

Efficient Two-Stream Network for Violence Detection using Separable Convolutional LSTM

Authors

Md. Zahidul Islam
160041010

Mohammad Rukonuzzaman
160041016

Raiyan Ahmed
160041037

Supervised by

Md. Hasanul Kabir, PhD
Professor

**A thesis submitted to the Department of CSE
in partial fulfillment of the requirements for the degree of
Bachelor of Science in CSE**



**Department of Computer Science and Engineering
Islamic University of Technology
Organization of the Islamic Cooperation (OIC)**

Dhaka, Bangladesh

March 15, 2021

Declaration of Authorship

This is to certify that the work presented in this thesis, titled, “**Efficient Two-Stream Network for Violence Detection using Separable Convolutional LSTM**”, is the outcome of the investigation and research carried out by Md. Zahidul Islam, Mohammad Rukonuzzaman, Raiyan Ahmed, under the supervision of Prof. Md. Hasanul Kabir, PhD. It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma, or other qualifications. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

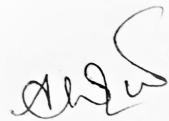
Authors:



Md. Zahidul Islam
Student ID: 160041010



Mohammad Rukonuzzaman
Student ID: 160041016



Raiyan Ahmed
Student ID: 160041037

Supervisor:



Md. Hasanul Kabir, PhD
Professor
Department of Computer Science and Engineering
Islamic University of Technology

Acknowledgement

We would like to convey our heartfelt gratitude and appreciation to everyone who contributed to the completion of our thesis by providing inspiration, encouragement, and guidance.

Our supervisor, **Prof. Md. Hasanul Kabir**, PhD, Dept. of Computer Science and Engineering, Islamic University of Technology (IUT), has provided us with invaluable supervision, expertise, and encouragement. We extend our gratitude towards him for his constant guidance.

We are indebted to **Moshiur Farazi**, PhD, researcher and postdoctoral fellow, CSIRO-Data61, Canberra, Australia for his guidance and advices in the course of this work.

Finally, we seize this opportunity to express our profound gratitude to our loving parents for their unwavering love and spiritual and mental support.

Contents

<i>Declaration of Authorship</i>	i
<i>Acknowledgement</i>	ii
List of Figures	v
List of Tables	vii
<i>Abstract</i>	viii
1 Introduction	1
1.1 Action Recognition	1
1.2 Violence Detection	2
1.3 Problem Statement	3
1.4 Challenges	3
1.5 Objectives	4
1.6 Contributions	5
1.7 Organization of the Thesis	5
2 Background Study	7
2.1 Hand-crafted feature based methods	8
2.2 Deep learning based methods	9
2.3 CNN-LSTM architectures	12
2.3.1 Detecting Violent Scenes and Affective Impact in Movies with Deep Learning	13
2.3.2 Multi-stream deep networks for person to person violence detec- tion in videos	14
2.3.3 Learning to Detect Violent Videos using Convolutional Long Short- Term Memory	15
2.3.4 Bidirectional Convolutional LSTM for the Detection of Violence in Videos	16

3	Proposed Method	18
3.1	Pre-processing	19
3.2	Network Architecture	20
3.3	Depthwise Separable Convolution	22
3.4	MobileNet	25
3.5	Separable Convolutional LSTM	26
3.6	Fusion Strategies	27
3.7	Classifier Network	28
3.8	Loss Function	29
3.9	Training Methodology	30
4	Result Analysis and Discussion	31
4.1	Datasets	31
4.2	Experiment on Standard Benchmark Datasets	32
4.2.1	Learning Curves	34
4.3	Ablation Studies	34
4.4	Comparative Analysis of Efficiency	38
4.5	Qualitative Analysis	38
5	Conclusions	40
	References	41

List of Figures

1.1	A typical methodology or workflow for general action recognition. For each video clip action recognition aims to predict a label or class defining the corresponding action.	2
1.2	Violent activity caught on surveillance camera. Video clips taken from RWF-2000 dataset [1]	2
2.1	Contrast between machine learning and deep learning approach for violence detection	7
2.2	Extraction of moSIFT features from video frames. [2]	8
2.3	Extraction of Oriented Violent Flows feature descriptors which utilizes change in motion magnitude and orientation. [3]	9
2.4	3D ConvNets architecture for detecting violence in video clips proposed by Ding <i>et al.</i> [4]	10
2.5	Visual-Auditory feature fusion network proposed by Peixoto <i>et al.</i> [5]	11
2.6	Basic architecture of a CNN-LSTM model for video data	12
2.7	Proposed Network Architecture by Dai <i>et al.</i>	13
2.8	Proposed Architecture by Dong <i>et al.</i> [6]	14
2.9	Convolutional LSTM Model Architecture	15
2.10	A diagram of Spatiotemporal Encoder model architecture proposed by Hanson <i>et al.</i> [7] which is comprised of VGG13Net and BiConvLSTM layers.	17
3.1	Schematic overview of our proposed network	18
3.2	Input pre-processing for the proposed model. (a) shows key-frames of an example video clip. (b) demonstrates the effect of performing background suppression on video frames of (a). The last row (c) shows time-steps of the frame difference derived from the video clip of (a).	19

3.3	The proposed model is composed of two CNN-LSTM streams with similar architecture. Each stream consists of a truncated MobileNet module generating spatial features from each time-step of the inputs. These features are passed to the SepConvLSTM cell in each stream to produce Spatio-temporal encodings. The outputs from each stream are fused using a Fusion layer and passed to the classifier network.	21
3.4	SepConvLSTM cell	23
3.5	bottleneck residual block in MobileNetV2 [8]	24
3.6	Model architecture summary of mobilenet	25
3.7	SepConvLSTM cell	27
3.8	classifier network of the proposed network	29
4.1	a) Training curve of experimenting with SepConvLSTM-A model. b) Training curve of experimenting with SepConvLSTM-C model. c) Training curve of experimenting with SepConvLSTM-M model. The SepConvLSTM-M model achieved the best accuracy among the three variants of our proposed model which has a fusion strategy of multiplying the LeakyRelu activation of the frames stream with sigmoid activation of the difference stream.	33
4.2	a) Training curve of experimenting with only Difference stream of SepConvLSTM-C model. b) Training curve of experimenting with only Frames stream of SepConvLSTM-C model. Accuracy using Difference stream is much higher than using Frames stream only. This indicates that body movements and motion patterns produce more discriminative features than appearance based features like color, texture, etc.	36
4.3	a) Training curve of experimenting by replacing SepConvLSTM layer with 3D convolutional layers. b) Training curve of experimenting by replacing SepConvLSTM layer with ConvLSTM in the model SepConvLSTM-M. c) Training curve of experimenting by replacing SepConvLSTM layer with ConvLSTM in the model SepConvLSTM-C. The lower accuracy and higher parameter count of these models indicates that SepConvLSTM is a more efficient and robust choice over these layers.	37
4.4	Qualitative results of the proposed model (SepConvLSTM-M) for violence detection on the RWF-2000 dataset. The first two rows contain examples of video clips for which our model correctly predicts the presence of violence. The last four rows contain examples of failure cases where ambiguous body movements and poor quality of surveillance footage may lead towards incorrect prediction.	39

List of Tables

3.1	Summary of the proposed model’s architecture with parameter counts and output shapes. Here, b stands for batch size.	22
4.1	Comparison of Classification Results on Standard Benchmark Datasets . .	32
4.2	Analyzing contribution of each stream to our model for violence detection on RWF-2000 dataset	35
4.3	Analyzing contribution of SepConvLSTM to our model by replacing it with 3D-Conv and ConvLSTM layers	35
4.4	Comparison of Efficiency with Earlier Models	38

Abstract

Automatic detection of violence from surveillance footage holds special significance among the various subsets of general activity recognition tasks due to its broad applicability in autonomous security monitoring systems, web video censoring, etc. In this paper, we propose a two-stream deep learning architecture based on Separable Convolutional LSTM (SepConvLSTM) and pre-trained truncated MobileNet, in which one stream processes difference of adjacent frames and the other stream takes in background suppressed frames as inputs. Fast and efficient input pre-processing techniques were used to highlight moving objects in frames by suppressing non-moving backgrounds and capturing motion in between frames. These inputs assist in producing discriminative features as violent activities are predominantly characterized by rapid movements. SepConvLSTM is built by replacing each ConvLSTM gate's convolution operation with a depthwise separable convolution, resulting in robust long-range spatio-temporal features with significantly fewer parameters. We experimented with three fusion strategies to merge the output feature maps of the two streams. Three standard public datasets were used to assess the proposed methods. On the larger and more difficult RWF-2000 dataset, our model outperforms the previous best accuracy by more than 2%, while matching state-of-the-art results on the smaller datasets. Our studies demonstrate that the proposed models excel both in terms of computational efficiency and detection accuracy.

Chapter 1

Introduction

We provide an overview of our thesis in this chapter. At first, we discuss the research area of general activity recognition and its applications. Then we introduce violence detection, one of the significant sub-tasks of action recognition. We present a brief overview of the research so far in the area of violent activity detection and its applications in real-world scenarios. Then we stated problem statement of our research work and mention the challenges. We discuss the objective and contributions of our thesis. Lastly, we provide the organization of the rest of our thesis.

1.1 Action Recognition

Human activity classification is a widely investigated task in the field of computer vision that has diverse applications in human-computer interaction, robotics, surveillance, etc. [9–12] In recent years, large-scale video action recognition has gained impressive improvements mostly because of the wide availability of large datasets, deep neural network architectures, video representation techniques, etc. Many works, on the other hand, focused on specific sub-tasks of action recognition such as spatial-temporal localization of activity, anomaly detection, action quality analysis (AQA) [13, 14], egocentric activity recognition [15], etc. One such important subset is violence detection which is widely applicable in public monitoring, surveillance systems, internet video filtering, etc.

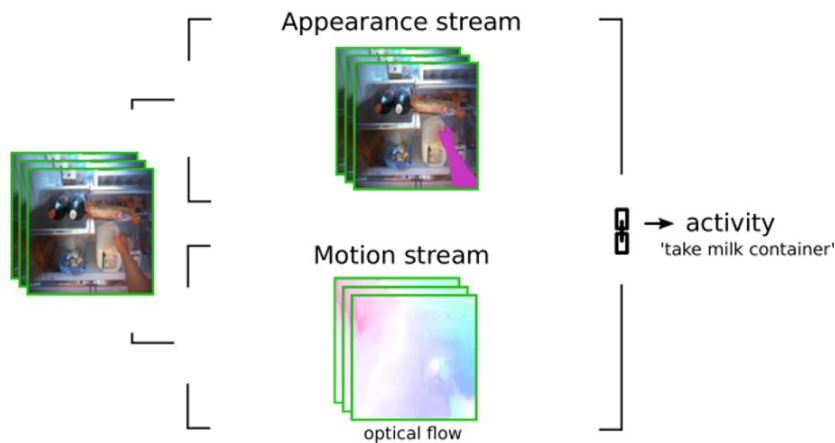


Figure 1.1: A typical methodology or workflow for general action recognition. For each video clip action recognition aims to predict a label or class defining the corresponding action.

1.2 Violence Detection

As digital media technologies like surveillance cameras are getting more and more ubiquitous, detecting violence from captured footage using manual inspection seems increasingly challenging. To counter this issue, researchers have suggested different methodologies that can detect violence from surveillance footage automatically without requiring any human interaction. Violence detection is a section of general action recognition task which specifically focuses on detecting aggressive human behaviors such as fighting, robbery, rioting, etc.



Figure 1.2: Violent activity caught on surveillance camera. Video clips taken from RWF-2000 dataset [1]

Earlier works on violence detection mostly focused on engineering various descriptors that could effectively capture violent motion present in the video [16–18]. Later on, the performance of these handcrafted features was surpassed by several end-to-end trainable deep learning methods which require little to no pre-processing [7, 19, 20]. To validate the effectiveness of these methods, commonly three standard violence detection datasets were used called Hockey, Movies, and Violent-Flows. Recently, a new dataset

called RWF-2000 has been proposed which is substantially bigger and more diverse. For applying these deep learning models in real-life practical scenarios both computational efficiency and accuracy need to be considered. In this respect, we present a novel two-stream CNN-LSTM based network that can produce discriminative Spatio-temporal features while requiring fewer parameters. In general action recognition tasks, surroundings or background information may serve as discriminative clues. For example, to identify the action *playing golf*, a background with green grass might be a good indicator. On the other hand, violent activities are mostly characterized by the body position, movements, and interactions whereas appearance-based features like color, texture, and background information play a minor role. Considering these factors, we used background suppressed frames and frame difference as the inputs to our network both of which help generate discriminative features to recognize violence.

1.3 Problem Statement

As violence detection is most likely to be applied in real-life time-sensitive scenarios where detection of violence needs to be both accurate and fast. Recent approaches in action recognition uses optical flow as inputs for enhancing the encoding of temporal information. But calculating optical flow for each frame is computationally expensive. Most of the networks proposed so far in the existing literature use modules with large parameter counts which also increases the computational burden. Reducing the computational burden while maintaining performance is a possible area of improvement. The existing literature don't reach very high accuracy in the most diverse and challenge dataset in the field of violence detection indicating the lack of generalization ability in many of these approaches. Proposing a network which generalizes well for diverse types of videos like black and white and color, day and night, various resolution and modalities can be an important contribution. The problem statement can be summed up as follows - *“Modelling a system for violent activity detection from surveillance footage which is robust in varying real-life situations and efficient enough to be deployed in low-end devices.”*

1.4 Challenges

There are many difficulties and challenges which make the problem of violence detection quite hard. The first issue is representation of the input video from which the presence of violent activity needs to be determined. Generally, video clips are repre-

sented as a 4d tensor of shape $T \times H \times W \times C$ where T is the number of time-steps, H and W is the height and width of each frame, C is the number of channels. In a video clip there are T time-steps or frames. But, many studies show that in most video clip where the fps is 30 or higher adjacent frames don't contain very different information. To put it differently, adjacent frames mostly contain redundant information. Producing robust and effective feature from each video clip that can help distinguishing violent videos from non-violent is crucial in the task of violence detection. In the classical hand-crafted feature based method designing the most effective feature to represent the video was an important area of study. Various works aimed to capture motion information using feature descriptors. Later on with the advent of deep learning research, these features are now mostly automatically learned by the deep neural network. Another challenge is finding the best neural network architecture that fits the task. Among the various architecture of neural networks, researchers in the field of action recognition initially opted to use 3D convolutional networks. Later some works used separate streams of networks where each stream specifically focuses on certain types of information. Recurrent neural networks are heavily researched in action recognition task because recurrent neural networks deal with sequential data and video is a sequence of frames or images. Another crucial aspect of developing a violence detection system is efficiency. If the system proposed for violence detection is resource inefficient or computationally expensive, then it would not effective in real-life time-sensitive scenarios. As surveillance cameras are very ubiquitous at the present, it would be particularly beneficial if violence detection systems can be deployed in low-end devices or mobile embedded vision applications as well.

1.5 Objectives

- To build a robust yet efficient violent activity detection method using modern deep learning methods in video classification that can be leveraged to implement real-time autonomous surveillance systems or internet video filtration.
- Finding out an optimum deep neural network architecture that best fits the task. The network should achieve good accuracy while ensuring that the number of parameters and FLOPs used is as less as possible making the computational cost of the proposed method low in terms of time and memory.
- Finding out an optimum type of recurrent neural network module (LSTM) that can effectively encode spatio-temporal features without requiring high computational cost.

- Analyzing the performance and effectiveness of our proposed methods in comparison to the methods presented in the existing published literature.

1.6 Contributions

We can encapsulate our significant contributions of this work in the following points:

- We developed an efficient two-stream deep learning architecture leveraging Separable Convolutional LSTM (SepConvLSTM) and truncated MobileNet.
- We used fast and efficient input pre-processing techniques to emphasize moving objects in frames by suppressing non-moving backgrounds and capturing motion in between frames.
- We employed SepConvLSTM, which is built by replacing each ConvLSTM gate's convolution operation with a depthwise separable convolution, allowing us to use much less parameters. Three fusion techniques for integrating the performance features of two streams were examined.
- Three standard benchmark datasets are used to validate our models' efficiency. On the RWF-2000 dataset, the proposed model outperforms the previous best result and matches state-of-the-art performance on the other datasets. Our model is also efficient in terms of the required number of parameters and FLOPs.

1.7 Organization of the Thesis

We organized the rest of this thesis as follows:

In chapter 2, we provide various existing published literature in the field of violence detection. We explain various methods used for detection violence presented in the existing literature in brief.

In chapter 3, we present our proposed methods with detailed explanation. We go over various parts of the proposed deep learning pipeline like input pre-processing, network architecture etc. This chapter provides various diagrams and schematics representing the proposed methodology for better understanding.

Chapter 4 presents analysis of results that we obtained in our experiments. It contains comparison of the proposed method with other existing methodologies. This chapter

also present discussion and analysis of the obtained results both qualitatively and quantitatively.

Lastly in chapter 5, we discuss possible future works and conclude this thesis.

Chapter 2

Background Study

We present a detailed study of the existing methods proposed for violence detection in the published literature. We discuss an overview of the hand-crafted feature based methods and deep learning based methods. As our proposed method for violence detection is based on CNN-LSTM network we dedicate a separate section on CNN-LSTM based methods on violence detection at the end of this chapter.

Several studies approached the problem of violence detection using various methods ranging from hand-crafted motion feature based methods to end-to-end trainable deep learning networks. In the most recent literature it is evident that deep learning based methods have gained superiority both in terms on accuracy and efficiency.

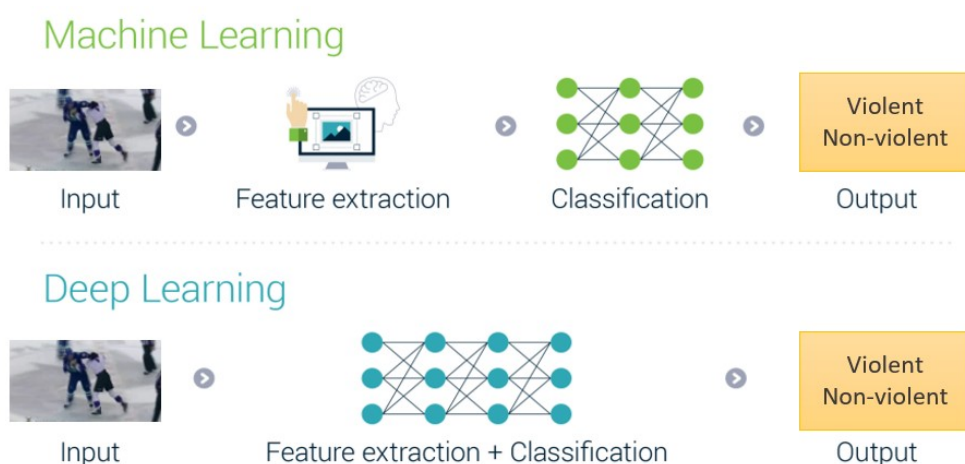


Figure 2.1: Contrast between machine learning and deep learning approach for violence detection

2.1 Hand-crafted feature based methods

Classical methods for violence detection were mostly focused on designing hand-crafted features that explicitly represent motion trajectory, the orientation of limbs, local appearance, inter-frame changes, etc.

Using two such features, Motion Scale Invariant Feature Transform (MoSIFT), and Spatio-temporal Interest Points (STIP), Nievas *et al.* [2] proposed leveraging Bag-of-Words framework. MoSIFT descriptor represents spatiotemporal points of interest at various scales is based on popular image feature descriptor called STIP. They also introduced two well-known violence detection datasets. They are called Hockey and Movies dataset.

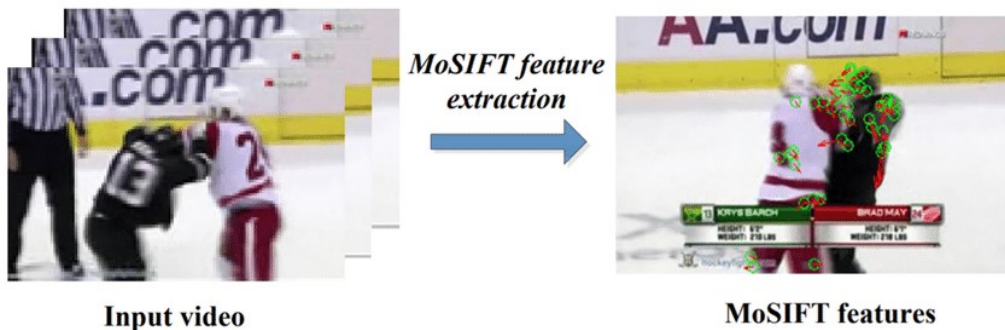


Figure 2.2: Extraction of moSIFT features from video frames. [2]

Hassner *et al.* [21] developed the Violent Flows (ViF) feature using changes of optical flow magnitudes. The Violent Flows feature is a descriptor that describes various statistics for short duration clips reflecting motion patterns. They used a Linear Support Vector Machine to classify these descriptor and thereby detect violence in videos.

Improving upon this work, Gao *et al.* [3] incorporated motion orientations and proposed Oriented Violent Flows (OVIF). They made two primary contributions in this field. They proposed a novel feature extracting method which they named Oriented Violent Flows. This method fully utilizes the change in motion magnitude and orientation. Extraction of OVIF features from video sequence is shown in figure 2.3. For classification, a combination of OVIF and ViF with AdaBoost and Linear SVM beat the previous best results on Violent Flows dataset.

Deniz *et al.* [22] proposed estimating extreme acceleration using Radon Transform on adjacent frames. But the computational cost of this method is a obstruction for its full utilization in real-life scenerios. Senst *et al.* [16] proposed using Lagrangian directional fields for background motion compensation. Seranno *et al.* [23] leveraged Hough Forests and 2D CNN to create a hybrid framework combining both handcrafted and learned fea-

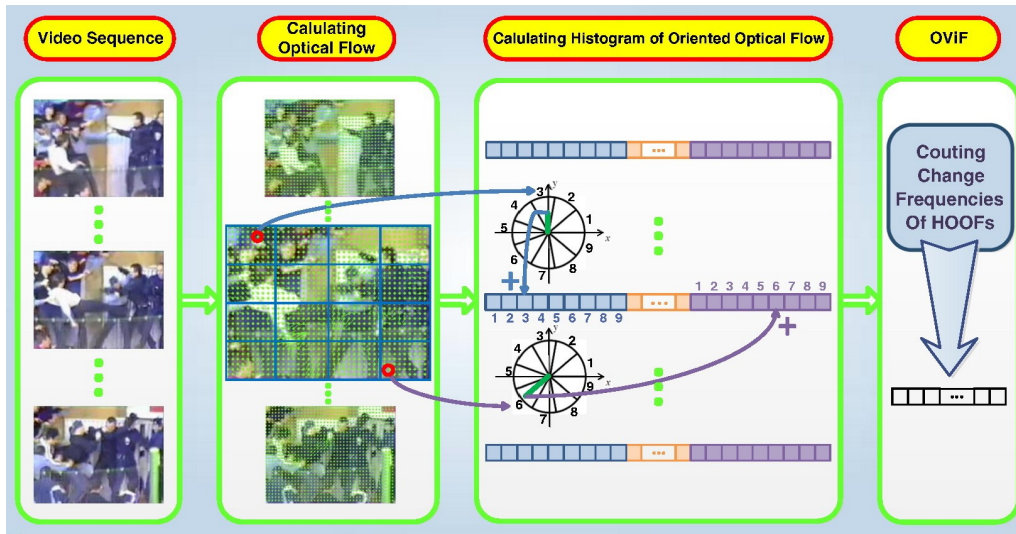


Figure 2.3: Extraction of Oriented Violent Flows feature descriptors which utilizes change in motion magnitude and orientation. [3]

tures. However, handcrafted feature-based methods are mostly unsuitable for deploying in real-world applications due to their restricted generalization ability in diverse situations.

2.2 Deep learning based methods

Popularity of deep learning based methodologies lead to many works on violence detection focusing on building end-to-end trainable networks that perform well with little to no pre-processing. Ding *et al.* [4] employed a 3D Convolutional Network to recognize violence directly from raw inputs. They evaluated the model in hockey dataset with a result accuracy of 91%, which shows that the method achieves better performance than using handcrafted features. The neural network architecture used by Ding *et al.* is illustrated in figure 2.4. This task, however, uses 3D convolution with 2D pooling which allow the input signals to lose temporal information.

Following the success of two-stream networks [24] on general activity recognition tasks, Dong *et al.* [6] added acceleration stream with spatial and temporal ones for detecting person to person violence. Trajectory based representation techniques such as optical flow, acceleration, or frame difference on separate streams boost temporal feature learning. The the spatio-temporal interest points are extracted and passed in the next layers in the model to extract features and perform classification.

Dai *et al.* [25] employed a CNN-LSTM based architecture with two streams. They train a CNN model with ImageNet dataset sub-classes that are particular to violence de-

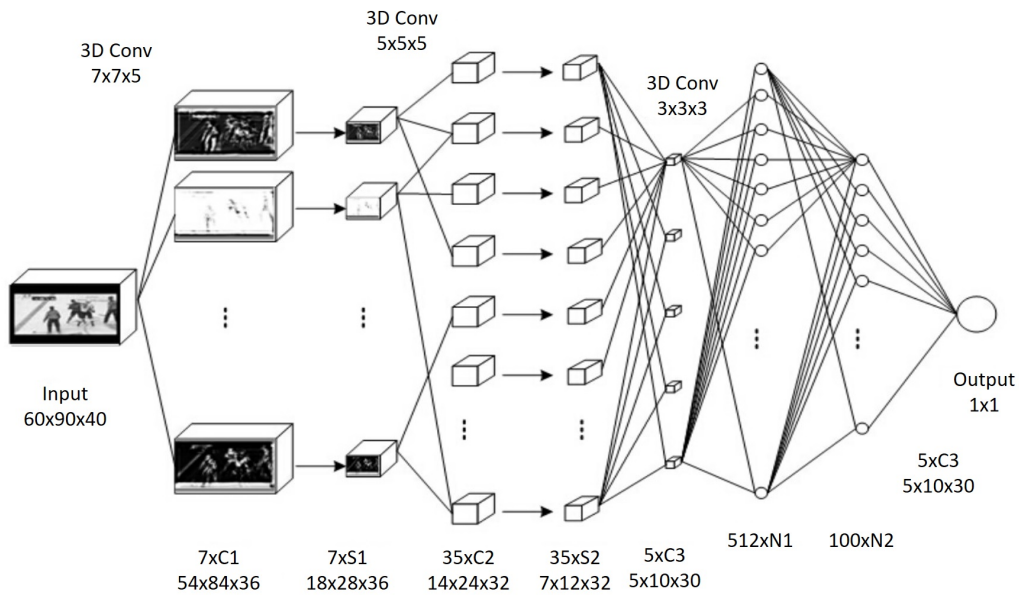


Figure 2.4: 3D ConvNets architecture for detecting violence in video clips proposed by Ding *et al.* [4]

tection. One of the streams in the two stream architecture is for feature extraction from static frames, while the other stream is for motion optical flows. The two stream CNNs are followed by a 1 dimensional LSTM to capture the temporal dynamics and a final SVM classifier for classification.

The initial works on CNN-LSTM models used a fully connected regular LSTM layer that takes in 1-dimensional feature vectors as inputs and does not retain the spatial properties of the features learned by CNNs [6]. On the other hand, using fully connected 2D LSTM layers is not feasible as they need a huge number of parameters.

Sudhakaran *et al.* [20] proposed using ConvLSTM [26] as the recurrent unit to aggregate frame-level features which implements gate operations inside LSTM cell using convolutions reducing parameter count to a great extent. ConvLSTM can preserve spatial information and are capable of working on 2D features without flattening them to 1D vectors. They also showed that training on the difference of adjacent frames enhanced performance. Later, Hanson *et al.* [7] extended this work to allow bidirectional temporal encodings in the feature vectors by using BiConvLSTM that leverages long-range information in both temporal directions. Li *et al.* [19] proposed an efficient 3D CNN based on DenseNet [27] architecture which requires significantly fewer parameters, which does not require the hand-crafted features or RNN layers solely for temporal feature encoding. The improved design follows lightweight units that capture motion patterns exploiting the DenseNet model to facilitate reuse of features and channels.

Peixoto [5] *et al.* employed two deep neural nets to extract spatio-temporal features representing specific concepts. One of the neural network streams is for visual feature extraction using the Inception v4 architecture, while the other extracts audio features using the proposed shallow network classifier. On a later stage the extracted visual features and respective auditory features are merged to make a single feature vector which is then fed into a fusion layer to predict the result. Their proposed network is illustrated in figure 2.5.

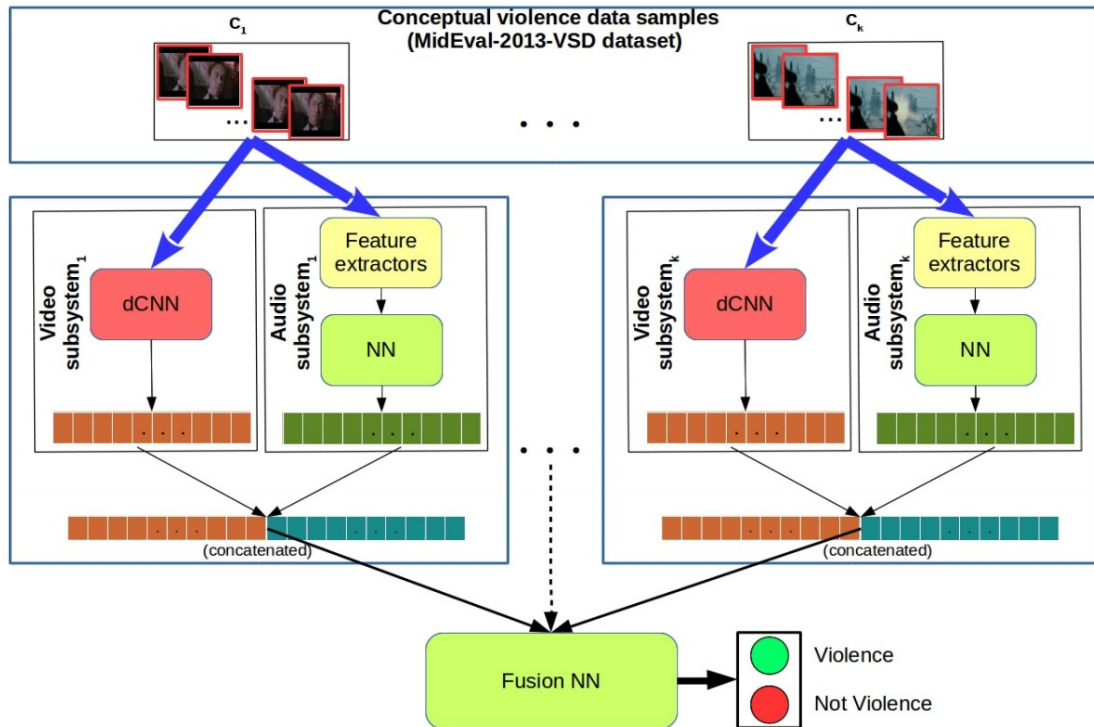


Figure 2.5: Visual-Auditory feature fusion network proposed by Peixoto *et al.* [5]

Peng *et al.* claim previous analysis was simplistic, for example short-clip grouped, single scenario or under-supplied (single modality). So they published a large-scale and multi-scene dataset named XD-violence [28] with a total length of 217 hours that contains 4754 untrimmed audio signals and weak labelled videos to resolve the issues mentioned. Some works [5, 28] reviewed here focuses on multimodal detection of violence by combining visual and auditory cues. However, as the audio signal is generally unavailable in surveillance footage, most works concentrated on visual information.

In our work, we leveraged MobileNet [29] which is a lightweight 2D CNN that uses depthwise separable convolutions and clever design choices to develop a fast and efficient model geared towards mobile and embedded vision applications. The paper present two global hyperparameters, width multiplier and resolution multiplier, that can effectively trade off between accuracy and latency. These hyperparameters allow models to have custom size suitable for the application based constraints. The model weights are trained

on the widely popular ImageNet dataset. Despite the low parameter count, MobileNet has a higher classification accuracy comparable to larger classification models. We also employed Separable Convolutional LSTM (SepConvLSTM) which is constructed by replacing the convolution operations in the LSTM gates with depthwise separable convolutions. In a recent study, Separable Convolutional LSTM has been used for speeding up video segmentation task [30]. However, we did not find any work in the field of activity recognition that focuses on utilizing SepConvLSTM.

2.3 CNN-LSTM architectures

Since our proposed method is based on CNN-LSTM networks, in this section we first discuss the basic architecture of a general CNN-LSTM based model. Then we go over two previous such architectures used for detecting violence in detail.

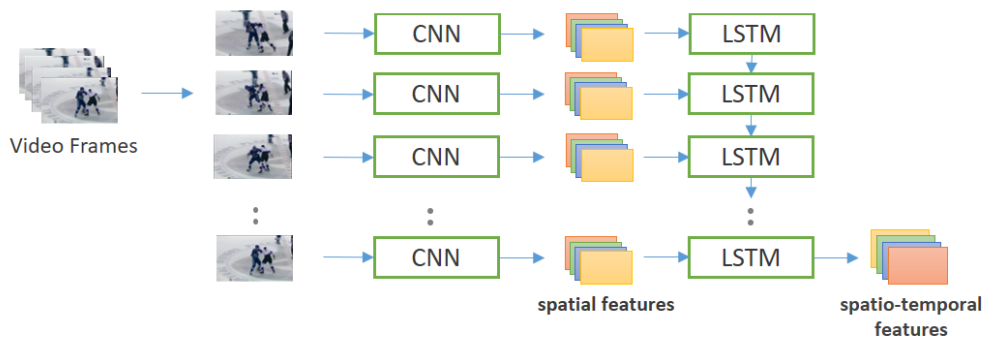


Figure 2.6: Basic architecture of a CNN-LSTM model for video data

In a basic CNN-LSTM based architecture, there are two parts. The first part is a Convolutional Neural Network or CNN part which is followed by a Long Short Term Memory or LSTM part. As shown in Figure 2.6, the CNN layer works on each frame separately and generate a spatial feature map set for each frame. As CNN acts of each frame separately it has no context or information of the temporal dynamics in-between the frames. On the other hand LSTM is designed to work time-series data. Here the LSTM layer works with a time-series of frames or more precisely the spatial feature maps. The spatial feature maps of each timestep is passed into the LSTM layer. As, LSTM is a recurrent neural network the outputs at each time-step is passed into the LSTM as input of the next time-step. The hidden state of the last time-step is extracted as the output of the LSTM layer which gives us spatio-temporal feature maps representing the entire

video clip as it contains information of both spatial and temporal dimensions.

2.3.1 Detecting Violent Scenes and Affective Impact in Movies with Deep Learning

Dai *et al.* [25] presented a violence detection method of concatenating two streams of ConvNet to LSTM and used an SVM classifier for final predictions. Following with the traditional popular trajectory-based models, the paper suggests the use of feature descriptors like HOG, HOF and MBH for object detection and feature extraction which is further used to identify the spatio-temporal interest points (STIP). Dai *et al.* aimed to formulate the scene features into consideration, the features used here limited absolute representations as they are local feature extractors. As illustrated in figure 2.7, the extracted features are fed into CNN streams one each for spatial, temporal and violence information extraction. The CNN features are then concatenated which is processed using an LSTM and finally passed through an SVM classifier.

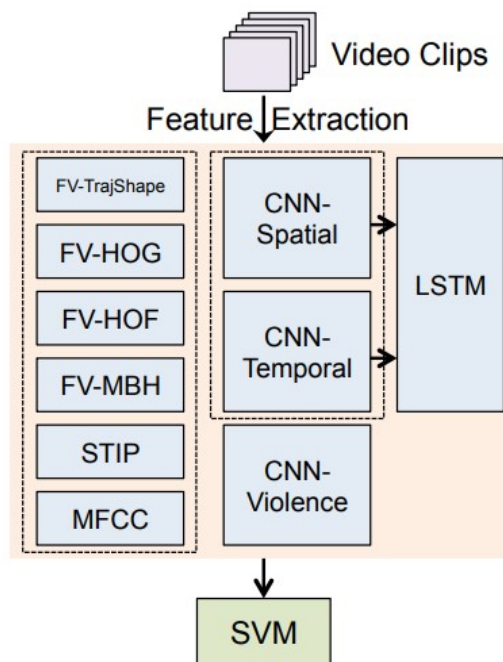


Figure 2.7: Proposed Network Architecture by Dai *et al.*

2.3.2 Multi-stream deep networks for person to person violence detection in videos

Dong *et al.* proposed a three-stream neural network comprising of 2D CNN and 1D LSTM layers. The proposed network architecture by Dong *et al.* [6] illustrated in figure 2.8. The first stream called the spatial stream learns to extract features from video frames and the other stream which is the temporal stream capture the information from neighboring frames and approximate the velocity variation per unit time which are used for general activity recognition applications. In addition to the prior mentioned two streams, the paper suggests the use of an acceleration stream that can capture more violent cues that deals with elements such as velocity and acceleration for actions.

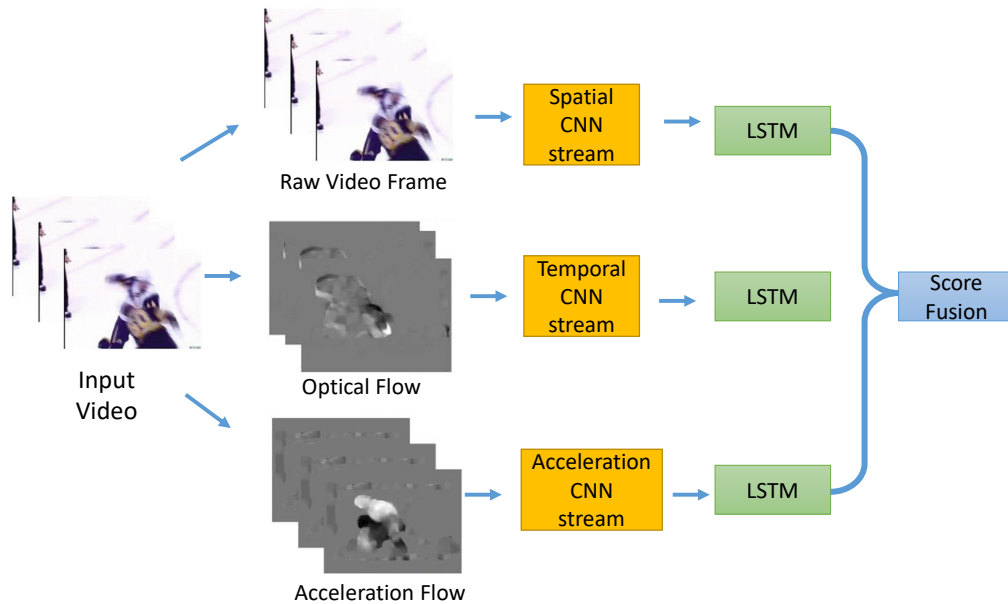


Figure 2.8: Proposed Architecture by Dong *et al.* [6]

Velocity is first order change while acceleration is second order change in frames, computed on the basis of 3 consecutive frames. The acceleration stream is a novel feature of this paper, used to extract the dynamic information resulting in multi stream ConvNets architecture. The acceleration stream uses a proposed novel feature descriptor, acceleration flow. The stacked optical flow or acceleration streams can only map short-term movements in specific time window, so LSTMs are required to capture the whole information. All the streams pass through respective CNN and LSTM layers and later fused to get a fusion score. Their proposed method uses 1D LSTM layers which can not retain the 2D spatial information learned by CNNs as 1D LSTM only works with flattened

1D features. Another drawback is that, The aforementioned optical flow and acceleration streams are computationally expensive to calculate.

2.3.3 Learning to Detect Violent Videos using Convolutional Long Short-Term Memory

Sudhakaran *et al.* [20] employed a CNN-LSTM based network for violence detection. CNN layers learn to extract 2D spatial features from video frames, which are then processed by the LSTM layer for exploring the temporal dynamics in-between the frames.

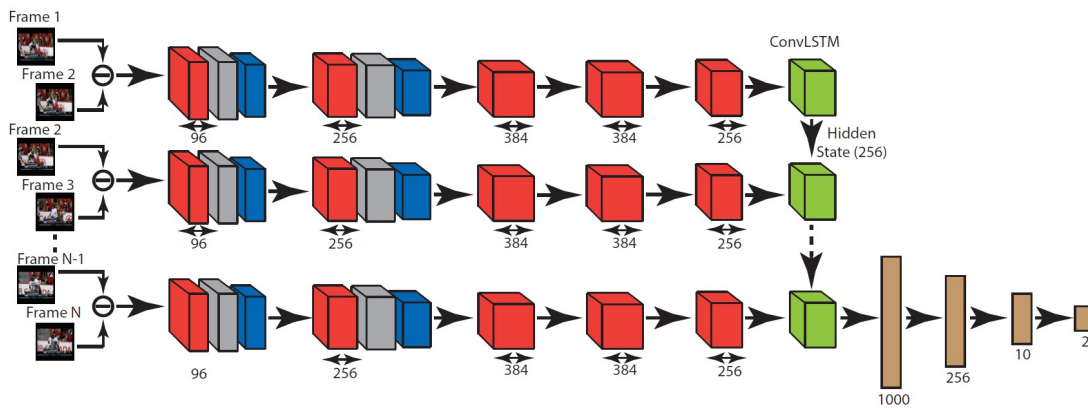


Figure 2.9: Convolutional LSTM Model Architecture

Their proposed neural network architecture for detecting violence in video clips is presented in figure 2.9. The following is an outline of how the network operates: one by one, the frames from the video are transferred to the model. We get the representative feature maps of the input video after all the frames have been applied from the hidden state of the convLSTM layer in this final time step. The feature maps derived from the convLSTM layer in passed to the classification layers which are basically some fully-connected layers with Relu activation in-between.

Authors employed the AlexNet network [31] pre-trained on ImageNet as the CNN model for extracting features from each frame of the input video clip. Each LSTM gate in the Convolutional LSTM cell has 256 filters. When they used the difference of adjacent frames as inputs, they showed a slight improvement in performance. For earlier and smaller datasets, this architecture worked well. However, when we tested this method on a recently proposed larger dataset, we discovered that it fell short of state-of-the-art accuracy. This structure has a lot of space. This architecture has a large memory and computation requirement which makes it unsuitable to be deployed in low end devices.

2.3.4 Bidirectional Convolutional LSTM for the Detection of Violence in Videos

Hanson *et al.* [7] expanded the work of Sudhakaran *et al.* [20] and proposed the model Spatiotemporal Encoder which is illustrated in figure 2.10. The Spatiotemporal Encoder consists of a VGG13 model as a spatial encoder, a BiConvLSTM layer (bidirectional convolutional LSTM), and lastly a classifier part constructed by some fully connected layers.

The input frames from each video clip are first resized to 224×224 . Then, the frame difference is calculated by performing subtraction between adjacent frames. In VGG13 network, the fully connected layers and the last 2D max-pooling layer have been truncated (shown in blue and red color). Feature maps derived from the CNN part for each frame (colored orange) are then resized to $14 \times 14 \times 512$. The spatial features given by the CNN part are fed into the BiConvLSTM (green), which generates spatiotemporal encodings for the frames (cyan). To create the final video representation, a max pooling layer is applied to the spatiotemporal features (gold) to shrink the spatial dimensions of the feature maps. A fully connected classifier is then used to classify this video representation as violent or nonviolent (purple).

Their Spatiotemporal Encoder model learns to produce feature maps that have both spatial and temporal information. The temporal encoding works in both directions, allowing future information to be accessed from the current state. They do well in datasets like Hockey, Movies, and Crowds. However, they have a large amount of redundant parameters that makes it inefficient to implement and deploy in real-world applications.

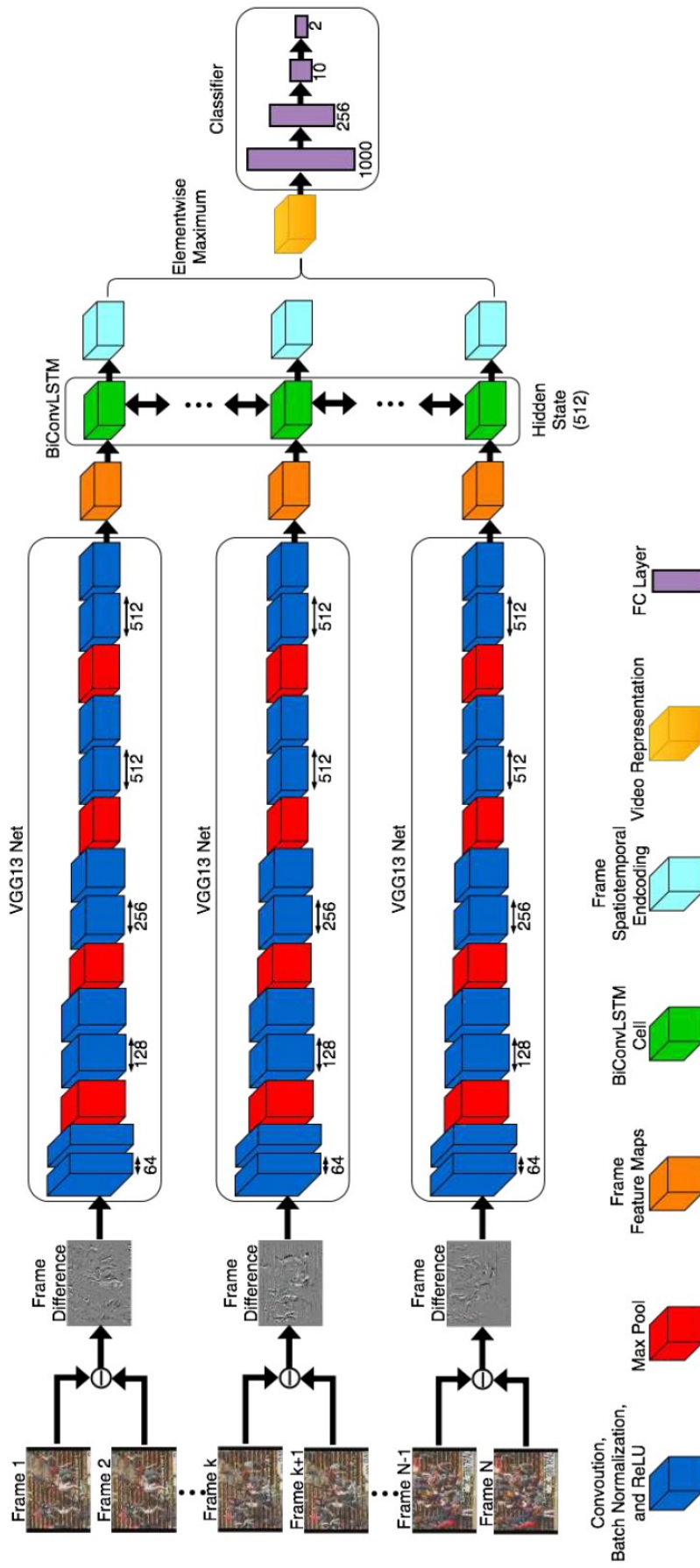


Figure 2.10: A diagram of Spatiotemporal Encoder model architecture proposed by Hanson *et al.* [7] which is comprised of VGG13Net and BiConvLSTM layers.

Chapter 3

Proposed Method

The objective of our proposed approach is to develop an end-to-end trainable deep network that can effectively capture long-range Spatio-temporal features to recognize violent actions while being computationally efficient. To this end, we developed a novel and efficient two-stream network for violence detection. We also developed a simple technique to highlight the body movements in the frames and suppress non-moving background information that promulgates the capture of discriminative features. In this section, we first describe Separable Convolutional LSTM (SepConvLSTM) which is an integral component of our model. We discuss depthwise separable convolution which is utilized in SepConvLSTM. Then, we discuss the input pre-processing steps that are utilized in our pipeline. A description of the architecture of the proposed network, the fusion strategies and fully connected layers are presented.

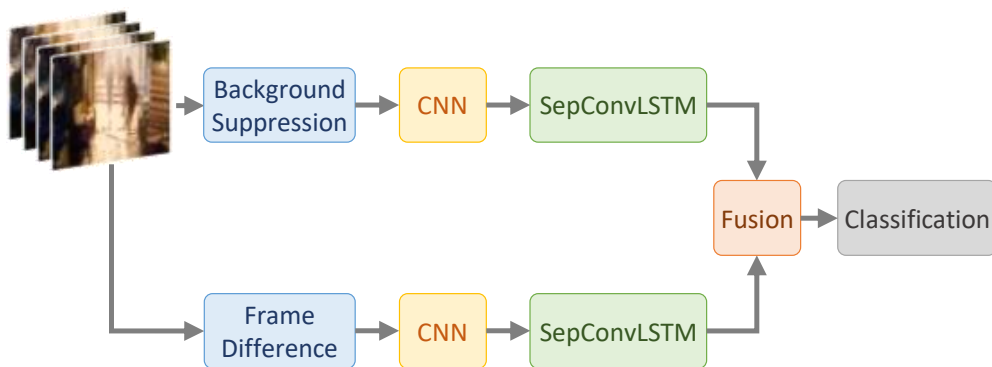


Figure 3.1: Schematic overview of our proposed network

Figure 3.1 illustrates an overview of our proposed methodology for violence detection. We employed a two stream deep learning architecture comprised of CNN and SepConvLSTM layers. We used a fusion layer to combine the outputs of two stream and pass the features into some fully connected layers for classification.

3.1 Pre-processing



Figure 3.2: Input pre-processing for the proposed model. (a) shows key-frames of an example video clip. (b) demonstrates the effect of performing background suppression on video frames of (a). The last row (c) shows time-steps of the frame difference derived from the video clip of (a).

On one stream of our network, we pass the difference of adjacent frames as inputs that promotes the model to encode temporal changes between the adjacent frames boosting the capture of motion information. They were shown to be effective in previous works [7, 20]. Frame differences serve as an efficient alternative to computationally expensive optical flow.

$$fd_i = frame_{i+1} - frame_i \quad (3.1)$$

In equation 3.1, $frame_i$ denotes i th frame and fd_i is the i th time-step of frame difference. A video clip with k frames produces a corresponding frame difference of $k - 1$ time-steps.

On the other stream, instead of using frames directly, we opted to use background suppressed frames. We employed a simple technique to estimate the background to avoid

adding computational overhead. We first calculate the average of all the frames. The average frame mostly contains the background information because they remain unvarying across multiple frames. Then we subtract this average from every frame which accentuates the moving objects in the frame by suppressing the background information. As violent actions like fighting etc. are mainly characterized by body movements and not the non-moving background features, this promotes the model to focus more on relevant information. Equations 3.2 represent this procedure formally.

$$avg = \sum_{i=0}^N \frac{frame_i}{N} \quad (3.2)$$

$$bsf_i = |frames_i - avg|$$

Here, $frame_i$ denotes i th frame, avg is the average of all the frames, and bsf_i is the i th time-step of background suppressed frames that we use as inputs to our model.

Figure 3.2 shows the effect of background suppression and frame difference on video frames. Frame difference mostly encodes temporal information like movements by highlighting the change in body positions. On the other hand, background suppressed frames subdue the background pixels while retaining some textural or appearance-based information of the foreground moving objects.

3.2 Network Architecture

The proposed network comprises two separate streams with the similar architecture. Each stream has a 2D convolutional network that extracts spatial features from each time-step of the clip. An LSTM layer learns to encode these spatial features to generate Spatio-temporal feature maps which are passed to the classification layers. On the first stream, background suppressed video frames are passed sequentially to the model. After all the frames of the input video clip is passed through the CNN, we extract the Spatio-temporal features from the hidden state of the last time-step of the LSTM. The same procedure is followed on the second stream but here we use the difference of adjacent frames as inputs. Frame differences serve as an efficient approximation of optical flow avoiding the computational complexity of calculating optical flow. The frame difference stream learns to encode temporal changes capturing the motion in-between frames while the other stream mainly focuses on spatial appearance-based information. The output features of both streams combined produce robust Spatio-temporal feature maps which are capable of detecting violent activities in videos. Our proposed network is illustrated in figure 3.3.

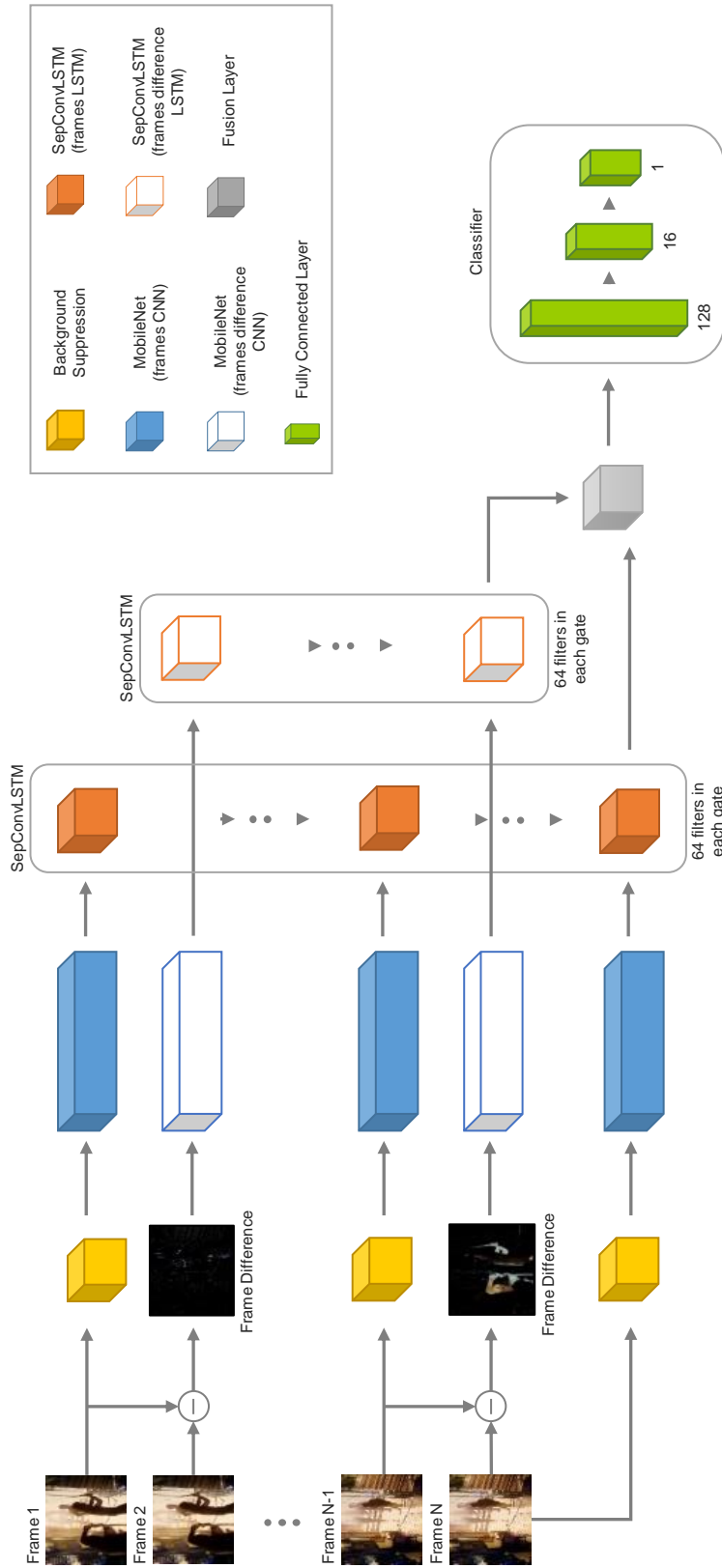


Figure 3.3: The proposed model is composed of two CNN-LSTM streams with similar architecture. Each stream consists of a truncated MobileNet module generating spatial features from each time-step of the inputs. These features are passed to the SepConvLSTM cell in each stream to produce Spatio-temporal encodings. The outputs from each stream are fused using a Fusion layer and passed to the classifier network.

Table 3.1: Summary of the proposed model’s architecture with parameter counts and output shapes. Here, b stands for batch size.

Layer	Output Shape	Param #
frames_CNN	(b, 32, 7, 7, 56)	111984
frames_diff_CNN	(b, 31, 7, 7, 56)	111984
frames_SepConvLSTM2D	(b, 7, 7, 64)	35296
frames_diff_SepConvLSTM2D	(b, 7, 7, 64)	35296
frames_Maxpool2D	(b, 3, 3, 64)	0
frames_diff_Maxpool2D	(b, 3, 3, 64)	0
Fusion	(b, 3, 3, 64)	0
Flatten	(b, 576)	0
Fully_Connected_1	(b, 64)	36928
Fully_Connected_2	(b, 16)	1040
Fully_Connected_3 + Sigmoid	(b, 1)	17

We used MobileNetV2($\alpha = 0.35$) [8] pre-trained on ImageNet dataset [31] as the CNN to extract spatial features where α is the width multiplier. The last 30 layers from the MobileNet models were truncated as we found them to be redundant in our preliminary experiments. Pretraining improves generalization and speeds up training. We use Separable Convolutional LSTM (SepConvLSTM) for producing localized Spatio-temporal features from the output feature maps of the CNN. Previously, SepConvLSTM has been used to speed up video segmentation tasks [30] but have not been explored for action classification tasks. Frames of shape $224 \times 224 \times 3$ are passed into the model. In each stream, the CNN extracts spatial features of shape $7 \times 7 \times 56$. As we used SepConvLSTMs with 64 filters, they output a feature map of shape $7 \times 7 \times 64$ each. After passing through a Max-Pooling layer with window size (2,2), the output features maps from the two streams are fused using a Fusion layer which is presented in the next section. Then, the combined feature maps are passed to fully connected layers for classification. LeakyRelu [32] activation is used in between the FC (fully connected) layers. Finally, binary cross-entropy loss is calculated from outputs of the last layer. We also experimented with one-stream variants of our model to analyze the contribution of each stream. One-stream variants are constructed by simply removing the layers of other stream and the Fusion layer from the proposed model.

3.3 Depthwise Separable Convolution

In the proposed network, both CNN and LSTM parts utilize depthwise separable convolutions. That is why in this section, we explain depthwise separable convolution in

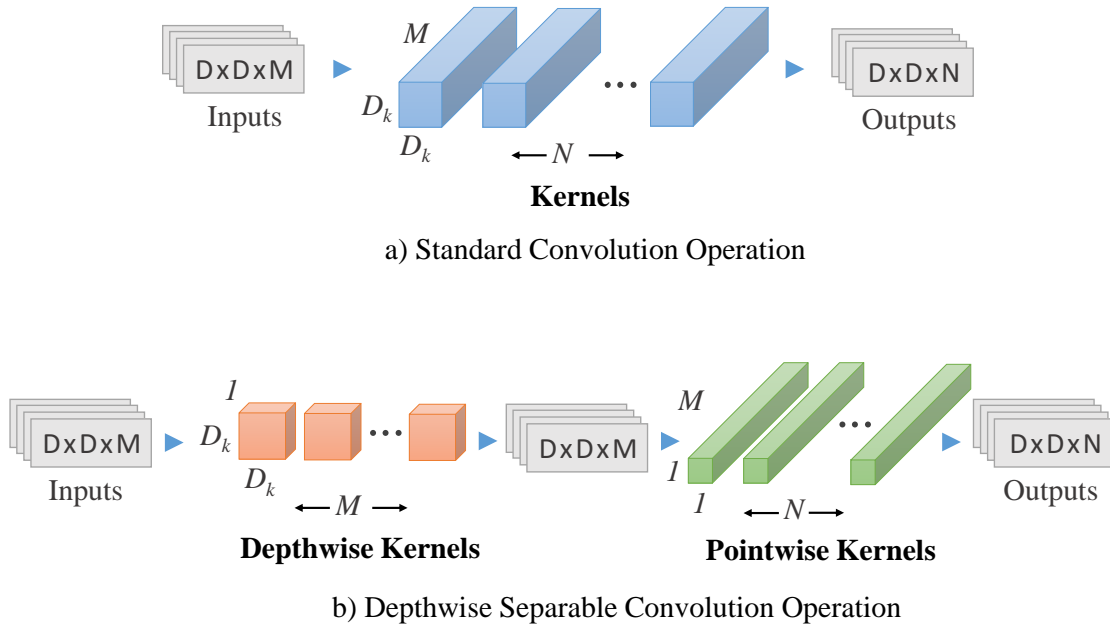


Figure 3.4: SepConvLSTM cell

detail by contrasting it with standard convolution.

Standard convolution operation performs spatial-wise and channel-wise computation in one-pass. But, depthwise separable convolution breaks down the process of convolution into two separate parts. Depthwise separable convolution is an efficient modification of standard convolution operation where one filter is used to perform convolution on each input channel separately to produce an output with the same number of channels. Then, a 1×1 convolution is applied to recombine the information across the channels. This results in a reduction of computation by a ratio of

$$\frac{1}{N} + \frac{1}{D_k^2}$$

where, D_k is kernel size and N is number of output channels [29].

In figure 3.4 the difference between a standard convolution operation and a depthwise separable convolution operation is illustrated. In standard convolution, an input of shape $D \times D \times M$ is transformed into an output of shape $D \times D \times N$ after being convolved with a kernel of shape $D_k \times D_k \times N \times M$. On the other hand, in case of depthwise separable convolution, this process is broken down into two parts. First part is called depthwise convolution. In this part, M kernels with shape $D_k \times D_k \times 1$ is convolved separately with M channel of the input. The output of this operation does not have any cross-channel information. Then using N kernels with shape $1 \times 1 \times M$ the channel information are

intermixed to produce an output of shape $D \times D \times N$. The reduction of computation can be found out from the ratio of kernel shapes of these two operations.

$$\frac{D_k \times D_k \times M \times 1 + 1 \times 1 \times M \times N}{D_k \times D_k \times N \times M} = \frac{1}{N} + \frac{1}{D_k^2} \quad (3.3)$$

This shows that depthwise separable convolution can speed up convolution operation and reduce the number of parameters.

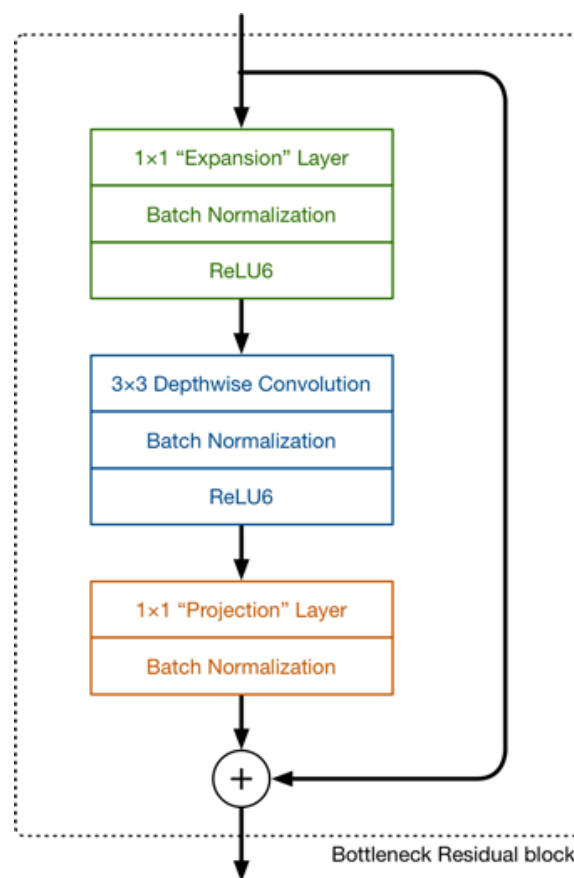


Figure 3.5: bottleneck residual block in MobileNetV2 [8]

3.4 MobileNet

The proposed network utilizes convolutional neural network modules to produce spatial features from each frame. Out of the different variants and designs of convolutional networks we opted to employ MobileNets [29]. MobileNets are light-weight and efficient deep convolutional neural networks which utilizes depthwise separable convolutions drastically reducing the number of parameters and computation without sacrificing much accuracy.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size	
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$	
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$	
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$	
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$	
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$	
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$	
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$	
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$	
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$	
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$	
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$	
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$	
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$	
5×	Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
	Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
	Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
	Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
	Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
	FC / s1	1024×1000	$1 \times 1 \times 1024$
	Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 3.6: Model architecture summary of mobilenet

Figure 3.6 shows the architectural summary of MobileNet model. In the summary, s represents standard convolution and dw signifies depthwise separable convolutions. The size of the model can be controlled using two parameters - α which the width multiplier controlling the number of channels at each layer of the network and ρ which controls the input image size.

We used MobileNetV2($\alpha = 0.35$) [8] as the CNN to extract spatial features where α is the width multiplier which controls the size of the network. For faster training and better generalization we used MobileNetV2 pre-trained on ImageNet dataset [31]. This pre-training significantly boosts the networks ability to learn quickly as the earlier layers of the CNN are already trained to detect shapes, edges etc. low level features. The main

building block of MobileNetV2 is called bottleneck residual block. Instead of a Conv-Relu-BatchNorm block in traditional CNNs, bottleneck residual block is comprised of depthwise convolution, pointwise convolution, Relu activation and residual connection.

3.5 Separable Convolutional LSTM

The proposed network uses a recurrent neural network layer for combining the spatial feature maps of each frame to produce spatio-temporal feature maps. We opted to use separable convolutional LSTM as the recurrent network layer which is an integral component of our model. In this section we describe separable convolutional LSTM in greater detail.

In regular convolutional LSTM [33] layer, each gate of the LSTM is constructed using standard convolution operation. Separable convolutional LSTM is a modification of standard convolutional LSTM. In each gate of the LSTM, the convolution operations are replaced with depthwise separable convolutions. This makes the LSTM layer compact and drastically reduces the number of weights because depthwise separable convolution is an efficient modification of standard convolution operation which results in a reduction of computation compared to a standard convolution operation by a ratio of $\frac{1}{N} + \frac{1}{K^2}$ where, K is kernel size and N is number of output channels [29]. Convolutional LSTM is a good choice to encode temporal changes in a sequence of spatial feature maps as it can preserve spatial information. We replace the convolution operations in the ConvLSTM cell with depthwise separable convolutions which reduces the parameter count drastically and makes the cell compact and lightweight. Equations 3.4 represent the operations inside a SepConvLSTM cell.

$$f_t = \sigma({}_{1 \times 1}W_f^x * (W_f^x \otimes x_t) + {}_{1 \times 1}W_f^h * (W_f^h \otimes (h_{t-1})) + b_f) \quad (3.4)$$

$$i_t = \sigma({}_{1 \times 1}W_i^x * (W_i^x \otimes x_t) + {}_{1 \times 1}W_i^h * (W_i^h \otimes h_{t-1}) + b_i) \quad (3.5)$$

$$\tilde{c}_t = \tau({}_{1 \times 1}W_c^x * (W_c^x \otimes x_t) + {}_{1 \times 1}W_c^h * (W_c^h \otimes h_{t-1}) + b_c) \quad (3.6)$$

$$o_t = \sigma({}_{1 \times 1}W_o^x * (W_o^x \otimes x_t) + {}_{1 \times 1}W_o^h * (W_o^h \otimes h_{t-1}) + b_o) \quad (3.7)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (3.8)$$

$$h_t = o_t \otimes \tau(c_t) \quad (3.9)$$

Here, $*$ denotes convolution, \otimes represents the Hadamard product and \circledast represents depthwise convolution. ${}_{1 \times 1}W$ and W are pointwise and depthwise kernels respectively. Mem-

ory cell c_t , hidden state h_t and the gate activations i_t, f_t and o_t are all 3D tensors. The proposed Seperable ConvLSTM is effective in encoding localized spatio-temporal feature maps which can be used to differentiate between videos containing violence and non-violent actions.

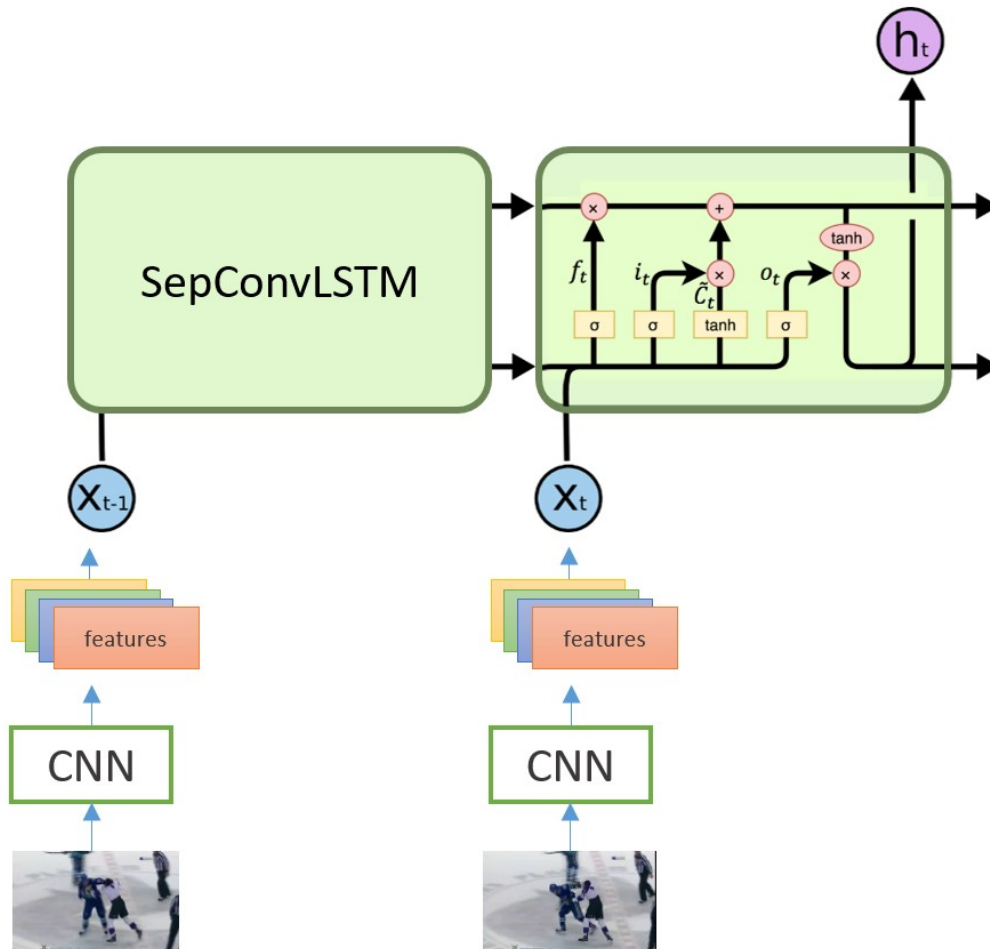


Figure 3.7: SepConvLSTM cell

3.6 Fusion Strategies

We get two sets of feature maps from the two streams of the proposed network. We utilized a fusion layer for merging these two sets of feature maps and passing it into the classifier network. Three fusion strategies were experimented to combine the output feature maps of the two streams. These three strategies produce three variants of our proposed model - *SepConvLSTM-M* (M for Multiply), *SepConvLSTM-C* (C for Concatenation) and *SepConvLSTM-A* (A for Addition). Fusion layers of these three variants are described below.

SepConvLSTM-M: In this variant of our model, the output of the frames streams is passed through a LeakyRelu activation layer. On the other hand, the feature maps from frame difference stream goes through a Sigmoid activation layer. Then, we use an element-wise multiplication to generate the final output feature maps.

$$F_{fused} = LeakyRelu(F_{frames}) \otimes Sigmoid(F_{diff}) \quad (3.10)$$

Here, F_{frames} and F_{diff} denotes the feature maps from frames stream and frame difference stream respectively. F_{fused} is the output feature map of the Fusion layer.

SepConvLSTM-C: In this variant, we simply concatenate the two output features of two streams and pass it to the classification layers.

$$F_{fused} = Concat(F_{frames}, F_{diff}) \quad (3.11)$$

Here, the *Concat* function concatenates F_{frames} and F_{diff} along the channel axis.

SepConvLSTM-A: In the last variant of fusion layer, the output feature maps of the two streams are added element-wise to generate the final video representation.

$$F_{fused} = F_{frames} \oplus F_{diff} \quad (3.12)$$

Here, \oplus refers to element-wise addition operation combining the output feature maps of the two streams.

The output of fusion layer which contains fused feature maps from both stream is flattened into 1D vectors and passed into the classifier network.

3.7 Classifier Network

The proposed network has a classifier network at the end which is comprised of some fully connected, leaky relu and dropout layers.

The output of the fusion layer is flattened into an 1D vector and passed into the classifier network. We used three fully connected layers of shape 128, 16 and 1 respectively. Between each of the two adjacent fully connected layers we placed a Leaky Relu activation layer and a dropout layer. As we used binary cross-entropy loss the output needs to be within the range of 0 to 1. That is why the last fully connected layer is connected to a sigmoid activation layer. The classifier network is illustrated in figure 3.8.

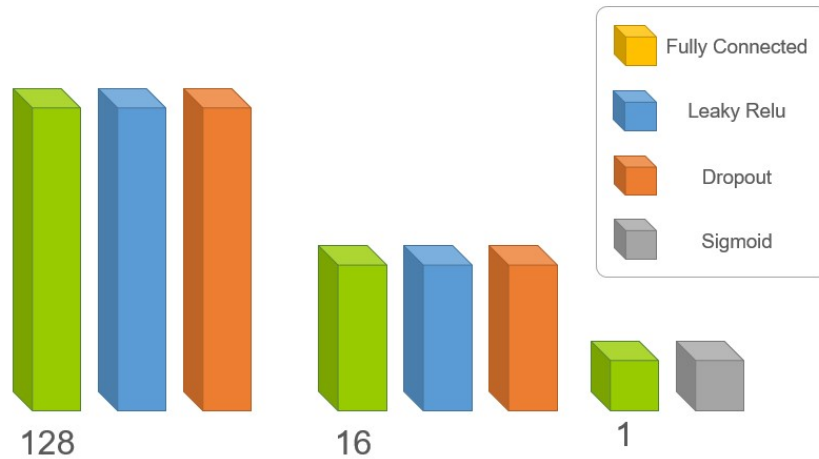


Figure 3.8: classifier network of the proposed network

3.8 Loss Function

Loss function is the measurement which tells the model how it is performance for a given training instance or example. The loss value is propagated backwards using the back-propagation algorithm which is turn updates the weights value of the model. After gradual weight updates through many iterations the model's weights are adjusted to minimize the loss.

We used binary cross-entropy loss as the violence detection task is a binary classification problem. Binary cross-entropy loss is used in binary classification tasks. It penalizes the model if the label predicted by the model does not match the ground truth label. The total loss a batch of training examples can be represented using the following equation -

$$J = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(a^{(i)}) + (1 - y^{(i)}) \log(1 - a^{(i)})]$$

Here, J is the total loss for each batch, m is the number of training examples, $y^{(i)}$ is the target ground truth and $a^{(i)}$ is the value predicted by the model.

3.9 Training Methodology

Adjacent frames in a video tend to contain redundant information. So, we extract only 32 frames from each video using uniform sampling and resize to 320×320 . Before passing onto the model they are cropped with random sizes and resized to 224×224 . This gives us video frames of shape $32 \times 224 \times 224 \times 3$. Performing elementwise subtraction between adjacent frames, we get frame differences of shape $31 \times 224 \times 224 \times 3$. We were restricted to a batch size of 4 due to the limitation of memory. Various data augmentation [34] techniques like random brightness, random cropping, gaussian blurring, random horizontal flipping were employed in the training phase to prevent overfitting.

The proposed model was implemented using Tensorflow library [35]. The CNNs are initialized using weights pre-trained on the ImageNet dataset. We used Xavier initialization [36] for the kernel of SepConvLSTM. Hockey and Movies datasets are very small which can cause overfitting. That's why we first train on the RWF-2000 dataset. Then, we use the weights of this trained model to initialize training on the other two datasets. For model optimization, we used AMSGrad variant of Adam optimizer [37]. We start our model's training with a learning rate of 4×10^{-4} . After every 5 epochs, we reduced the learning rate to half until it reaches 5×10^{-5} . We keep it unchanged since that epoch. The model is optimized to minimize sigmoid loss between the ground truth and the predicted label.

Chapter 4

Result Analysis and Discussion

We evaluate the performance of our proposed models on three standard benchmarks violence detection datasets.

4.1 Datasets

RWF-2000 [1] is the largest dataset on violence detection containing 2000 real-life surveillance footage. Each video is a 5-second clip with various resolutions and a frame-rate of 30 fps. The videos have diverse backgrounds and lighting conditions.

Hockey [2] contains 1000 videos collected from different footage of ice hockey. Each video has 50 frames. All the videos have similar backgrounds and violent actions.

Movies [2] is relatively smaller dataset containing 200 video clips with various resolutions. The videos are diverse in content. The videos with the ‘violent’ label are collected from different movie clips.

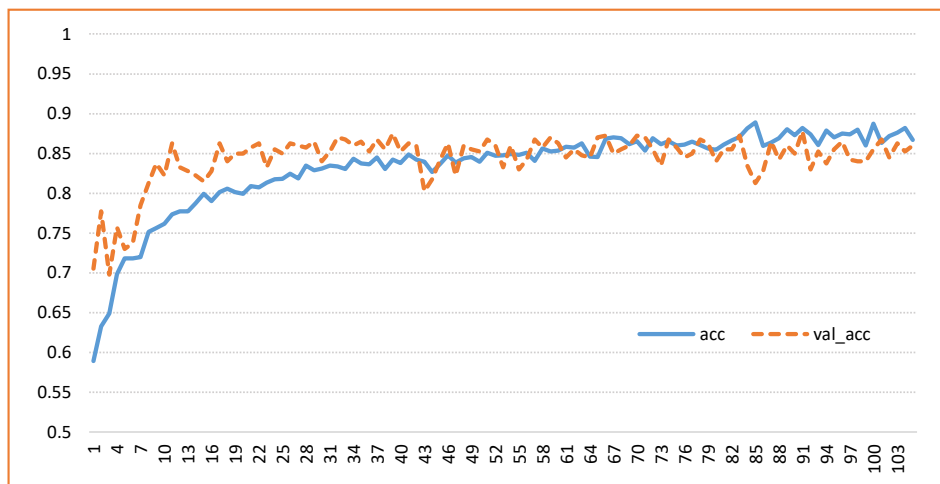
The mentioned datasets contain an equal number of videos containing violent and non-violent action to prevent class imbalance. We found RWF-2000 to be the most challenging one because of its wide variety in its content.

Table 4.1: Comparison of Classification Results on Standard Benchmark Datasets

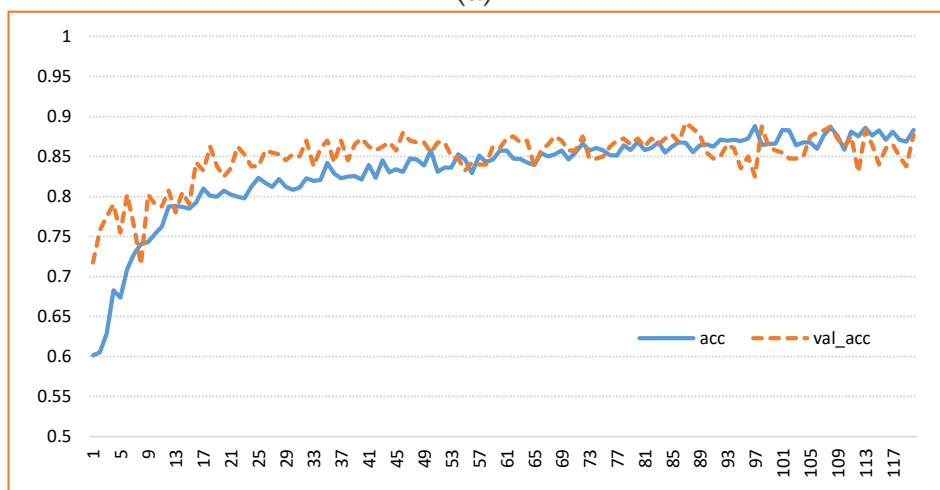
Method	RWF-2000	Hockey	Movies
ViF [21]	-	82.90%	-
ViF + OViF [3]	-	87.50%	-
Radon Transform [22]	-	98.9%	90.1%
Hough Forest + 2D CNN [23]	-	94.6%	99%
Improved Fisher Vector [38]	-	93.7%	99.5%
Three Streams + LSTM [6]	-	93.9%	-
FightNet [39]	-	97.0%	100%
ConvLSTM [20]	-	97.1%	100%
BiConvLSTM [7]	-	98.1%	100%
Efficient 3D CNN [19]	-	98.3%	100%
Flow Gated Net [1]	87.25%	98.0%	100%
Proposed (SepConvLSTM-A)	87.75%	99%	100%
Proposed (SepConvLSTM-C)	89.25%	99.50%	100%
Proposed (SepConvLSTM-M)	89.75%	99%	100%

4.2 Experiment on Standard Benchmark Datasets

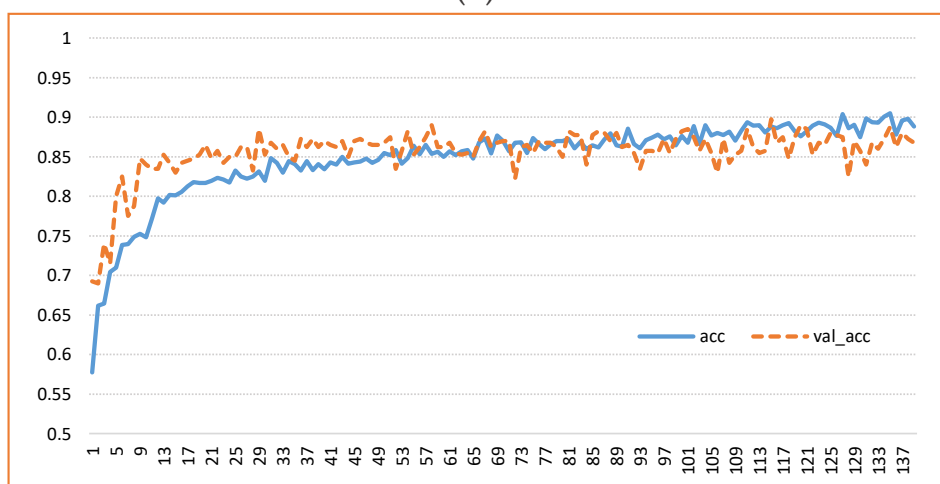
Evaluation of the proposed methods was done on 20% of the dataset. The rest 80% of the clips are used for training our models. From Table 4.1, we can see that newer deep learning methods outperform the earlier methods which focus on extracting hand-crafted features. All three variants of the proposed model outperforms the previous best result on the larger and more challenging RWF-2000 dataset while matching the state-of-the-art results on the smaller datasets. The SepConvLSTM-M model achieved more than 2% margin in terms of accuracy in RWF-2000 dataset which has a fusion strategy of multiplying the LeakyRelu activation of the frames stream with sigmoid activation of the difference stream. In Hockey fights dataset, the SepConvLSTM-C variant of our model performed the best. Out of the three variants, SepConvLSTM-A achieved the lowest accuracy in RWF-2000 dataset which indicates that simple element-wise addition is not as effective as the other fusion strategies. We speculate that the proposed models were able to achieve good performance due to the use of robust and compact modules like SepConvLSTM which mitigates the chances of overfitting, especially when working with datasets that are not large enough. Even though many ambiguous body movements in sports are similar to violent behavior, still the proposed models achieve state-of-the-art accuracy on the Hockey dataset indicating the model’s effectiveness at handling ambiguous movements. The videos on the two categories of the Movies dataset are easily distinguishable. That’s why almost all of the methods achieve very good accuracy on this dataset. Our experiments show that our models can effectively capture Spatio-temporal feature representation to distinguish between violent and non-violent videos.



(a)



(b)



(c)

Figure 4.1: a) Training curve of experimenting with SepConvLSTM-A model. b) Training curve of experimenting with SepConvLSTM-C model. c) Training curve of experimenting with SepConvLSTM-M model. The SepConvLSTM-M model achieved the best accuracy among the three variants of our proposed model which has a fusion strategy of multiplying the LeakyRelu activation of the frames stream with sigmoid activation of the difference stream.

4.2.1 Learning Curves

Training curves shows how a model is performing at every epoch of training. We have plotted the training accuracy and test accuracy against epochs to understand the progress of learning for each model.

Figure 4.1 shows the training curves derived from training three different variants of our model - SepConvLSTM-A, SepConvLSTM-C, SepConvLSTM-M. We can see that accuracy on SepConvLSTM-A model's training is not as consistently high as the others. After about 80 epochs, the test accuracy curves flattens out whereas the train curve is still rising. This points to overfitting. The SepConvLSTM-M gives consistent high accuracy on the test set without overfitting as much as the others. The fluctuations in different epochs are also slightly less.

4.3 Ablation Studies

In deep learning, ablation is the process of removing a component of the model to understand its contribution to the performance of the model. In this section, we present some ablation studies that we used to understand the significance of different components of the proposed model. In the first ablation study, we aimed to find out the individual contribution of each stream to our model's performance. On the other hand, in the second ablation study, we sought to understand out the contribution of SepConvLSTM to the proposed model.

In Table 4.2, we analyze the individual contribution of each stream to our model's performance by evaluating one-stream variants of the model SepConvLSTM-C. Using the variant with only frame difference stream, we get 88.25% accuracy that is better than the previous best result while using only 0.186 million parameters. On the other hand, using the variant with only frames stream, we get an accuracy of 83.75%. The regular variant of SepConvLSTM-C which uses both streams together achieves an accuracy of 89.25%. This indicates that body movements and motion patterns produce more discriminative features than appearance-based features like color, texture, etc.

In Table 4.3, we analyze the contribution of the SepConvLSTM module to the proposed models by replacing it with other modules. Replacing the SepConvLSTM module of the SepConvLSTM-C model with a block of some 3D Convolutional layers, we get an accuracy of only 84% which is much lower than our best performing model. It also increases the number of parameters by a factor of 2. Replacing the SepConvLSTM module

Table 4.2: Analyzing contribution of each stream to our model for violence detection on RWF-2000 dataset

Model	Accuracy	Parameters
SepConvLSTM-C (only frames stream)	83.75%	185,521
SepConvLSTM-C (only differences stream)	88.25%	185,521
SepConvLSTM-C (both streams)	89.25%	371,009

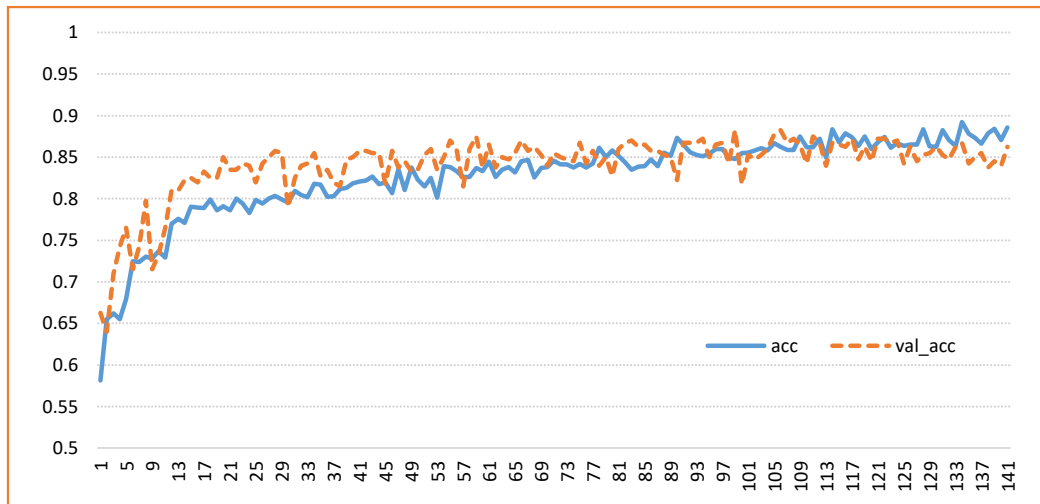
Table 4.3: Analyzing contribution of SepConvLSTM to our model by replacing it with 3D-Conv and ConvLSTM layers

Model	Accuracy	Parameters
Proposed (using 3D-Conv Layers, C Fusion)	84.00%	685,697
Proposed (using ConvLSTM, M Fusion)	87.50%	815,937
Proposed (using ConvLSTM, C Fusion)	88.50%	853,889
Proposed (using SepConvLSTM, M Fusion)	89.75%	333,057

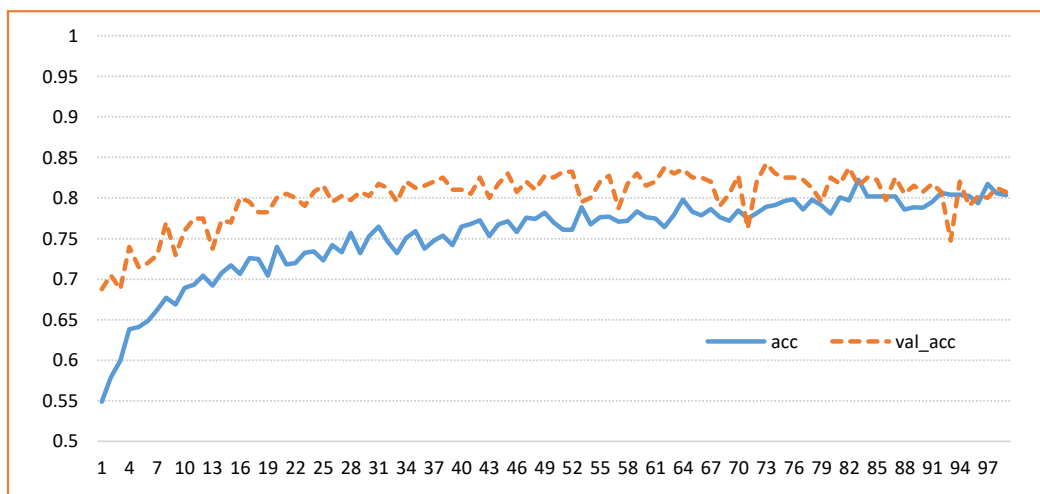
with a regular ConvLSTM module in SepConvLSTM-M and SepConvLSTM-C variants of the proposed model we get accuracies slightly lower than our best performing models. But, using the ConvLSTM module increases the parameter count by a great deal. This indicates that SepConvLSTM is a more efficient and robust choice over ConvLSTM for this particular task.

Figure 4.2 shows the training curves derived from training using the one stream versions of our model. One stream variants of SepConvLSTM-C model can be easily constructed by removing the layers of other stream and fusion layer. Accuracy curve of Difference stream is consistently higher than Frames stream. Difference of adjacent frames serves as a much more discriminative input feature than the frames of the clip themselves indicating that body movements and motion patterns produce more discriminative features than appearance based features like color, texture, etc.

Figure 4.3 shows the training curve of experimenting by replacing SepConvLSTM layer with 3D convolutional layers and by replacing SepConvLSTM layer with ConvLSTM in the model SepConvLSTM-M and SepConvLSTM-C. The lower accuracy and higher parameter count of the model after these modifications indicates that SepConvLSTM is a more efficient and robust choice over these layers.

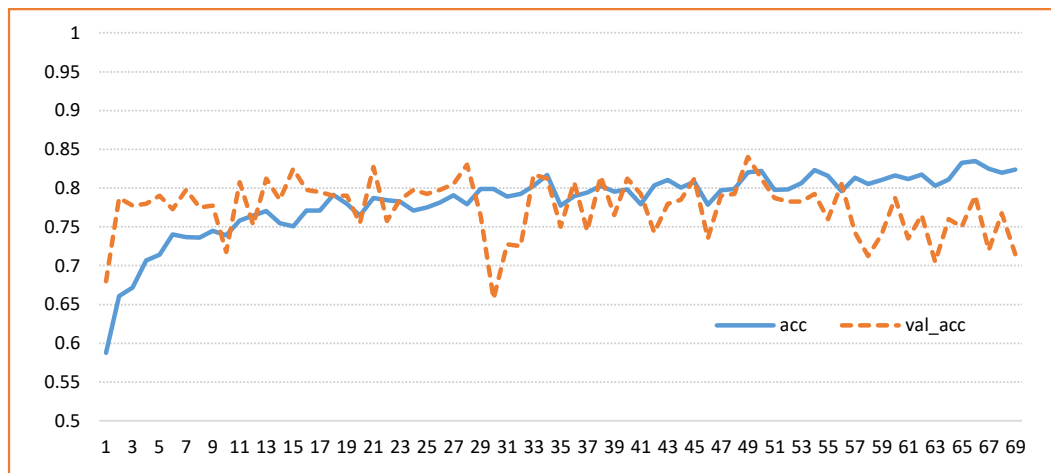


(a)

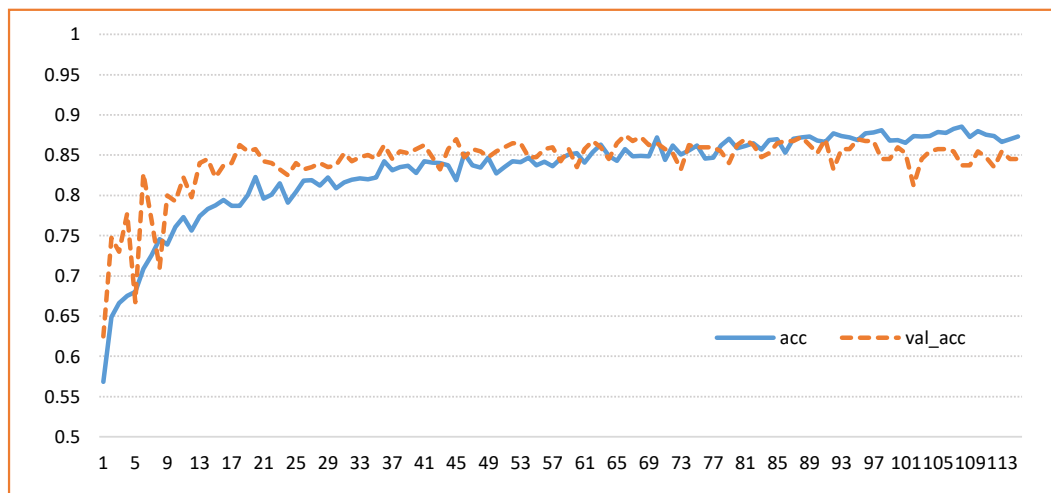


(b)

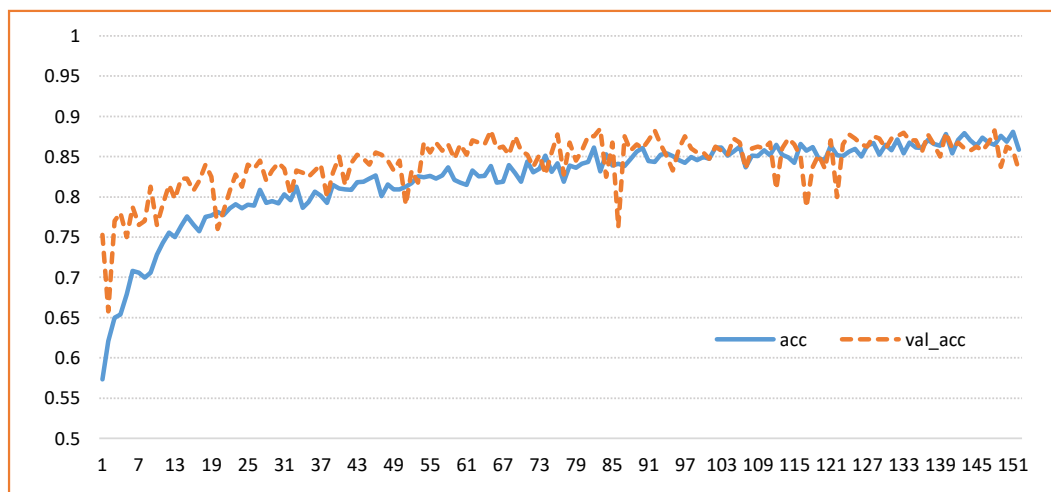
Figure 4.2: a) Training curve of experimenting with only Difference stream of SepConvLSTM-C model. b) Training curve of experimenting with only Frames stream of SepConvLSTM-C model. Accuracy using Difference stream is much higher than using Frames stream only. This indicates that body movements and motion patterns produce more discriminative features than appearance based features like color, texture, etc.



(a)



(b)



(c)

Figure 4.3: a) Training curve of experimenting by replacing SepConvLSTM layer with 3D convolutional layers. b) Training curve of experimenting by replacing SepConvLSTM layer with ConvLSTM in the model SepConvLSTM-M. c) Training curve of experimenting by replacing SepConvLSTM layer with ConvLSTM in the model SepConvLSTM-C. The lower accuracy and higher parameter count of these models indicates that SepConvLSTM is a more efficient and robust choice over these layers.

Table 4.4: Comparison of Efficiency with Earlier Models

Model	Parameters	FLOPs
AlexNet + ConvLSTM [20]	9.6M	14.40G
Efficient 3D CNN [19]	7.4M	10.43G
Flow Gated Net [1]	0.27M	0.54M
Proposed (SepConvLSTM-C, 1 Stream)	0.186M	1.004M
Proposed (SepConvLSTM-C, 2 Streams)	0.371M	2.009M
Proposed (SepConvLSTM-M/A, 2 Streams)	0.333M	1.933M

4.4 Comparative Analysis of Efficiency

To evaluate the efficiency of the proposed model, we compared the number of parameters and FLOPs count with that of the previously proposed models for violence detection.

Table 4.4 shows that our model is significantly more light-weight than previous models. Compared to models proposed in [20] [19], our models have a very low parameter count enabling them to require a drastically fewer number of floating-point operations (FLOPs) and making them faster and computationally efficient. The one-stream variant of our proposed models has the lowest number of parameters. In spite of that, the one-stream variant of SepConvLSTM-C with difference stream achieves an accuracy higher than the previous best results. Flow Gated Net [1] uses only 0.27 million parameters but it uses optical flow as inputs which are computationally expensive to calculate. Whereas, the proposed models are light-weight and do not require any computationally expensive pre-processing on the inputs. The low parameters and FLOPs count will be particularly beneficial if they are deployed for time-sensitive applications or in low-end devices like mobile or embedded vision applications.

4.5 Qualitative Analysis

We demonstrate the qualitative results of the proposed method on the RWF-2000 dataset in Figure 4.4. We used the variant SepConvLSTM-M of our proposed model as it achieved the best performance on the RWF-2000 dataset. In Figure 4.4, each row contains six key-frames from a video clip with a corresponding ground truth label and the predicted label. The first two rows contain examples of video clips for which our model gives a correct prediction. The key-frames of first video clip show that the body positions are not aggressive and the body movements are very slow and minimal. These

are good indicators of the absence of violence in this video clip which enables our model to give correct prediction. On the other hand, the key-frames of the second clip contain fast fighting movements of multiple persons which helps the model to identify it as a violent clip. The last four rows contain examples of failure cases of our proposed model. The key-frames of the third and fifth row contain ambiguous body movements which may cause incorrect prediction. In the key-frames fourth example video clip, a large portion of the bodies of the people involved in fighting is occluded which may cause the network to incorrectly classify the clip as non-violent. The video clip of the last row has very poor quality and resolution. Moreover, the people involved in the fighting are far from the camera. These factors may contribute towards incorrect classification of this clip by our model.

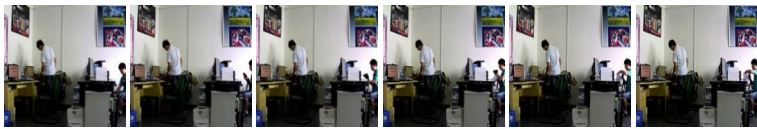



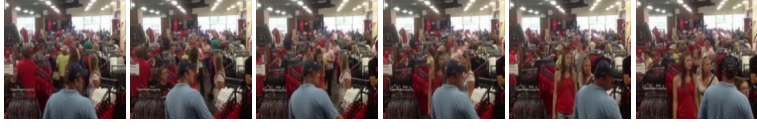
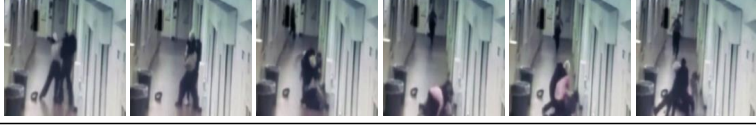
Video Frames	Ground Truth	Predicted Label
	Non-violent	Non-violent
	Violent	Violent
	Violent	Non-violent
	Violent	Non-violent
	Non-violent	Violent
	Violent	Non-violent

Figure 4.4: Qualitative results of the proposed model (SepConvLSTM-M) for violence detection on the RWF-2000 dataset. The first two rows contain examples of video clips for which our model correctly predicts the presence of violence. The last four rows contain examples of failure cases where ambiguous body movements and poor quality of surveillance footage may lead towards incorrect prediction.

Chapter 5

Conclusions

In our works so far, we present a novel and efficient method for detecting violent activities in real-life surveillance footage. The proposed network can learn discriminative Spatio-temporal features effectively which is reflected in its high recognition accuracy in the standard benchmark datasets. Furthermore, it is computationally efficient making it suitable to deploy in time-sensitive applications and low-end devices. We showed that the SepConvLSTM cell is a compact and robust alternative to the ConvLSTM cell. As SepConvLSTM uses fewer parameters, stacking multiple layers of LSTM with residual connections seems feasible and may improve the results further. As the datasets for violence detection are not large enough, pre-training on large-scale action recognition datasets like Sports 1M [40], UCF-101 [41] might help achieve better generalization. Extracting Object-level features from recent object detection deep models such as YOLO [42], Faster R-CNN [43] and adding them as additional input might help, as object-level features inherently focus on relevant objects like people. We hope to investigate such possibilities in the future.

References

- [1] M. Cheng, K. Cai, and M. Li, “Rwf-2000: An open large scale video database for violence detection,” *arXiv preprint arXiv:1911.05913*, 2019.
- [2] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, “Violence detection in video using computer vision techniques,” in *International conference on Computer analysis of images and patterns*, pp. 332–339, Springer, 2011.
- [3] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu, “Violence detection using oriented violent flows,” *Image and vision computing*, vol. 48, pp. 37–41, 2016.
- [4] C. Ding, S. Fan, M. Zhu, W. Feng, and B. Jia, “Violence detection in video by using 3d convolutional neural networks,” in *International Symposium on Visual Computing*, pp. 551–558, Springer, 2014.
- [5] B. Peixoto, B. Lavi, P. Bestagini, Z. Dias, and A. Rocha, “Multimodal violence detection in videos,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2957–2961, IEEE, 2020.
- [6] Z. Dong, J. Qin, and Y. Wang, “Multi-stream deep networks for person to person violence detection in videos,” in *Chinese Conference on Pattern Recognition*, pp. 517–531, Springer, 2016.
- [7] A. Hanson, K. Pnvr, S. Krishnagopal, and L. Davis, “Bidirectional convolutional lstm for the detection of violence in videos,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
- [9] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558, 2013.

- [10] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.
- [11] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, “Temporal pyramid network for action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 591–600, 2020.
- [12] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Actional-structural graph convolutional networks for skeleton-based action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3595–3603, 2019.
- [13] A. Elkholy, M. E. Hussein, W. Gomaa, D. Damen, and E. Saba, “Efficient and robust skeleton-based quality assessment and abnormality detection in human action performance,” *IEEE journal of biomedical and health informatics*, vol. 24, no. 1, pp. 280–291, 2019.
- [14] Q. Lei, H.-B. Zhang, J.-X. Du, T.-C. Hsiao, and C.-C. Chen, “Learning effective skeletal representations on rgb video for fine-grained human action quality assessment,” *Electronics*, vol. 9, no. 4, p. 568, 2020.
- [15] T. Liu, R. Zhao, J. Xiao, and K.-M. Lam, “Progressive motion representation distillation with two-branch networks for egocentric activity recognition,” *IEEE Signal Processing Letters*, vol. 27, pp. 1320–1324, 2020.
- [16] T. Senst, V. Eiselein, A. Kuhn, and T. Sikora, “Crowd violence detection using global motion-compensated lagrangian features and scale-sensitive video-level representation,” *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 2945–2956, 2017.
- [17] D. Chen, H. Wactlar, M.-Y. Chen, C. Gao, A. Bharucha, and A. Hauptmann, “Recognition of aggressive human behavior using binary local motion descriptors,” *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2008, pp. 5238–41, 02 2008.
- [18] T. Deb, A. Arman, and A. Firoze, “Machine cognition of violence in videos using novel outlier-resistant vlad,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 989–994, 2018.

- [19] J. Li, X. Jiang, T. Sun, and K. Xu, "Efficient violence detection using 3d convolutional neural networks," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–8, IEEE, 2019.
- [20] S. Sudhakaran and O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.
- [21] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–6, IEEE, 2012.
- [22] O. Deniz, I. Serrano, G. Bueno, and T.-K. Kim, "Fast violence detection in video," in *2014 international conference on computer vision theory and applications (VIS-APP)*, vol. 2, pp. 478–485, IEEE, 2014.
- [23] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno, "Fight recognition in video using hough forests and 2d convolutional neural network," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4787–4797, 2018.
- [24] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, pp. 568–576, 2014.
- [25] Q. Dai, R.-W. Zhao, Z. Wu, X. Wang, Z. Gu, W. Wu, and Y.-G. Jiang, "Fudanhuawei at mediaeval 2015: Detecting violent scenes and affective impact in movies with deep learning.," in *MediaEval*, 2015.
- [26] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, pp. 802–810, 2015.
- [27] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks. arxiv 2016," *arXiv preprint arXiv:1608.06993*, vol. 1608, 2018.
- [28] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang, "Not only look, but also listen: Learning multimodal violence detection under weak supervision," in *European Conference on Computer Vision*, pp. 322–339, Springer, 2020.
- [29] 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," *CoRR*, vol. abs/1704.04861, 2017.

- [30] A. Pfeuffer and K. Dietmayer, "Separable convolutional lstms for faster video segmentation," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1072–1078, IEEE, 2019.
- [31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [32] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.
- [33] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *arXiv preprint arXiv:1506.04214*, 2015.
- [34] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.
- [35] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.
- [36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.
- [37] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.
- [38] P. Bilinski and F. Bremond, "Human violence recognition and detection in surveillance videos," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 30–36, IEEE, 2016.
- [39] P. Zhou, Q. Ding, H. Luo, and X. Hou, "Violent interaction detection in video based on deep learning," *Journal of Physics: Conference Series*, vol. 844, p. 012044, 06 2017.

-
- [40] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.
- [41] K. Soomro, A. R. Zamir, and M. Shah, “Ucf101: A dataset of 101 human actions classes from videos in the wild,” *arXiv preprint arXiv:1212.0402*, 2012.
- [42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [43] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.