek. Videolstm convolves, attends and flows for action recognition. *CVIU*, pages 41–50, 2018.

[22] Q. Liang, Y. Li, K. Yang, X. Wang, and Z. Li. Long-term recurrent convolutional network violent behaviour recognition with attention mechanism. In *MATEC Web of Conferences*, page 05013. EDP Sciences, 2021.

[23] J. Mahmoodi and A. Salajeghe. A classification method based on optical flow for violence detection. *ESA*, pages 121–127, 2019.

[24] I. Serrano, O. Deniz, G. Bueno, G. Garcia-Hernando, and T.-K. Kim. Spatio-temporal elastic cuboid trajectories for efficient fight recognition using hough forests. *MVA*, pages 207–217, 2018.

[25] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno. Fight recognition in video using hough forests and 2d convolutional neural network. *TIP*, pages 4787–4797, 2018.

[26] J. Su, P. Her, E. Clemens, E. Yaz, S. Schneider, and H. Medeiros. Violence detection using 3d convolutional neural networks. In *AVSS*, pages 1–8. IEEE, 2022.

[27] Y. Su, G. Lin, J. Zhu, and Q. Wu. Human interaction learning on 3d skeleton point clouds for video violence recognition. In *ECCV*, pages 74–90. Springer, 2020.

[28] F. U. M. Ullah, K. Muhammad, I. U. Haq, N. Khan, A. A. Heidari, S. W. Baik, and V. H. C. de Albuquerque. Ai-assisted edge vision for violence detection in iot-based industrial surveillance networks. *TII*, pages 5359–5370, 2021.

[29] F. U. M. Ullah, M. S. Obaidat, K. Muhammad, A. Ullah, S. W. Baik, F. Cuzzolin, J. J. Rodrigues, and V. H. C. de Albuquerque. An intelligent system for complex violence pattern analysis and detection. *IJIS*, pages 10400–10422, 2022.

[30] J. Yu, W. Song, G. Zhou, and J.-j. Hou. Violent scene detection algorithm based on kernel extreme learning machine and three-dimensional histograms of gradient orientation. *MTA*, pages 8497–8512, 2019.

[31] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*. BMVA, 2016.