*A Dissertation Report*

*On*

# Violence Detection in Videos using ConvNets and RNN

Submitted in partial fulfillment of the
requirements for the award of course credits for
Master of Technology

Submitted By:

**Rahul Chauhan**

**M.Tech CSE**

**Enrolment No. 17535020**

Under the guidance of

**Dr. Balasubramanian Raman**

(Professor)

**Department of Computer Science and Engineering**

**INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE**

**Roorkee – 247667**

**May, 2019**

# Candidate's Declaration

I hereby declare that the work presented in this dissertation entitled "Violence Detection in Videos using ConvNets and RNN" submitted in the fulfillment of the requirements for the award of the Degree of Master of Technology in Computer Science & Engineering is an authentic record of my own work, carried out during the period from May 2017 to May 2018 under the guidance of Dr. R. Balasubramanian, Professor, Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India. The results embodied in this report have not been submitted by me for the award of any other degree of this or any other Institute/University.

Date: .................
Place: Roorkee

**Rahul Chauhan**
M.Tech. CSE (17535020)
IIT Roorkee

---

# Certificate

This is to certify that the statement made by the candidate in the declaration is correct to the best of my knowledge and belief.

Date: .................
Place: Roorkee

**Dr.R.Balasubramanian**
Professor
Department of CSE
IIT Roorkee

# Acknowledgements

It gives me a great sense of pleasure to present this report on “Violence Detection in Videos using ConvNets and RNN”. I have received enormous help, guidance, and advice from many people. Firstly, I would like to express my utmost gratitude to my supervisor **Dr.R.Balasubramanian** for his constant support and regular guidance throughout the course of my work. His veracity, thoroughness, and persistence have been a constant source of encouragement for me. I am also grateful to the Department of Computer Science and Engineering, IIT Roorkee for providing valuable resources to aid for the research. A hearty thanks to my friends and family for encouraging me in good and bad times.

**Rahul Chauhan**

# Abstract

Detecting Violence in videos automatically and fast is very much essential as it's neither possible nor feasible to continuously monitor a huge database of videos. It's critical to quickly determine violence as it can be crucial in saving lives in real time violence cases in public places or fast detection among millions of files on the internet to flag content, etc. Deep learning techniques have been proven to do well at various tasks as compared to traditional algorithms but require many more resources like huge memory, high computational power, etc. There is a great need to design deep learning architectures which are more suitable to work in a constrained environment. We propose an architecture having Convolutional Neural Network (CNN) for extracting the spatial information, which works as a feature extractor followed by a Recurrent Neural Network (RNN) specifically Gated Recurrent Units (GRUs) to learn temporal cues. Our contribution is proposing an architecture which is simple yet powerful. It uses very fewer parameters (0.85million) without any degradation in performance as compared to state of the art results on benchmark datasets like Hockey Fight, Violent Flows, and Movies. The simplicity of the architecture makes it suitable for low constraint environments having the low computational power and less memory like mobile devices, smart watches, etc.

# Contents

**Candidate's Declaration** i

**Certificate** i

**Acknowledgments** ii

**Abstract** iii

**Contents** iv

**List of Figures** v

**List of Tables** v

1. **INTRODUCTION** 1
   - 1.1 Problem Statement and Research gap . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2. **LITERATURE REVIEW** 3
3. **PROPOSED SOLUTION** 6
   - 3.1 Convolutional Neural Networks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   - 3.2 Gated Recurrent Units as RNN (GRUs) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   - 3.3 Adaptive Moment Estimation (Adam) Optimization Algorithm . . . . . . . . . . . . . . . . . . . . . . . 9
4. **EXPERIMENTS** 10
   - 4.1 Datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
   - 4.2 Experimental Settings . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
5. **RESULTS** 12
6. **CONCLUSIONS AND FUTURE SCOPE** 15

**REFERENCES** 16

## List of Figure

Fig. 2.1 A 3D CNN architecture consists of one hardwired layer, three convolution layers, and two Subsampling layers, and one full connected layer. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Fig. 3.1 Illustration of the architecture of we propose. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

Fig 3.2 Block diagram of a Gated Recurrent Unit (GRU) with reset $r$ and update $z$ gate. . . . . . . . . . . . 8

Fig. 5.1 Minimum and maximum accuracy for Violent Flows Dataset . . . . . . . . . . . . . . . . . . . . . . . . . 12

Fig. 5.2 Minimum and maximum accuracy for Movies Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

Fig. 5.3 Minimum and maximum frequency for Hockey Fight Dataset . . . . . . . . . . . . . . . . . . . . . . . . .13

## List of Tables

Table 2.1 Results of very first 3D CNN on action recognition dataset.. . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Table 5.1 The classification accuracy on the benchmark with previous works done. . . . . . . . . . . . . . . . 14

## Chapter 1

# INTRODUCTION

Video data has become part of our lives and a study by Cisco estimates that by 2022 about 82% of the data on the internet will be composed of IP video traffic [1]. With the ubiquity of mobile device and video cameras this huge increase in video data is obvious hence it's has become necessary to develop robust systems and algorithms that help leverage this huge amount of data. While several work has been done on action recognition but violent action recognition hasn't been much explored. Mob attacks, ravage, fights, snatching, thefts, threats or assaults with a weapon, etc. are only some of the cases where it has become important to monitor people and their activities to detect any violence in it and mitigate the issue without further escalation.

ConvNets are well established to produce great results in the realm of several image processing tasks such as image classification [2], object recognition, segmentation, detection, tracking and so on. A natural question is to ask whether these cutting edge architectures can somehow be used to solve the problems where input is a video instead of an image. Even though the videos can be simply be seen as a sequence of images and we can come up with some naive methods to use these models but it isn't as simple as it seems. Extracting the information from videos has been difficult due to several issues like the motion of cameras, change in point of view, variation in illumination and so on.

Recently great strides are being made to solve action recognition in videos using Convolution Neural Networks (ConvNets) extensions [3, 4]. Such as two – stream ConvNets which uses two different streams having 3D filters to process the input data frames. Where one stream process the data in low-resolution frames and other processes central region of the actual sized frame [3, 4]. Another technique to tackle the issue is by extending the architecture along the entire time axis to create a 3D ConvNet as the videos can be naturally depicted by extending images along time axis [5, 6]. ConvNets tends to work as a feature extractor which can be used to reduce the dimensionality of the input meanwhile keeping the necessary information intact. Recurrent neural networks are [7] naturally good at understanding time series data and were developed to solve problems in which input data exhibit sequential properties such as natural language processing, speech recognition, etc. We explore the ability of these type of networks to process videos which are naturally a sequential data. The main focus of our work is to come up with an architecture which as compared to the previous state of the art is faster to train, simpler to be deployed even in a resource-constrained environment, uses more modern research tools.

## 1.1 Problem Statement and Research gap

Action recognition is the prediction of human activity based on some sensor data in our case from the camera capturing video data. Action as a whole is only meaningful when seen in a sequence. Let us consider a set $\boldsymbol{T}$ containing $\boldsymbol{N}$ videos. $\boldsymbol{T} = \{X_1, X_2, \dots , X_n\}$ with each video $\boldsymbol{X_i}$ labelled $\boldsymbol{Y_i}$ such that $\boldsymbol{Y_i \in \{0 , 1\}}$ Where $\boldsymbol{1}$ being violent , $\boldsymbol{0}$ being non-violent. Now our problem is to design and train a function $\boldsymbol{f}$ that takes an input and tries to correctly evaluate test data as violent or non-violent.

Most of the methods, we studied as shown in [3, 4, 5, 6 ] use 3D ConvNets. Deep learning architectures already deemed to be computationally expensive increases in complexity and cost when 3D convolutional filters are used. These bigger networks take order of magnitude time to train. Other approaches like in [8, 9, 10] are more natural but still used bigger ConvNets resulting in several million parameters. Even though deep architectures have shown better results it should not be taken for granted. We experiment with the architectures of CNN first in order to reduce the total trainable parameters which then decrease the training time, prediction time, memory requirements. [8, 9, 10] uses LSTM as a recurrent unit, but [11, 12] have shown that newer GRUs performing at par with LSTM units on many frontiers leading to fewer parameters hence better training time and prediction time.

## Chapter 2

# LITERATURE REVIEW

Convolutional Neural Networks has long been present and been widely used after their introduction in ILSVRC [2] competition in 2012. In this competition a deep neural network AlexNet was introduced made up of 5 convolutional layers, dropout layers, max polling layers and 3 fully connected layers producing a vector of 4096 dimensions to represent the image. The architecture has a total of 60 million parameters. The results were amazing with top-5 test error rate of 15.3% as compared to 26.2% rate for the next best entry. This work set the foundations for the more fascinating work yet to come in the subsequent years. VGGNet in ISLSVRC 2014 [13] and GoogLeNet by [14] were among the most successful architectures.

Convolutional Neural Networks ability to learn and extract features from the image automatically using the structure of the image and convolutional operation is the key idea to able to process the huge array of pixel data a single image consists of. Success of ConvNets in image classification naturally lead to the idea of using them for video classification though this seems easy at first but the videos are far lot complex then static images and pose several challenges.

Purely using ConvNets for video classification is demonstrated in [3] using a method that comprises two streams of ConvNets. This introduced a novel approach to learn the local motion information. The two streams of processing are fed different kind of inputs from the image frames. Firstly a *context* stream which learns from low-resolution frames and Second a high-resolution *fovea* stream that operates on the middle part of the frame. Time information fusion in ConvNets was shown by using a filter which extends in time dimension also called early fusion or by using two different single frame ConvNets few frames apart and then merging their output in later stage called late fusion or a mixed approach of balancing out both early & late fusion called slow fusion. From (178 x 178 pixels) frame video the context stream is fed a downsampled frame of (89 x 89 pixels) while fovea stream receives the center (89 x 89) region from the original resolution. Both the streams are identical in structure with a concatenation of outputs just before the fully connected layer. The average results for all the various fusion techniques were 63.9% and 82.4% for top-1 and top-5 predictions on then newly introduced Sports-1M dataset.

Another method proposed in [5] is based on early fusion like proposed in [3], it uses a filter that extends in time to perform 3D convolution to learn motion information present in continuous frames. The architecture proposed generated multiple channels of information from continuous frames and did convolution and subsampling for each channel.

It used seven frames of (60 x 40 pixels) as input applies a set of predefined filters to produce 33 feature maps on which 3D convolution filter (7 x 7 x 3) is then applied where (7 x 7 is spatial and 3 is along temporal) followed by 2x2 subsampling layer, next convolutional layer is a result of applying a filter of size (7 x 6 x 3) followed by a (3 x 3) subsampling and a convolution of (7 x 4) producing a 128D vector encapsulating the motion information of 7 frames followed by a fully connected layer which is then fed to a linear classifier.

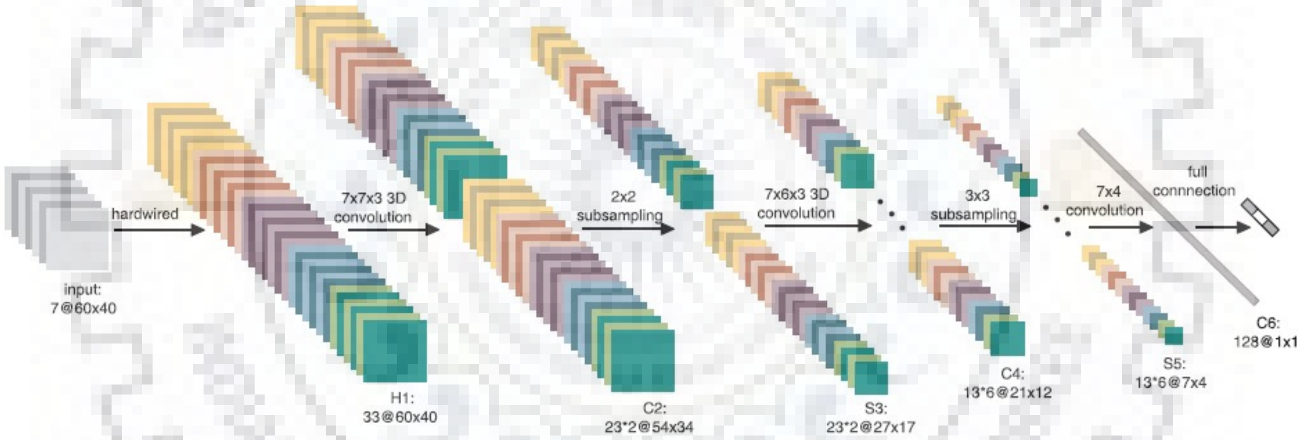


Fig. 2.1 A 3D CNN architecture consists of one hardwired layer, three convolution layers, and two Subsampling layers, and one full connected layer [5].

Following results were obtained on KTH dataset containing six human actions with 2391 sequences recorded with a static camera at 25fps and down-sampled to (160 x 120 pixels)

| | Boxing | Clapping | Handwriting | Jogging | Running | Walking | Average |
|---|---|---|---|---|---|---|---|
| 3D CNN | 90 | 94 | 97 | 84 | 79 | 97 | 90.2 |

Table 2.1 Results of very first 3D CNN on action recognition dataset

[8, 9, 10] Describes the usage of CNN followed by RNN (LSTM) for action recognition tasks. RNNs are suitable for video analysis since they already established to work greatly over time series data and video frames can be depicted as same. The general architecture overview is a composition of a ConvNet followed by RNN, LSTM in these case. Firstly a ConvNet takes a sequence of frames from video and converts it into a sequence of features. Having computed this feature-space representation it is passed to a recurrent neural net (LSTM) to deduce outputs. The ConvNets used here [10] uses pre-trained models like AlexNet or some deeper ConvNets. Retrained networks are known to perform better on various action recognition tasks preventing overfitting.

## Chapter 3

# PROPOSED SOLUTION

Our methodology is composed of two main components a carefully crafted Convolutional Neural Network (CNN) followed by a Recurrent Neural Network (RNN). In a nutshell, we designed an end-to-end trainable architecture for violence detection. First, the ConvNet is fed raw input frames from the videos, let the length of the sequence be ***T*.** Time distributed ConvNet then learns and extracts features from these raw frames and converts every single frame into a ***4096-D*** feature vector [2]. That is, now we end up with a sequence of length ***T*** where each element is a ***4096-D*** feature vector. For recurrent Neural Network, we used Gated Recurrent Units **(GRUs)** [11, 12]**.** This sequence then serves as an input to our recurrent layers of GRUs, which then followed by two fully connected layers and a sigmoid unit terminated by the cross-entropy loss function.

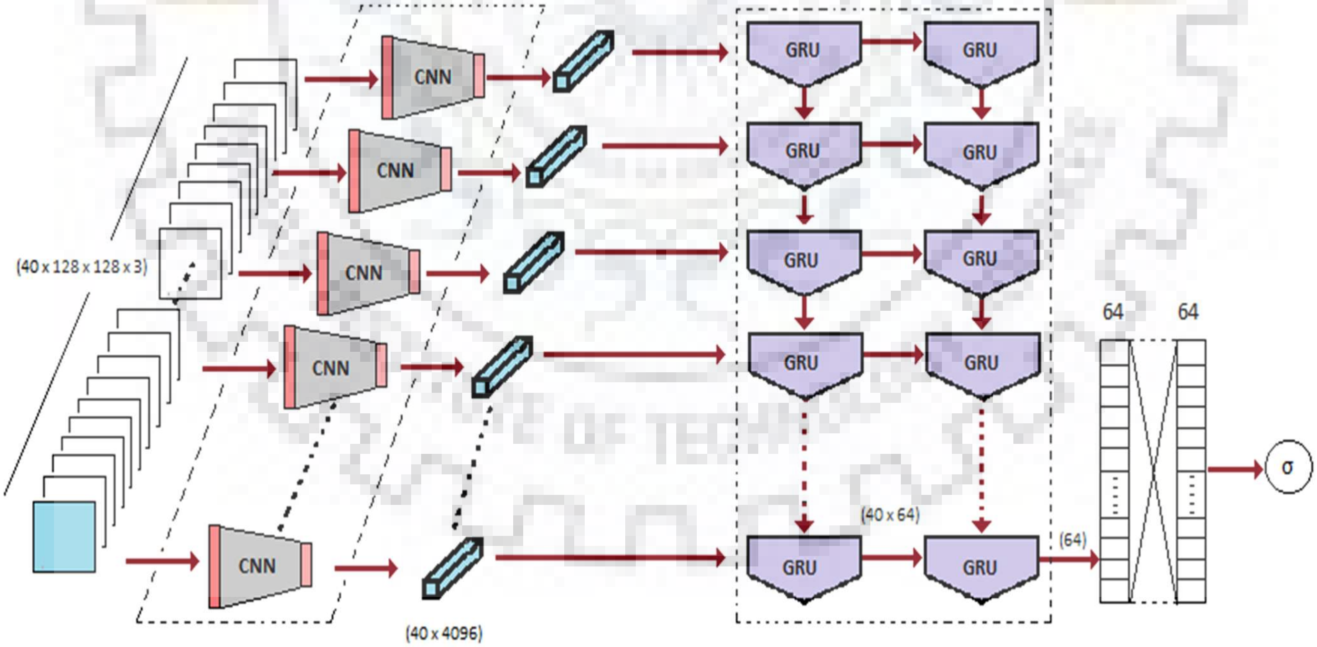


Fig. 3.1 Illustration of the architecture of we propose

### *3.1 Convolutional Neural Network*

ConvNets has been extensively used in several tasks related to image processing success of them has led to advent of new and innovative architectures. The principle of ConvNets is the convolutional operator which exploits the structure of the input of image and uses a kernel having weights which are learning during backpropagation. It passes over the entire input performing element-wise product followed by an added non-linearity, resulting in an output with reduced dimensionality.

Earlier [10] has used Alexnet [2] which is pre-trained with the ImageNet dataset as ConvNet for learning and extracting frame level features. It has ~60 million parameters alone for ConvNet only. Pre-trained deep ConvNets like Alexnet etc has been shown to perform better on several action recognition tasks but here we use our own architecture from scratch following some good practices like introducing max-pooling, using ReLU non-linearity [2], learning several features on the first layer itself and then slowly reducing the dimensionality of the data, etc.

ConvNet in our proposed architecture consist of 4 Convolutional layers and 3 max-pooling layers,

**[INPUT - CONV1 - RELU - MAXPOOL1 - CONV2 - RELU - MAXPOOL2 – CONV3 – RELU - CONV4 – RELU – MAXPOOL3]**

- INPUT (128 x 128 x 3) gets the raw pixel values of image with height, width and channels respectively
- CONV layer uses kernel of size 3 and a padding appropriate so that dimensions of output doesn’t change across height and width.
- RELU layer applies a function ***max (0, x)*** over the output obtained by CONV layer.
- MAXPOOL layer performs down sampling with a window size of (2 x 2), simply takes the maximum value from the window.

The output obtained after MAXPOOL3 is flattened and is a 4096-D vector. Total trainable parameters are 27000 only as compared to 60 million in Alexnet used in [10].

### *3.2 Gated Recurrent Units as RNN (GRUs)*

We use GRUs as our recurrent layer, specifically, GRUs were introduced in [15] as a way to model sequential data and aimed to solve the vanishing gradient problem widespread in an RNN. These can be seen as a variant of Long-Short Term Memory (LSTM) cells [7] but with a more simplified architectural design. [10] uses a recurrent layer of LSTM units following a ConvNet. GRUs has shown equally promising results on various tasks [11, 12], with an added advantage of a simple structure, fewer parameters and lesser training time.

GRUs have two gates an update gate $\boldsymbol{z}$ and a reset gate $\boldsymbol{r}$. Update gate can be seen as a barrier to decide how much of previous memory to keep around and Reset gate is used to set up relevance to previous values.

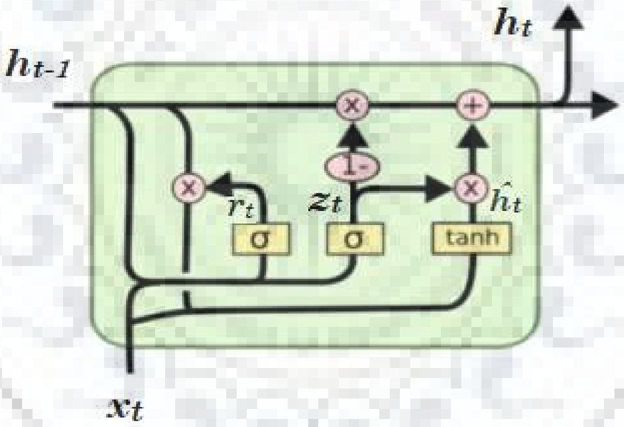


Fig. 3.2 Block diagram of a Gated Recurrent Unit (GRU) with reset $\boldsymbol{r}$ and update $\boldsymbol{z}$ gate

$$z_t = \sigma ( W_z \bullet [ h_{t-1}, x_t ] ) \qquad (3.1)$$

$$r_t = \sigma ( W_r \bullet [ h_{t-1}, x_t ] ) \qquad (3.2)$$

$$\hat{h}_t = tanh ( W \bullet [r_t * h_{t-1}, x_t ] ) \qquad (3.3)$$

$$h_t = ( 1 - z_t ) * h_{t-1} + z_t * \hat{h}_t \qquad (3.4)$$

We use two layers of GRU of size 40 units, each having 64 cells which are fed an input sequence of length 40. With each element of the sequence being a ***4096-D*** vector extracted from raw images, the first layer of GRUs is designed to pass the outputs to next layer of inputs forming a many-to-many model. Second layer used many-to-one model output of second layer is then fed to a two consecutive fully connected layer of 64 each followed by a sigmoid function.

### *3.3 Adaptive Moment Estimation (Adam) Optimization algorithm*

Adaptive Moment Estimation (Adam) [16] is the most popular method for learning gradients and is known for computing adaptive learning rates for each parameter respectively. Along with storing an exponentially decaying average of past squared gradients $v_t$ like RMSprop [17] and it also keeps an exponentially decaying average of past gradients $m_t$ like the momentum. $m_t$ and $v_t$ are estimates of the first moment (the mean) and the second moment (variance) of the gradients respectively.

Equations for computing decaying averages is as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t \qquad (3.5)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g^2_t \qquad (3.6)$$

Bias-corrected first and second moment estimates:

$$\tilde{m}_t = m_t / (1 - \beta^t_1) \qquad (3.7)$$

$$\tilde{v} = v_t / (1 - \beta^t_2) \qquad (3.8)$$

They then use these to update the parameters yielding the Adam update rule on each iteration:

$$\theta_{t+1} = \theta_t - (\eta * \tilde{m}_t) / ( \sqrt{\tilde{v}} + \varepsilon) \qquad (3.9)$$

Proposed default values of the hyper-parameters in Adam is 0.9 for $\beta 1$, 0.999 for $\beta 2$, and $\varepsilon$ $10^{\wedge-8}$ for compares favorably to other moment based optimization algorithms and that works well in practice.

## Chapter 4

# EXPERIMENTS

### *4.1 Datasets*

We evaluated our architecture on three benchmark datasets. Violent Flows Crowd Violence Dataset [18], Movies Dataset [19], Hockey Fight dataset [19] containing videos recorded using CCTV cameras, high-resolution movies, mobile phones, and etc.

i) **Violent Flows Dataset:** This database consists of 246 videos, 123 violent & 123 non-violent, depicting the violent events in which the number of people are very large. These are clips from soccer matches where crowd gets excited and fights with one another.

ii) **Movies Dataset:** It consist of total 200 videos, 100 violent & 100 non-violent. Most of the videos are collected from the scenes in movies having fights.

iii) **Hockey Fight Dataset:** A dataset of 1000 videos, split equally in violent & non-violent type. All the videos are from ice hockey where players get into a fight with one another. Fights pertain interaction of two or three humans at the most wearing jersey of their teams.

### *4.2 Experimental Settings*

We implemented our architecture in Keras running on top of TensorFlow, it is a high-level neural network API, which is written in Python. Using OpenCV library in python we extract 40 frames per video at equally spaced intervals and do embedding with null frames for videos which have less than 40 frames. We tried other values for frame size like 32, 64 and 256 pixels. Whereas images with lower resolution tend to allow for a bigger batch size but only on the cost of lesser features. On the other hand a frame size of 256 tends to reduce batch size hence incurring high training time. We select a frame size of (128 x 128) pixels providing us a balance of both features and batch size.

As a preprocessing step, we normalized the input pixels to a range between 0 and 1. We tried both zero centering and min-max normalization, latter was shown to have improved performance. The network is trained end-to-end using Adam algorithm [16] with hyper-parameters, learning rate = 0.0005, beta_1 = 0.9, beta_2 = 0.999. Weights of the entire network is initialized using Glorot uniform initializer also called Xavier uniform initializer. We hyper-tuned our model for the best batch size and number of epochs and found out a number of epochs resulting in best accuracy were 25, 5, 10 respectively for the datasets (i) (ii) and (iii) respectively. For batch size we tried batch size of 8, 16 and 32 videos. Not to mention this translates to 320, 640 and 1280 images per frame, batch size of 16 seems to work best giving as lesser time. Our learned models are stored in .h5 type files for each fold for each dataset. System configuration on which network was trained and tested is as follows,

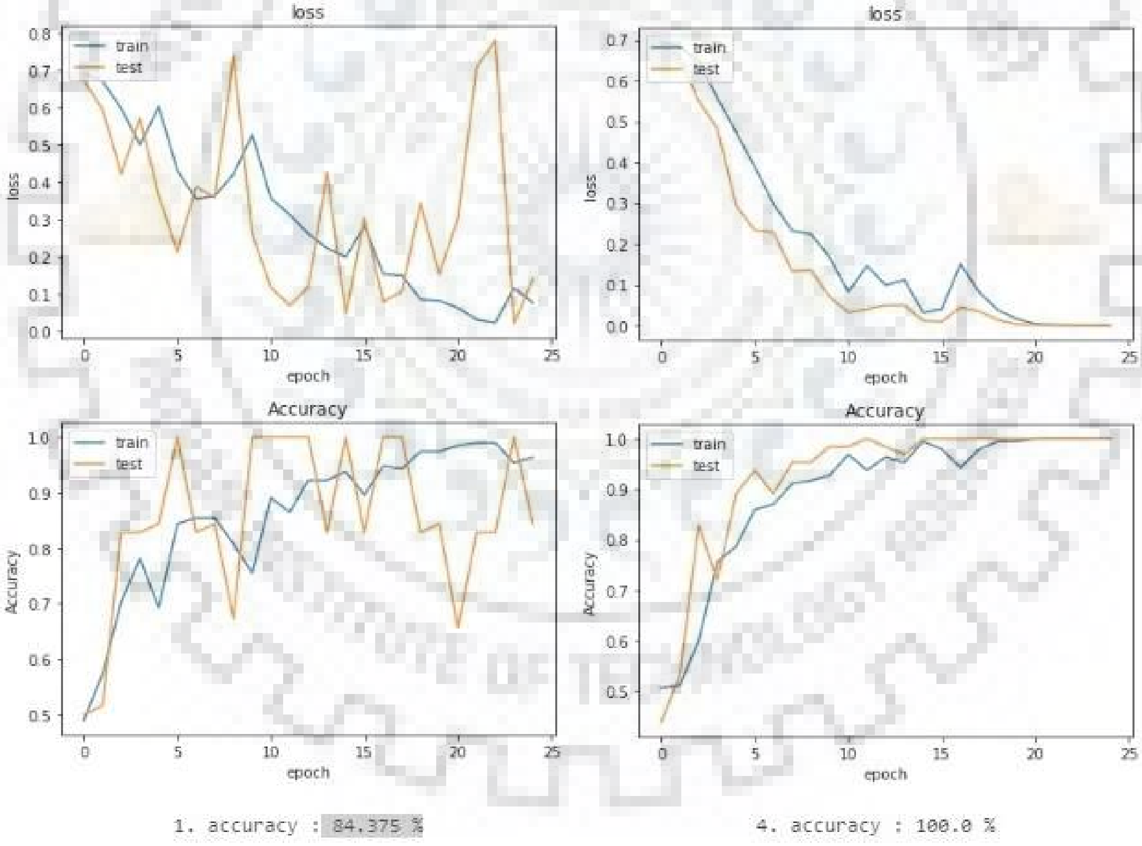
CPU: Intel(R) Xeon(R) CPU @ 2.30GHz
RAM: 13 GB
GPU: Tesla T4, 16 GB

## Chapter 5

# RESULTS

5-fold cross-validation technique is used to evaluate the architecture performance as used in literature. For each dataset, we graph the accuracy and loss achieved on both train and test set for minimum and maximum accuracy achieved during 5 iterations. We explicitly give the accuracy value on test set for respective fold.

### i) Violent Flows Dataset

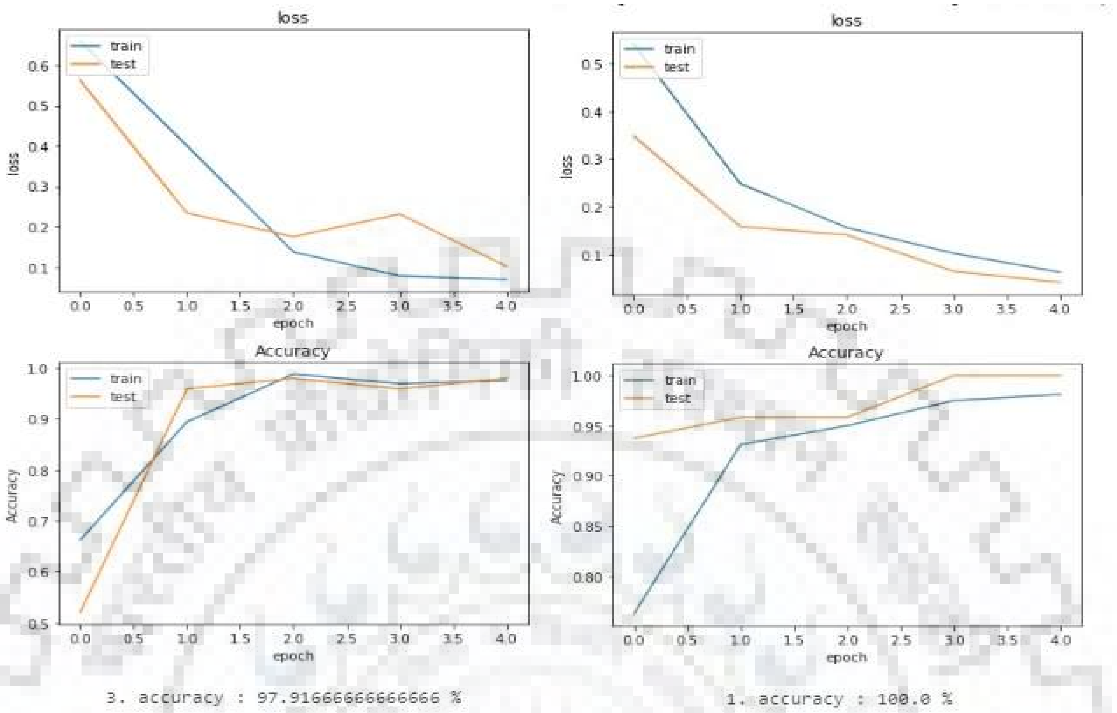


```
1. accuracy : 84.375 %
```

```
4. accuracy : 100.0 %
```

Fig. 5.1 Minimum and maximum accuracy for Violent Flows Dataset

### ii) Movies Dataset:

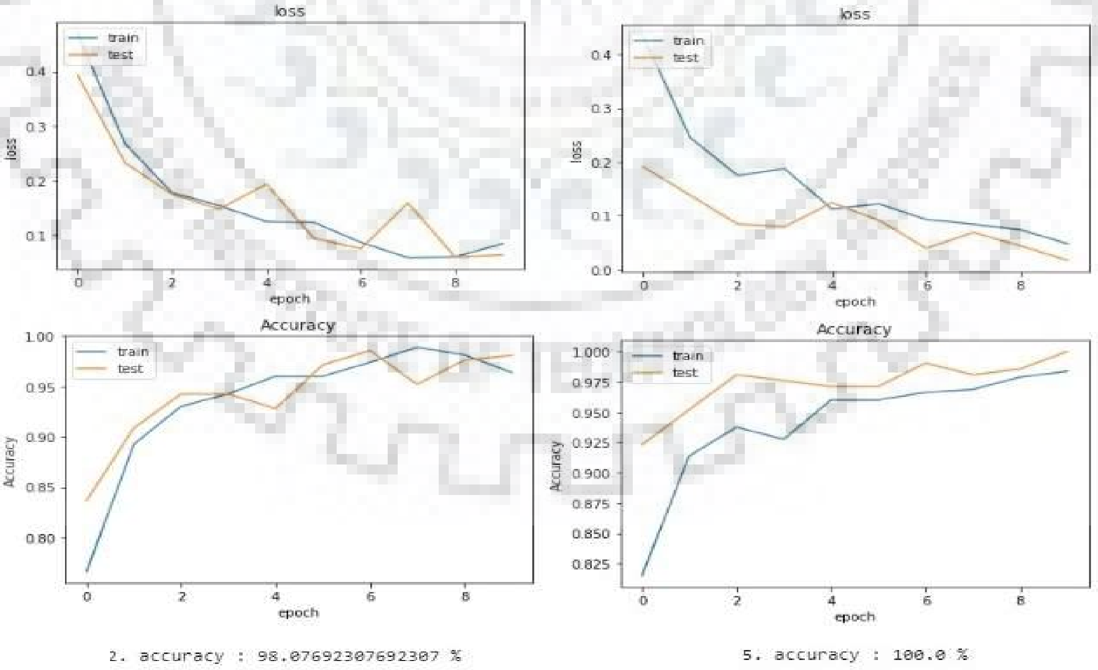


```
3. accuracy : 97.91666666666666 %
```

```
1. accuracy : 100.0 %
```

Fig. 5.2 Minimum and maximum accuracy for Movies Dataset

### iii) Hockey Fight Dataset:



```
2. accuracy : 98.07692307692307 %
```

```
5. accuracy : 100.0 %
```

Fig. 5.3 Minimum and maximum frequency for Hockey Fight dataset

| Method | Violent Flows | Movies | Hockey Fight |
|---|---|---|---|
| MoSIFT + HIK | - | 89.5% | 90.9% |
| ViF | 81.3±0.21% | - | 82.9±0.14% |
| MoSIFT+KDE+Sparse Coding | 89.05±3.26% | - | 94.3±1.68% |
| Deniz et al | - | 98.0±0.22% | 90.1±0% |
| Gracia et al | - | 97.8±0.4% | 82.4±0.4% |
| Substantial Derivative | 85.43±0.21% | 96.89±0.21% | - |
| Bilinski et al. | 96.4 | 99 | 93.4 |
| MoIWLD | 93.19±0.12% | - | 96.8±1.04% |
| ViF+OViF | 88±2.45% | - | 87.5±1.7% |
| Three streams + LSTM | - 93.9% | - | 93.9 |
| ConvLSTM | 94.57±2.34% | 100±0% | 97.1±0.55% |
| **Proposed(Accuracy, std. dev)** | **96.88% , 6.2** | **99.17%, 1.02** | **99.04%, 0.68** |

Table 5.1 The classification accuracy on the benchmark with previous works done [10].

[10] is the closest to our methodology but uses a pre-trained ConvNet (AlexNet) followed by a LSTM layer resulting in a total of ~68 million trainable parameters. Our proposed architecture gives best known results for Hockey Fight dataset and is at par with best results on Violent Flows and Movies dataset. With nearly 0.85 million parameters there is an 80 fold reduction in model size as compared to [10] yet at par with the state of art results.

## Chapter 5

# CONCLUSIONS AND FUTURE SCOPE

Violence Detection is a pressing issue and requires systems that are easy to train and are able to perform well in resource constraint environment. Our study comes up with such an architecture. It clearly shows the superiority of ConvNet-GRU architecture over other methods which is suitable since videos are necessarily a time series kind of data. It also indicates that having a deep ConvNet is fine but not necessary to produce great results and the same can be achieved by carefully crafted custom architecture. The competitiveness of GRUs against LSTMs is again supported by our study by observing no loss in accuracy as compared to other LSTM based methods The sheer reduction in the total parameters makes our model easy to be deployed in resource constraint environment such as mobile devices, smart watches, etc. Generative Adversarial Networks (GANs) are gaining widespread popularity and since video data is particularly less for violence detection they can be used to generate short samples to be used to train architectures as such. Videos in existing benchmark are limited background and kind of similar so more diverse datasets are required. Other architectures must be tried for ConvNet as a feature extractor with various depths and widths. Efforts should be made to create lite architectures first and building upon complexity rather than choosing a deeper net beforehand.

# REFERENCES

[1] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet classification with deep convolutional neural networks”. In Proc. NIPS, pages 1097–1105, 2012.

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. “Large-scale video classification with convolutional neural networks”. In Proc. CVPR, pages 1725–1732, 2014.

[4] K. Simonyan and A. Zisserman. “Two-stream convolutional networks for action recognition in videos”. In Proc. NIPS, pages 568–576, 2014.

[5] Shuiwang Ji, Wei Xu, Ming Yang and Kai Yu. “3D Convolutional Neural Networks for Human Action Recognition”. In IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 221 – 231,Vol 35, 2013.

[6] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani and Manohar Paluri. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In CVPR, pages 4489-4497 , 2014.

[7] Sepp Hochreiter and Jurgen Schmidhuber. “Long-Short Term Memory”. In Neural Computation, Vol 9, pages 2039-2041, 1997.

[8] Jeff Donahue, Lisa Anne Hendricks et al. “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”. In CVPR, pages 677 – 691, 2014.

[9] Joe Yue-Hei Ng, Matthew Hausknecht et al. “Beyond Short Snippets: Deep Networks for Video Classification”. In CVPR, pages 1725-1732, 2015.

[10] Swathikiran Sudhakaran, Oswald Lanz. “Learning to Detect Violent Videos using Convolutional LSTM”. In CVPR, arXiv 1709.06531, 2017.

[11] Junyoung Chung, Caglar Gulcehre et al. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In CVPR, arXiv:1412.3555, 2014.

[12] Rafal Jozefowicz, Wojciech Zaremba, Ilya Sutskever. “An empirical exploration of recurrent network architectures”. In ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, pages 2342-2350, 2015.

[13] Karen Simonyan, Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In CVPR, arXiv:1409.1556, 2014.

[14] Christian Szegedy, Wei Liu et al. “Going Deeper with Convolutions”. In CVPR, arXiv:1409.4842, 2014.

[15] Kyunghyun Cho, Bart van Merrienboer et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In Computation and Language, pages 1724–1734 2014.

[16] Diederik P. Kingma, Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In international Conference for Learning Representations (ICLR), arXiv:1412.6980, 2015.

[17] https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[18] Hassner, Y. Itcher, and O. Kliper-Gross, Violent Flows: Real-Time Detection of Violent Crowd Behavior. In Computer Vision and Pattern Recognition (CVPR), pages 1-6, 2012

[19] E. Bermejo, O. Deniz et al. “Violence Detection in Video Using Computer Vision Techniques”. In international conference on Computer analysis of images and patterns - Vol II, pages 332-339 2011.