Using Deep Learning Algorithms to Detect Violent Activities

by

S.M. Rojin Ammar
15101026
Md. Tanvir Rounak Anjum
16301140
Md. Touhidul Islam
15301133

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
May 2019

© 2019. BRAC University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

S.M. Rojin Ammar
15101026

Md. Tanvir Rounak Anjum
16301140

Md. Touhidul Islam
15301133

Approval

The thesis/project titled "Using Deep Learning Algorithms to Detect Violent Activities" submitted by

1. S.M. Rojin Ammar (15101026)
2. Md. Tanvir Rounak Anjum (16301140)
3. Md. Touhidul Islam (15301133)

Of Spring, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 16, 2019.

Examining Committee:

Supervisor:
(Member)

Md. Ashraful Alam, PhD
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Md. Abdul Mottalib, PhD
Professor and Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

It is of extensive importance to develop a technique for automatic surveillance video analysis to recognize the presence of violence. In this work, to identify violent videos, we put forward a deep neural network. For extracting frame level features from a video, a convolutional neural network is used with a pre-trained ImageNet model. The characteristics of the frame level are then aggregated using a long short-term memory variant that uses fully connected layers and leaky rectified linear units. Together with the long short-term memory, the convolutional neural network is capable of capturing localized spatio-temporal features that enable the analysis of local motion in the video. The performance is further evaluated in terms of accuracy of recognition on three standard benchmark datasets. In order to determine the capabilities of our proposed model, we also compared our system results with other techniques. The approach proposed outperforms state-of - the-art methods while processing the videos in real time.

Keywords: Machine Learning; Violent Activities Detection; Convolutional Neural Network; Recurrent Neural Network; Long Short-Term Memory; DarkNet-19

Dedication

We would like to dedicate this thesis to our loving parents, friends and everyone that helped us with this paper

Acknowledgement

We would like to thank Dr. Md. Ashraful Alam for agreeing to supervise us with our thesis. His patience and confidence in us has been a source of encouragement and this thesis would not have been possible without the great support, inspiration and influence of him. We believe his dedication to this paper deserves to be reciprocated with great gratitude. We also would like to give a special thanks to our friend Md. Tahmid Hossain for his help and motivation with the research. A special thanks to the Thesis committee for taking the time to review and evaluate our thesis as part of our undergraduate program.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$BoVW$ Bag of Visual Word

CNN Convolutional Neural Network

$GMOF$ Gaussian Optical Flow Model

iDT Improved Dense Trajectories

KDE Kernel Density Estimation

KNN K-Nearest Neighbors

$LReLU$ Leaky ReLU

$LSTM$ Long Short-Term Memory

$MoSIFT$ Motion SIFT

$OHOF$ Optical Flow Orientation Histogram

$OViF$ Oriented Violent Flows

RF Random Forest

RNN Recurrent Neural Network

$STIP$ Space-Time Interest Point

STV Spacetime Volume

SVM SupportVector Machine

ViF Violent Flows

Chapter 1

Introduction

The technological advancement in video and image processing has been unprecedented due to their importance in finding out intricate contents for various applications and purposes which includes search, summarization and recognizing action. The emphasis on recognizing actions from video stream has been growing in recent years due to the rise of violent acts involving terror groups to various single or multiple person attacks. This has resulted into the usage of surveillance cameras throughout the world on an ubiquitous level. The footage that these cameras streams are manually inspected for such violent acts by humans all the time which is really tenacious and not feasible in the long run in order to scale this operation. Also, the process of detecting such scenarios can be error prone due to the fact that humans can make mistakes or they might not catch a significant event due to inspecting other feeds. There's been millions of cameras around the world for surveillance purposes but even if the error percentages are low still the dangers are there for thousands of people. This is just an estimation which allowed us to think about the current situation of violence detection systems and methodologies. This calls for certain measures and fast detection of violence without the help of humans. Hence we turn towards deep learning methods which are able to learn by itself and it is required as it can be used effectively to detect potential violent activity before anyone.

The technology to detect objects and movements has a come a long way in terms of developments and allows us to merge these technologies to create a system which can detect potential violent activities effectively that happens in our everyday life.

This is where our system comes in which proposes to use deep learning algorithms in order to detect violent activity automatically. This involves various stages of process such as object detection, action detection and video classification. In recent years there has been research conducted by academics and also many companies are eyeing to build a system which detects violent activities automatically through the videos[2], [8], [35], [48]. There's been great deal of advancement too. We are creating a system with techniques that will help us detect violent activities without the help of manual detection or a presence of a human. In this system we propose methodologies that will be able to recognize violent threats and activities using deep learning methodologies. We have used Convolutional Neural Network(CNN), Recurrent Neural Network(RNN) along with Long Short-Term Memory(LSTM) and various methods which made our system validate its action recognition techniques.

Our system will be able to detect violent activities seamlessly from video streams or recorded videos. Firstly, We have to take video inputs and put these inputs through the Neural Network and get an output using deep learning methods which tells us if the actions are violent or not. We had to go through many trial and error because detecting actions and differentiating between non-violent activities and violent activities are still a major setback. But we have tried to make it accurate as possible. Our paper is divided into following chapters: Chapter 2 directly plunges into some of the work related to our research such as Action Recognition, violence Recognition. Chapter 3 highlights and gives an introduction on the algorithms, datasets and Libraries that we have used for our research and to create our system. This is followed by chapter 4, which describes the proposed design created for the system, its proposed interface, working process and an explanation on the algorithm and functions we have used. Chapter 5 explains the implementation of the system - starting from the experimental setup, accuracy evaluation and gradient clipping. The results are finally obtained in chapter 6, and the conclusions are drawn along with potential future implementations in chapter 7.

Chapter 2

Literature Review

2.1 Action Recognition

A sequence of images makes up a video and the system should be able to recognize the location of humans and understand how human motion is changing with time in each video frame. Action recognition task involves identifying various actions from video clips (a 2D frame sequence)[76].The action may or may not be performed throughout the video's entire duration. This appears to be a natural extension to multiple frames of image classification tasks and then aggregating the predictions from each frame. For the image frame feature extraction from a video, multiple approaches are observed. Ding et. al used 3d convolutional layers to extract the spatial and temporal features whereas fully-connected layers are applied to classify them, which will be expounded on later[24]. Over the last decade, several works have been presented on action recognition by researchers. These works can be classified into mainly two categories:

1. Hand-crafted
2. Deep-net

Most of the traditional Computer Vision algorithm variants for action recognition could be broken down into the following 3 broad steps: local high-dimensional visual characteristics that describe a region of the video are either densely extracted[16] or at a sparse set of points of interest[9]. The extracted characteristics are combined into a video level description of a fixed size. One popular variant of the step is to bag visual words (derived from hierarchical or k-means clustering) for video-level encoding features. A classifier, such as Support Vector Machine(SVM) or Random Forest(RF), is trained for final prediction on a bag of visual words These algorithms that use shallow hand-crafted features in Step 1, improved Dense Trajectories[28] (iDT) that use densely sampled trajectory features was the state-of - the-art. At the same time, in 2013, 3D convolutions were used without much help as is for action recognition[21]. Soon after this, two breakthrough research papers were published in 2014 . The main differences between them were the choice of design to combine spatiotemporal information. The earlier work was based on hand-crafted features for non-realistic actions, where an actor used to perform certain actions in a simple background scene[76]. These systems extract low-level features from the video data and then feed them for action recognition to a classifier such as support vector

machine (SVM), decision tree, and K-Nearest Neighbors(KNN). For example, Yilmaz and Shah analyzed the geometric properties of spacetime volume (STV) called action sketch[10]. By capturing STV direction, speed, and shape for action recognition, they stacked body contours in time axis.

In addition to hand-crafted features based action recognition approaches, several methods based on deep learning have also been proposed in recent years but despite the stratospheric success of ImageNet deep learning architectures, progress in video classification and representation learning architectures has been slower. In many areas such as image classification, person re-identification, object detection, speech recognition, and bioinformatics, profound learning has shown significant improvement[64]. Authors Karpathy et al. explore multiple ways to use2D pre-trained convolutions to fuse temporal information from consecutive frames[27]. Consecutive video frames are presented in all configurations as input. Single frame uses a single architecture that at the last stage fuses information from all frames. Late fusion uses two networks with shared parameters, separating 15 frames, and combining predictions at the end as well[76]. Early fusion combines over 10 frames in the first layer. Slow fusion involves multi-stage fusion, an early-to-late fusion balance. Multiple clips were sampled from whole video for final predictions and averaged prediction scores from them for final prediction. The authors found that the results were significantly worse than state-of - the-art hand-crafted feature-based algorithms despite extensive experimentation[76]. To capture local spatio-temporal information, a multi-resolution CNN framework for connectivity of time domain features is proposed. This method is evaluated experimentally on a new 487 class "YouTube 1 million video dataset." The authors claimed that CNN's foveated architecture had accelerated the training complexity. They improved the recognition rate for large datasets up to 63.9 percent, but their recognition rate on UCF101 is 63.3 percent, which is still too low to recognize such important task of action. Multiple reasons for this failure were there as the learned spatiotemporal features did not capture motion features The data set was also less diverse and it was difficult to learn such detailed features. The authors, Simmoyan and Zisserman[28], build on the failures of Karpathy et al's previous work. [28] proposed a two-stream CNN architecture in which the first stream captures spatial and temporal information between frames and the second shows the dense optical flow of multiple frames. By combining two datasets, they increased the amount of data for the CNN model training. Given the toughness of deep architectures to learn movement characteristics, authors explicitly model movement characteristics in the form of stacked optical flow vectors. Therefore, instead of a single spatial context network, this architecture has two separate networks-one for spatial context (pre-trained), one for context of motion[76]. The spatial net input is a single video frame. Authors experimented with the input to the temporal net and found that the best performance was bi-directional optical flow stacked across 10 successive frames. The two streams were trained and combined separately using SVM. The final prediction was the same as the previous paper, i.e. the sampled frame average. Although this method improved the performance of single stream method by explicitly capturing local temporal movement. There were still some drawbacks, because video level predictions were obtained from average predictions over sampled clips, the long-range temporal information was still missing in learned features[76]. Since training clips are sampled from videos uni-

formly, they have a problem with assigning false labels. It is presumed that the fundamental truth of each of these clips is the same as the basic truth of the video which can not take place if the action takes place only for a short period in whole video[76]. The method involved the pre-computation and separate storage of optical flow vectors. Special training was also offered for both streams, which means that end-to-end on-the-go training is a long road still[76]. Ji et al. develops a straightforward implementation of deep networking action recognition through 3D convolution networks[21]. In a time axis, 3D convolution kernels were applied to video frames to capture spatial and temporal information. They also claimed that their approach could capture information on movement and optical flow as frames are connected at the end by fully connected layers. Ng et al. proposed two CNN models to process each frame of the input video for action recognition[48]. Special 1x1 kernels process the output of intermediate layers of both architectures in fully connected layers. Finally, the method used 30 frames of unrolled LSTM cell connected to CNN's output in training. Bilen et al. analyzes the feature maps of the pre-trained model for video representation called dynamic image[49]. In the fine tuning phase, they added rank pooling operator and approximate rank pooling layer that combines maps of all frames into a dynamic image as one video representation. Deep learning based approaches have the ability to accurately identify hidden patterns in visual data due to their enormous pipeline of representation of features. On the other hand, for its processing, it requires enormous amount of data for training and high computational power.

Several models using LSTM, RNNs[2] have recently been developed to address sequence problems such as machine translation[29], speech recognition[20], caption generation[44], [47] and video action recognition[35], [43]. In 1997, the LSTM was introduced to combat the effect of the vanishing gradient problem that plagued the community of deep learning. The LSTM incorporates a memory unit that contains information on the LSTM unit's viewing inputs and is regulated using a number of fully connected gates. In Ng et al's work. authors investigated the idea of using LSTMs on individually trained feature maps to determine the possibility of temporary data captured in clips[48].]. Regrettably, they concluded that temporary pooling of compressed features is more efficient than stacking LSTM after trained feature maps[76]. In Donahue et al. paper, authors build on the same idea of using LSTM blocks (decoder) after convolution blocks but using end-to-end architecture-wide training[35]. They compared the RGB and the optical flow as the input choice and found that the weighted predictive score on both inputs was the preferred choice[76]. 16 frame clips will be sampled from video during training. The architecture is trained end-to-end with RGB input or 16 frame clips optical flow. For each clip, the final prediction is the average of predictions over each time step. The final video-level prediction is an average of each clip's predictions. The UCF 101 dataset benchmark test was found to be 82.92 percent. Even though the authors proposed end-to-end training frameworks, there remained some disadvantages, such as the false label assignment as video clips were breached and long-term temporal information was not captured[76]. Using optical flow meant separately pre-computing flow functions. During their work, Varol et al. tried to compensate for the stunted problem in the temporal range by using less spatial video resolution and longer clips (60 frames). This produced overwhelmingly improved results[8],

[76]. Recently, Xingjian et al. replaced the LSTM's fully connected gate layers with convolutional layers and used this improved model to predict nowcasting precipitation with improved performance from radar images[45]. This latest LSTM model is called the convolutional LSTM (convLSTM). It was subsequently used to predict optical flow images from videos[40] and to detect anomalies in videos[53]. The convLSTM model is capable of encoding spatiotemporal information in its memory cell by replacing the fully connected layers in the LSTM with convolution layers.

2.2 Violence Detection

In the specific area of the identification of violence, the bulk of the work is focused on the low-level characteristics. Usually, features such as optical flow, gradients, intensities and other local characteristics are extracted[69]. As a result of human body variations, it is difficult to capture effective and discriminatory features in the field of video-based violence detection. The main causes of the variations are scale, point of view, mutual occlusion and dynamic scenes. By recognizing violent components such as flame, blood[10], weapons and audio[64], most methods focused on detecting violent scenes. Such methods, however, may not be appropriate for monitoring videos with poor image quality. The disadvantages, such as low detection rate and high false alarm, limit this type of method. Furthermore, these features are not suitable for general surveillance systems that are always lacking in audio information. Nam et al. is one of the earlier works. They proposed threshold values for auditory and visual properties[3]. For auditory functionalities, amplitude and energy were taken into account as well as sudden changes in 2 of the total entropy [69]. They calculate the dynamic activity as visual features in order to identify rapid movements as well as pixel color thresholds for blood detection. Cheng et al. proposed an auditory approach to detect basic audio events such as gunshots, explosions, engines, car breaks, etc[5].Hidden Markov models (HMM) were trained in the recognition and targeting of sound activities and modeling correlations between multiple events with Gaussian mixtures to extract complex sémantic contexts[69]. However, these earlier methods relied on specific events and individually looked for each. Bermejo et al. are proposing a method that generalizes and attempts to identify violence through motion[15]. Using low-level features such as Space-Time Interest Points (STIP) and Motion SIFT (MoSIFT)[11], an extension of the SIFT[7] image descriptor for video, they exploited a Bag of Visual Word (BoVW) approach, adding an optical flow histogram representing local motion. For each video classified via Support Vector Machine (SVM), these features were then used to stabilize a bag of words. Souza et al.[30] also used a BoVW-based approach to classify video shots as violent or not with local spatio-temporal features. Multiple descriptions identified by STIP were hard coded to use spatio-temporal information and encode a bag of features for each shot and then a linear SVM was trained to classify videos so that relatively better results were obtained[69]. These approaches emphasize the importance of using movement and space-time features in the detection of violence. While the descriptors extracted around interest points may capture some meaningful information to detect violent actions, they may not be robust in the surveillance video scenario against scene variations. In addition, the enormous amounts of interest points extracted can hurt system efficiency directly.

Xu et al. used the MoSIFT algorithm to extract low-level video description. The non-parametric Kernel Density Estimation (KDE) and sparse coding were used to select the MoSIFT descriptors and process the selected features to eliminate redundant features and obtain more discriminative features[30]. Then, prior to classification, the typical BoW model was used. Senst et al.proposed the LaSIFT descriptor as a model for the classification of violent video appearance information and Lagrangian-based motion features[41]. The LaSIFT feature was evaluated with the BoW framework and showed better performance on the Hockey Fight dataset[15] and the Crowd Violence dataset[17] than the SIFT[7] and MoSIFT[11] descriptors. A novel approach was reported for violence detection in Reference[60], which could effectively describe dynamic characteristics in violent videos. The incorporation of the direction-based Lagrange field measurement into the SIFT descriptor developed a new characteristic for analyzing violence[74]. Then, an extended BoW procedure processed the features further. Similar to MoSIFT[11], a new descriptor named MoWLD for violence detection was developed by Zhang et al.[62]. MOWLD combines a WLD histogram describing space and an HOF that shows the movement of the points of interest [74]. Then, in a similar way to Reference[30] Zhang et al. processed the descriptors[62]. Although some appearance and motion information could be captured by the descriptors extracted around the interest points, they are limited to the locations of the interest points and omit the valid information beyond the neighborhood of interest points. There are other models for detecting violence. Zhang et al. proposed a quick and robust video surveillance video framework to detect and locate violent behaviour[56], [74]. First, it proposed a Gaussian Optical Flow Model (GMOF) to extract regions of candidate violence. Secondly, in the candidate regions, a novel descriptor called the Optical Flow Orientation Histogram (OHOF) was proposed. The descriptors were finally fed into a linear SVM to differentiate between violent and non-violent events. However, if the background is messy and dynamic, the GMOF algorithm will show low discriminative efficiency. Datta et al. detected violence by using information from a person's limbs about the trajectory of motion and orientation[4]. To obtain the position of the limbs, the precise silhouettes are required, but the segmentation of the object is difficult due to the serious occlusion. Some other works represent violent videos by combining statistical characteristics extracted from spatiotemporal motion blobs, including mean, variance, standard deviation, centroid, area, etc.[36], [37], [57]. The models with such features benefit from low computer complexity but have limited accuracy of classification. Deniz proposed a new method, which used the most important feature of violent conduct to use extreme acceleration patterns [23], [74]. By applying the Radon transform to the power spectrum of consecutive frames, these extreme acceleration features are effectively estimated. However, the dynamic background affects the extreme acceleration patterns, resulting in a high false alarm. Due to the serious occlusion and moving crowd, detection of violence in crowded scenes presents more challenges. Reference[17] has taken into consideration of the statistics of adjustments in the magnitude of speed vector in violent crowd behaviour. These statistics are represented using the Violent Flows (ViF) descriptor, which is collected for short frame sequences. The linear SVM classifies ViF descriptors. his method provided a computationally effective means of detecting crowd violence. However, the performance of the ViF-based method significantly decreased in the data set of non-crowd behavior. In order to identify non-crowded violence in videos[52] based on a ViF

descriptor a new feature known as Oriented Violent Flows (OViF) has been proposed[74]. The OViF features describe the movement magnitude changes based on the movement orientation statistics. However, in crowded scenarios, this approach could not work well. Huang et al. introduced a statistical method for detecting violent crowd behaviors based on optical flow fields. This method considered the optical flow field statistical characteristics and extracted the optical flow descriptor (SCOF) statistical characteristics to represent the video frames[26]. The descriptors of SCOF were then classified using linear SVM into either normal or violent events. However, this approach is limited to the descriptor of SCOF which only models the motion information and could not capture the features of the appearance. In this work, we aim to develop a method that in general scenes and crowded scenes can effectively detect violent behaviors.

Because of the great success of deep convolutional video action networks such as the Temporary Segment Networks[55] researchers created strong neural networks [46], [50], [51], [61], [63], [74]. Ding et. al extracted the spatial and temporal characteristics from the 3D convolution layers and classify the characteristics from the fully connected layers (as shown in Fig. 1)[24]. However, the variable length of videos is not supported by both of the proposed methods. And the 3D convolution computational time is growing rapidly due to the depth of the temporal axis. Dong et al. proposed the idea of using LSTM for the aggregation of features for violence detection[50]. This technique comprised of extracting features from raw pixels, optical flow images and acceleration flow maps through a convolutional neural network and late fusion[67]. Zhou et al. have established a FightNet to portray the complex relationship between visual violence and three kinds of input methods, i.e. RGB image coverage for space networks, optical flow and short term acceleration pictures for the network[63].Experimental results in the identification of violence have shown positive results [74]. In general, UCF101[19] is usedto pre-train deep neural networks for video-based violence detection to prevent over fitting. However, the networks on the targeting datasets do not always perform excellently, especially for datasets that differ significantly from the pre-trained dataset, like the Crowd Violence dataset[17], [74]. In this sense, a major obstacle impedes deep learning-based methods: lacking a sufficiently large training dataset for violence. Moreover, deep neural networks are inevitably suffering from higher computational complexity, which requires more advanced hardware devices. Regardless, motivated by the great success of deep neural networks in computer vision[12], [14], [31], [32], we are proposing an alternative deep learning approach to the detection of video-based violence.

Chapter 3

Algorithms, Datasets and Tools

3.1 Algorithms

3.1.1 CNN

Convolutional Neural Network (CNN) has been regarded as one of the most common categories in neural networks for image recognition, classification, object detection, face recognition etc. CNN is one of the most important features of neural networks that's rigorously used in the field of Computer Vision. CNN image classification usually takes an image as an input then process it and then classifies the image under different sort of categories such as Cat, Dog, Bear, Tiger etc. The input that the computer gets, it sees the image as an array of pixels and it entirely depends on the image resolution. It sees the height, width and the dimension of the input image. Basically, CNN is made up of different layers and the input images goes through these layers to be processed and categorized. These CNN layers usually consist of convolutional layers (Kernals), pooling layers, fully connected layers and normalization layers.

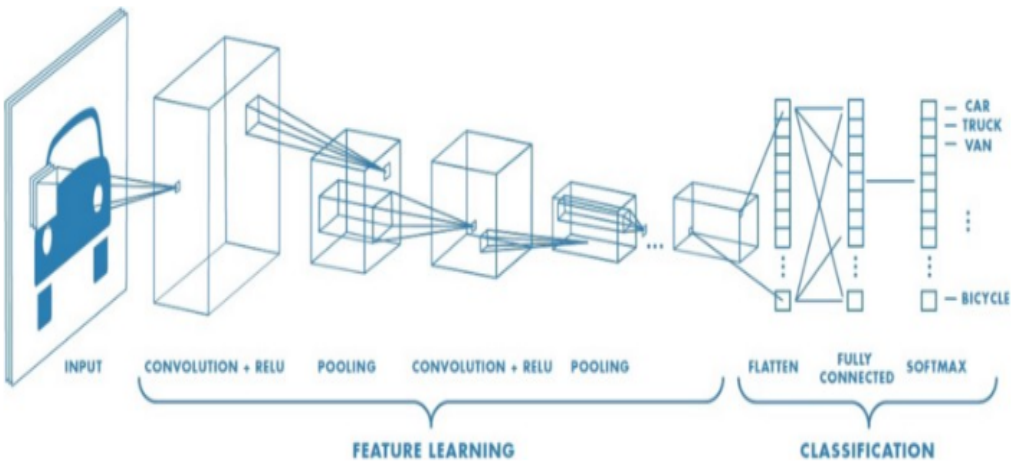

Figure 3.1: Convolutional Neural Network.

The convolution layer is the first layers the input image needs to go through and it is the first layer that extract different features from an input image. It learns about the image features by using small squares of data. The convolution layer usually take the input image and uses a filter on the input image which results in an output image so it takes two images as an input then produces a third as an input[75]. If we need to define this mathematically then the layer multiplies the signal from input with the kernel to get a modified signal.

For image processing its usually the few image matrices that becomes one matrix which is known as Kernel matrix. Below in the image it's a Feature Map which has taken a 5 x 5 image matrix multiplies with a 3 x 3 matrix.

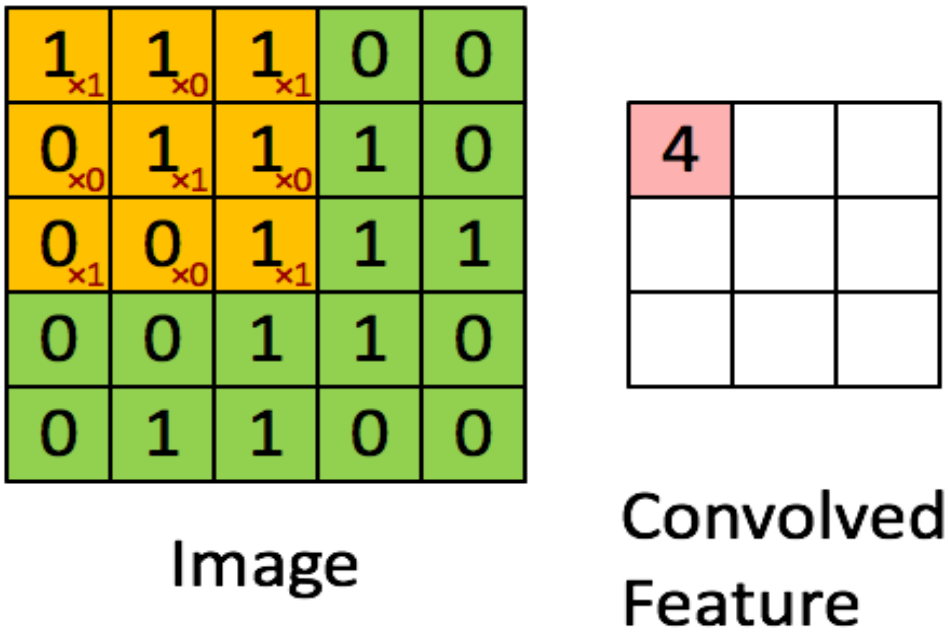

Figure 3.2: Feature Map.

In CNN the pixels also deal with strides. Stride is basically the number of pixels that shift over the input matrix. If the stride is 2 then the filters move to 2 pixels at a time. Also to take the input images perfectly and process them perfectly we need padding too. The filter needs to fit perfectly over the input image. We either cut out the image where the filter did not fit or pad the picture with zero where the filter did not fit perfectly.

Furthermore, we need pooling which is a sample based discretization process[68]. Pooling reduces the input images dimension and allows assumptions to be made about the features of the input image. There are different sort of pooling techniques that can be used to make assumptions such as Max pooling, Average pooling and Sum pooling. Max pooling is the most effective technique as it takes the largest element.

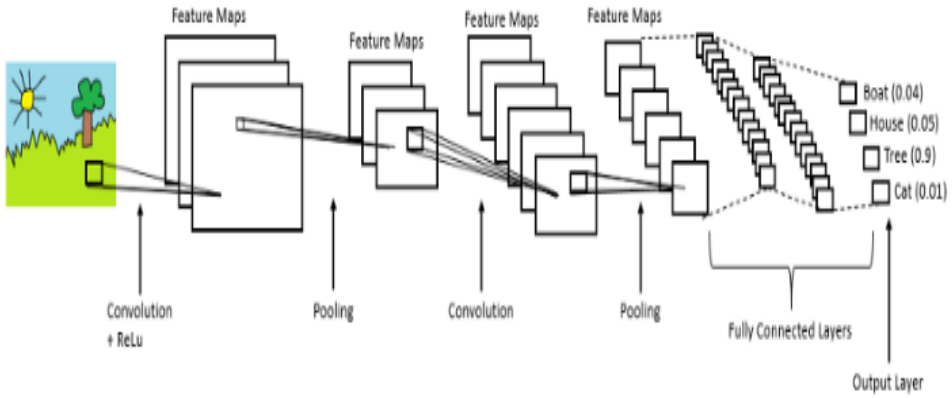

Figure 3.3: Simplified Representation of A CNN Architecture.

After the input image goes through the pooling layer the matrix turns into vector and then it goes into a fully connected layer like neural network. Lastly to summarize how we will be using CNN to make assumption from an image is we would feed an input image into the convolution layer. Then we would have to choose different parameters, apply filters with strides, use padding if the image required it then let the computer perform convolution on the image. The system then needs to perform pooling to cut down on the dimensionality size. After reducing the dimensionality we need to add as many as convolutional layer until its satisfied. Then feed this into the fully connected layer[70].

3.1.2 RNN

People do not start their thinking from scratch. We understand each word based on our understanding of previous words. We don't throw away everything and start thinking again from scratch. Our thoughts persist. This is not possible for traditional neural network and this is a major deficiency. This is where Recurrent Neural Networks comes in. They are networks with loops within them, which allow information to persist. They do not work like the traditional neural network which cant remember their previous input due to their nature but Recurrent Neural Networks is the first algorithm that recalls its input due to an internal memory which makes it perfectly suited to problems involving sequential data in video classification or predicting violent activities. Recurrent Neural Networks (RNN) are a powerful and robust type of neural networks and are currently among the most promising algorithms out there as they are the only ones with an internal memory[65]. Recurrent Neural Networks are able to remember important things about the input they received because of their internal memory, which allows them to be very accurate in predicting what will come next. This very function of RNN helps us in our research because we are predicting violent activities from videos so our system needs to remember which input it was given due to the fact that those videos are cut down into frames which goes as input one after another. If our system cant seem to remember the previous inputs then it wont be able to connect the dots and predict which is

a violent action and which is a non-violent action. To work with RNN's we need to understand what sequential data is. It is basically just ordered data, where the things or the inputs are related to each other and follows such as the sequence of the DNA or time series data.

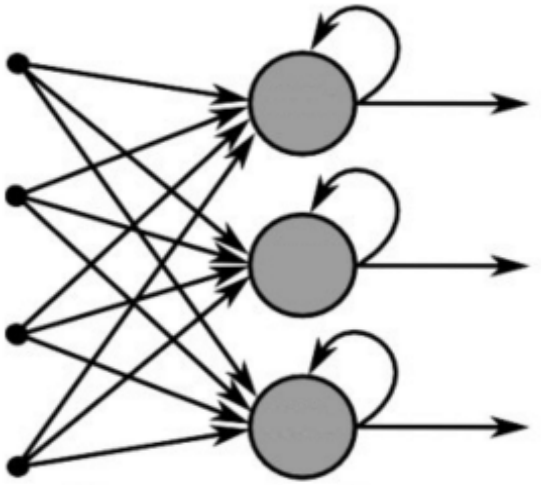

Figure 3.4: Recurrent Neural Network.

In RNN, the cycles of information through a loop. When making a decision, it takes the current input into account as well as what it has learned from the previously received inputs. RNN's usually has a short term memory. A good way to describe how RNN works in a simple manner is to imagine that we are feeding a word which is say " Working". Now if we input the word sequentially so that it can predict the word after some inputs it needs to remember the inputs from the start or else it won't be able to predict. Traditional neural network does not have any internal memory that's why for these sorts of problems they aren't much helpful, this is why we have chosen Recurrent Neural Network which will be able to remember the input beforehand and able to produce the output.

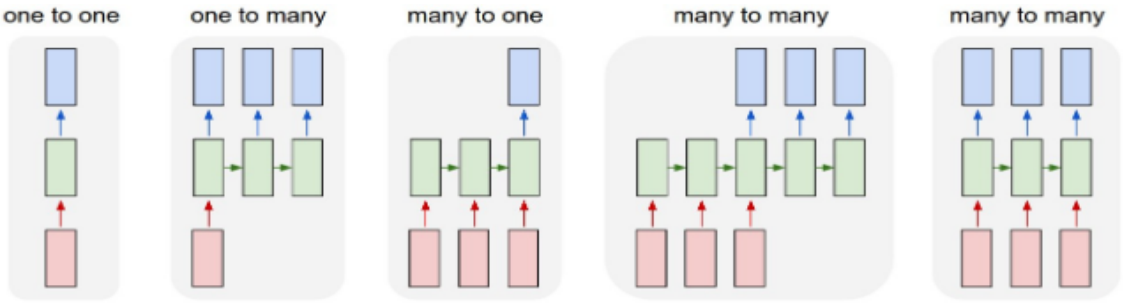

Figure 3.5: RNN's Map.

Each rectangle is a vector and functions are represented by arrows . Input vectors are red, output vectors are blue and green vectors hold the state of the RNN (more about that soon). (1) Vanilla processing mode without RNN, from fixed input to fixed output (e.g. image classification). (2) Sequence output (e.g. image capture

takes a picture and outputs a word sentence). (3) Sequence input (e.g. sentiment analysis if the sentence is classified as positive or negative). (4) Sequence input and sequence output (e.g. Machine Translation: the RNN reads the English sentence and then the French sentence).

3.1.3 LSTM

Long Short Term Memory networks (LSTM) are usually an extension of Recurrent Neural Network(RNN) and its capable of learning long-term dependencies. LSTM network extends RNN's memory to remember things even more that what the usual RNN can remember. For the layers of an RNN, which is then often called an LSTM network, an LSTM units are used as building units. LSTM allows RNN to keep track of their inputs for a long time. This is because LSTM's contain their information in a memory, which is similar to a computer's memory because the LSTM is able to read, write and delete information from its memory. This memory can be viewed as a gated cell, where gated means the cell decides whether or not to store or delete information (e.g. whether or not it opens the gates), based on the information's importance. Importance assignment occurs through weights that the algorithm also learns. This simply means that the information is important and not learned over time.

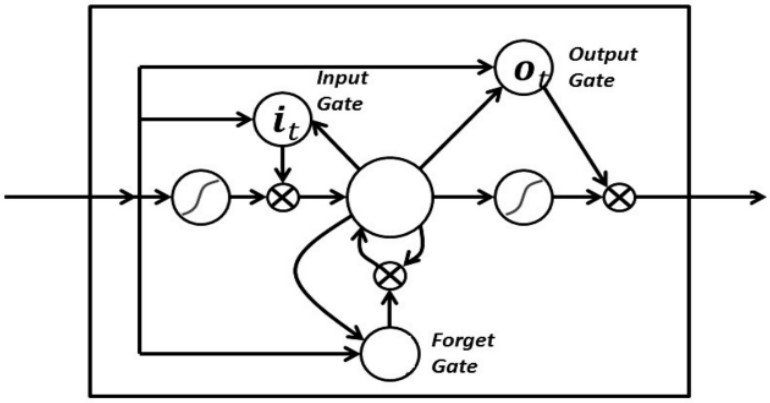

Figure 3.6: Three Gates in A LSTM.

We have three gates in an LSTM: the gate of input, forgetting and output. These gates determine whether to allow new input (input gate) or not, delete the information as it is not important (forget gate) or let it affect the output at the current time step (output gate). A picture of an RNN with its three gates can be seen above.

Equations for LSTM,

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{3.1}$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{3.2}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{3.3}$$

$\mathrm{i}_t \rightarrow represents\ input\ gate.$

$f_t \rightarrow represents\ forget\ gate.$

$o_t \rightarrow represents\ output\ gate.$

$\sigma \rightarrow represent\ sigmoid\ functions$

$w_x \rightarrow weight\ for\ the\ respective\ gate(x)\ neurons.$

$h_{t-1} \rightarrow output\ of\ the\ previous\ LSTM\ block\ (at\ timestamp\ t-1).$

$x_t \rightarrow\ input\ at\ current\ timestamp.$

$b_t \rightarrow biases\ for\ the\ respective\ gates(x).$

The first equation 3.1 is for the Input gate. This equation tells us the whether to allow a new input or not. The second equation 3.2 is for forget gate which tells us of the information is not important or not. And the third equation 3.3 is for output gate which tells us whether we let the unimportant information affect our output at the current time step.

3.2 Dataset

For our research we came across lots of datasets that are commonly available from different sources. We had to accurately choose between lots of options in order to use the datasets for our research. In most of the cases we see that the availability of datasets for particular research work are scarce but for our thesis work we had lots of available datasets that's been used before in similar research work. But in order to use them according to our research need we had to go through the descriptions and the compatibility with the system that we have created. Firstly, we came across violent-Flows database[17] which has both crowd violence and Non-violence database and benchmarks. The database includes all real world video footage of crowd violence along with standard benchmark protocols that are specifically designed to test both violent and non-violent classification and outbreak of violence. This dataset contains two hundred and forty seven videos. These videos have been downloaded from Youtube. The average length of all the videos footage is 3.60 seconds.

Another alternate dataset that we have used for our research project has been used before[17] in a similar research work as ours. According to the paper[15] action recognition research has been mostly given preference on working towards detecting simple actions such as walking, jogging but action detection for violent behavior has been comparatively less studied. This dataset was analyzed with the well-known Bag-of-Words framework to detect actions, along with two action descriptors STIP and MoSIFT. The database contains 1000 sequences that's divided into two groups fights and non-fights. The fight dataset is names Hockey Fight Dataset and the

non-fight dataset is called Movies Dataset.

The third and the most up to date dataset that we have used to train our system is called The VSD benchmark and it's a collection of violent events extracted from movies and videos, with high level audio and video concepts[34]. The creation of this dataset is intended towards using it for assessing the quality of methods for the detection of violent scenes or the recognition of violence related concepts in movies and web videos. The data consist of 3 different sub packages. First one is the data that's been used for the 2013 and before versions of the benchmark. This package uses old naming annotations. Second one is the data that's been used in 2014 benchmark. It follows new naming annotations, web videos and features and old naming of the annotations[78]. This dataset includes video footage from over 32 movies that describes the violence level from extremely violent movies to non-violent movies. Later in 2014 86 short videos were downloaded from YouTube and the frame rate was 25. The definition of the violent scenes according to the source are "scenes one would not let an 8 year old child see because they contain physical violence" [78]. In this dataset they have used different concept in the videos such as presence of blood, Fights, presence of fire, presence of firearms. We will primarily work with the Video concept – Fights. In this segment different types of fights were annotated which results in different tags in the file such as:

1. 1vs1: only two people fighting
2. Small: For a small group of people (number of people was not counted, it will roughly correspond to less than 10).
3. Large: for a large group of people (greater than 10).

3.3 Libraries

3.3.1 OPENCV

OpenCV(Open Source Computer Vision Library) is an open-source library which includes hundreds of computer vision algorithms[77]. OpenCV has a module structure that includes several shared or static libraries. As our thesis works primarily with video footage we had to utilize OpenCV and it has been an integral part of it. OpenCV has some modules that we can use such as Core functionality, it is a module that defines basic data structure, including dense multi-dimensional array Mat and basic function used by all the other modules available[77]. Its module includes linear and non-linear image filtering, geometric image transformations (resize, affine and perspective warping, generic table-based restoration), color space conversion, histograms, etc.[77]. We needed the image processing module because we primarily work with images from video footages and process them using OpenCV. Also one of the most important modules that we needed from OpenCV is the Video Analysis module which includes motion estimation, background subtraction and object tracking algorithm. OpenCV also includes modules such as Object detection of the predefined classes and an Video I/O module which is an easy to use interface to simple UI capabilities.

3.3.2 Tensorflow

Tensorflow is a distributed environment created by Google that has variety of use. TensorFlow is an open source library for large-scale machine learning and numerical computing. TensorFlow bundles a slew of machine learning models and algorithms together with deep learning (aka neural networking) and makes them useful through a common metaphor. It uses Python to provide the framework with a convenient front-end API to build applications while running high-performance applications [73]. Tensorflow is being used implement the complete CNN and RNN using deep learning methodologies. We have used Tensorflow to train the model. Then Validate and evaluate the models with sequence of images.

Chapter 4

Proposed Methodology

4.1 Overview

As mentioned previously, the aim of creating this system is to propose such a study through our prototype that is able to identify potential threats and provide alerts in likely cases like when fights break out. An applicable scenario for example, would be school fights or bullying. Figure 4.1 provides a comprehensive look at the overall working process of the proposed system. A CNN takes the input video frames and outputs the features to the Long Short-Term Memory (LSTM) to learn global temporal features and finally classify the features by fully-connected layers. This network can not only be implemented by the pre-trained models in ImageNet, but also have the flexibility to accept variable length videos.

The proposed architecture of the network is displayed in Figure 4.1 It has been shown that the local temporal characteristics that can be obtained from the optical flow are also important in addition to adding the LSTM (which is supposed to extract global temporal characteristics) after the CNN[72]. It has also been reported that the virtue of optical flow is due to its appearance invariance as well as its accuracy at boundaries and at small displacements[71]. Therefore, in this work, by taking two video frames as input, the effect of optical flow should be mimicked. The pre-trained CNN processes the two input frames. The first neural network is a convolutional neural network aimed at extracting high-level image features and reducing input complexity. We are using a pre-trained DarkNet model trained on the large visual recognition challenge ImageNet dataset. This is a conventional computer vision task, in which the models attempt to categorize whole images in 1,000 classes like "zebra", " cow" and "mop". Contemporary object recognition models have millions of parameters and therefore can take weeks to finish their training. Transfer learning is a technology that significantly improves many of this work with a fully trained model for a number of categories such as ImageNet and retraining for new classes from existing weights[58]. The two frame outputs of the pre-trained model's bottom layer are concatenated in the last channel and then fed into the additional CNN (labeled in Figure 4.1 by orange color). Since the outputs from the bottom layer are considered to be the low-level features, by comparing the two frames feature map, the additional CNN should learn both the local motion features and the appearance invariant features. The two frame outputs from the pre-trained network's top layer are also concatenated and fed into the other additional CNN to

compare the two frames ' high-level features. In order to learn the global temporal features, the outputs from the two additional CNN are then concatenated and passed to a fully connected layer and the LSTM cell. Lastly, the LSTM cell outputs are classified by a fully connected layer containing two neurons representing the two categories (violence and non-violence), respectively.

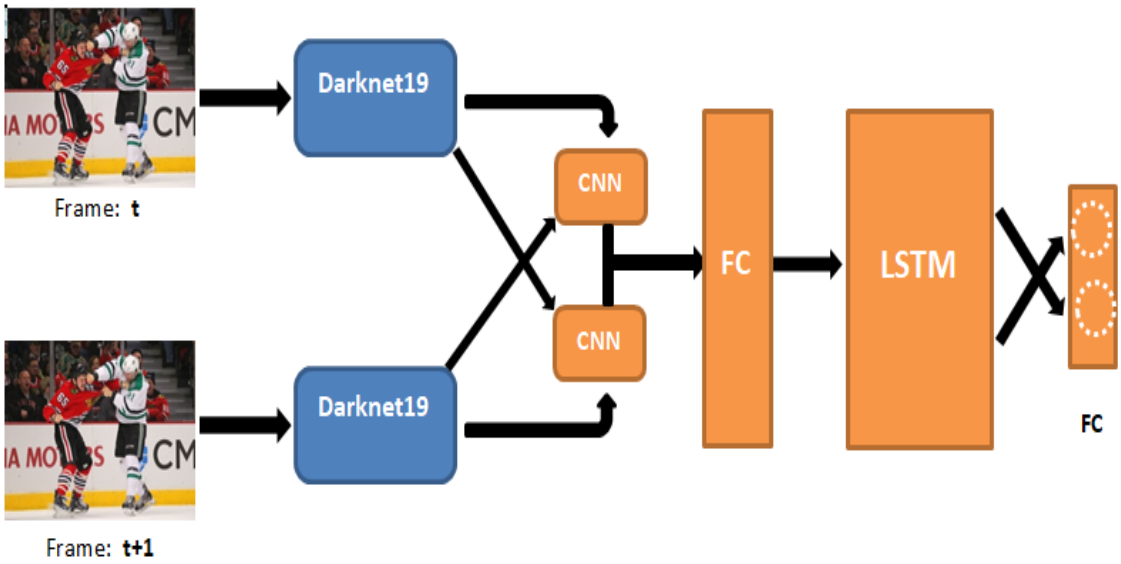

Figure 4.1: The Proposed Network Architecture.

The blue-colored layers are pre-trained on the ImageNet dataset and frozen during training. On the video dataset, the layers marked by the orange color are trained. Due to its accuracy on ImageNet and the above-mentioned real-time performance, Darknet19[59] implements the pre-trained model. Since the Darknet19 already contains 19 convolution layers, the additional CNN is implemented by the residual layers[27] to avoid the degradation problem.

Figure 4.2 shows a prototype for system application. The end-user process is straightforward: the goal is to have a front-end where you can upload and start classifying a video in real-time. You can see how the classes and the respective accuracy for that class are constantly changing. These values constantly update every three seconds until the video is over One of the things we can do with this video classifier is to connect it to a security camera and keep analyzing the video in real time, and when the system detects criminal or suspicious activity, it could activate an alarm or alert the police. In addition, we can use a similar system trained with the appropriate data to detect different types of activity, such as a camera located in a school where our goal would be to detect and stop bullying. This is further mentioned in our future plans.

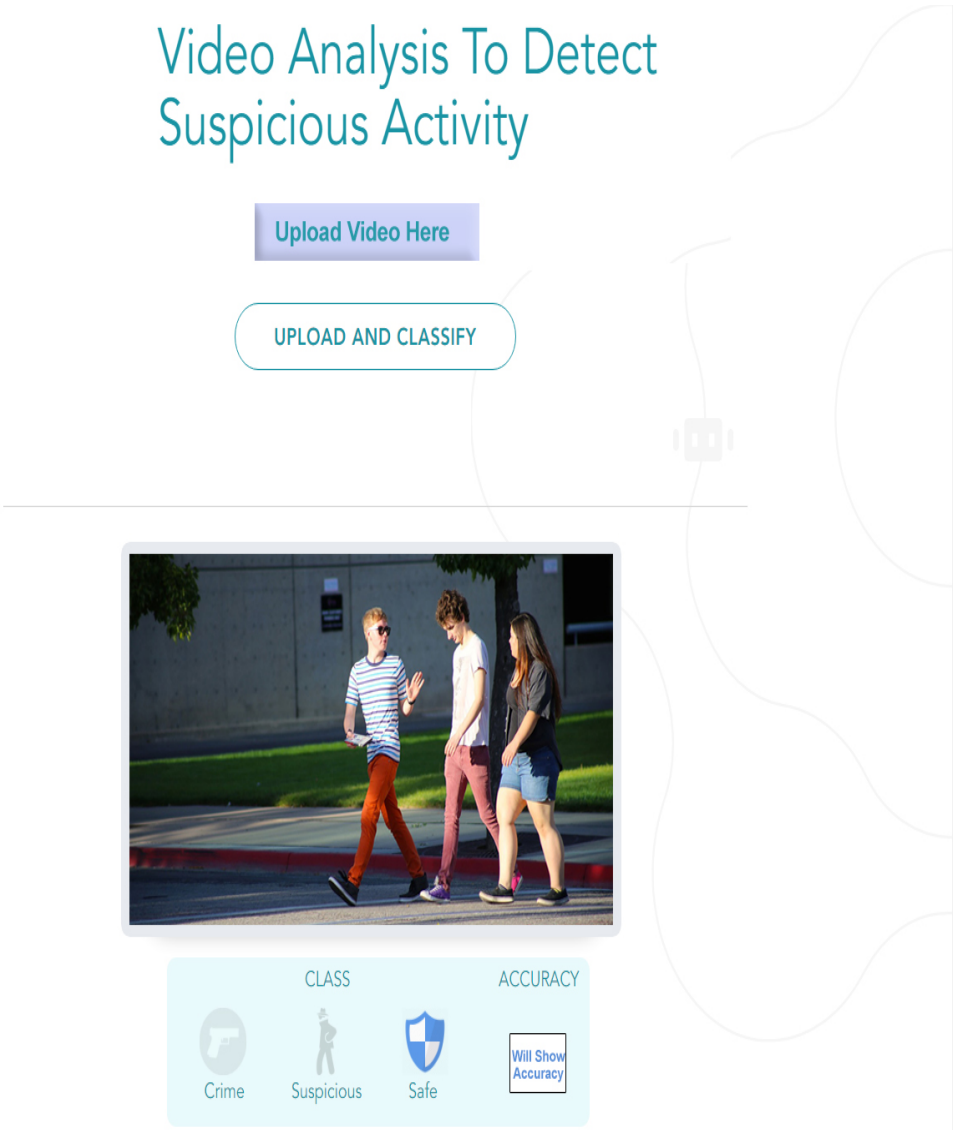

Figure 4.2: Possible Front-end of Our System.

4.2 Violence Representation

The best deep-learning solutions work with very clear and concrete concepts, such as well-defined objects, facial expressions and specific actions. On the other hand, the concept of violence is subjective and complex, posing the challenge of how to reliably represent it in a neural network. Consequently, our suggestion is to separate the concept of violence into sharper and more practical definitions[69]. With these, a broader concept such as violence can be detected by aggregating specific features.

4.2.1 Defining concepts of violence

It is not entirely new to divide violence into more specific concepts. Back in 2003, Cheng et al identified audio signatures that could signal various types of violence, such as explosions, gunshots and car crashes[5]. In line with this, to find more specific types of violence, we can train different detectors as suggested by [69]. Instead

of fusing several general features that attempt to encapsulate the entire concept of violence, we can break down into smaller features and find features for each, later combining them for a more robust system. This is not without its challenges, since data is scarce for specific events. Here we are examining the Hockey Dataset to see whether a fight scene is occurring or not. A considerable amount of the labelled data on which the study is based on treats violence as a general idea and it is very subjective, as in one can be identified as a violent person in a specific scene, but not another [69]. We can find a better definition of violence itself by using specific concepts of violence as a starting point, recognizing common characteristics between them. The first step towards a more robust representation of this problem is to grasp the nuances of the concept of violence and understand its definitions. We used fights as the defined concepts of violence from the available dataset as it was more evident in it. We then trained individual CNNs to learn their specific characteristics for each concept.

We then trained individual CNNs to learn their specific characteristics for each concept. Our architecture maps outcomes of the last layer into two filters (violence versus non-violence). For each concept, the network is also finetuned. The features were extracted from the network's last fully connected (FC) layer, which is the last pre-classification layer. This output's dimensionality is 1024-d. Each of these particular concepts has its own characteristics, so the neural network must learn independently. For example, still images can detect and confidently classify the presence of firearms, cold arms and blood. Firearms have a well-defined range of shape and color, whereas the color and texture of blood can be confidently characterized. We plan to include these classifications later. Each one of these concepts has its own properties and needs to be learned separately. However, fights and detonations are timely events and can use a network to extract features related to those concepts using this information[69]. On the other hand, gunshots can be identified through audio and a network that can analyze this video signal can give us a better set of classification features. We train a specialized detector for each concept and send its characteristics to independent classifiers and only then do we fuse the scene to be classified as violent or not.

4.3 Network Architecture

4.3.1 Input Frames and CNN

Videos are picture sequences. In order for a system to identify if a fight is taking place between the people present in the video, it should be able to identify the locations of the people and understand how the movement of the said people is changing over time. Convolutional Neural Networks (CNN) are capable of giving each video frame a good representation. A Recurrent Neural Network (RNN) is needed to encode the temporal changes. Since we are interested in spatial and temporal dimensional changes, LSTM will be an appropriate option. The LSTM will be able to encode the spatial and temporal changes compared to LSTM using the convolutionary gates in them. This will lead to a better representation of the video being analyzed.

4.3.2 Transfer Learning using DarkNet-19

This is a conventional computer vision task, in which the models attempt to categorize whole images in 1,000 classes like "zebra", " cow" and "mop". Contemporary object recognition modelshave millions of parameters and therefore can take weeks to finish their training.Transfer learning is a technology that significantly improves many of this work witha fully trained model for a number of categories such as ImageNet and retrainingfor new classes from existing weights[58]. There are two pre - trained models, one for images of 224x224 and one for images of 448x448. We chose the 224x224 model.Taking the 224x224 sizes would mean less overall parameters and thus less computational power. Furthermore, taking larger sample size would require more data to train on resulting in insufficient samples that could lead to underfitting. This issue is also alleviated by taking the smaller patches. The input image channels are in RGB order (not BGR), with values within [0, 1] normalized.

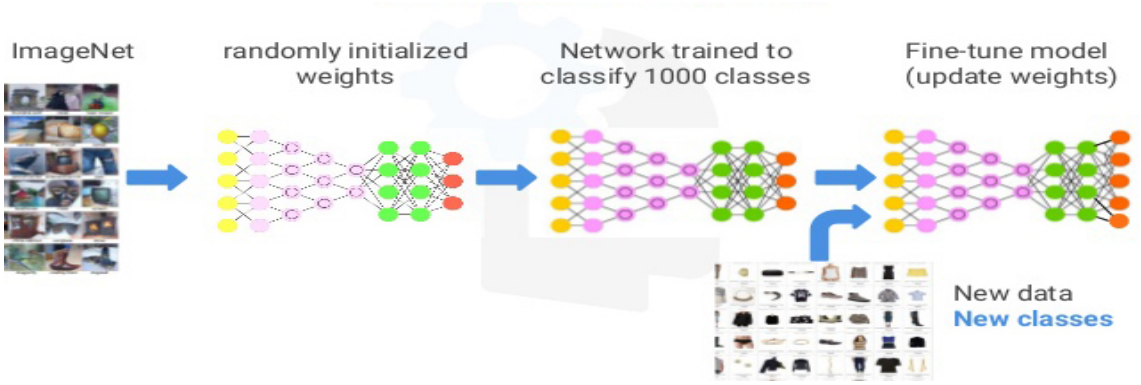

Figure 4.3: Transfer Learning.

Darknet-19 consists of mostly 3x3 filters and it uses global average pooling to make predictions as well as 1x1 filters to compress the feature representation between 3x3 convolutions. It has 19 convolutional layers and 5 max-pooling layers[59]. It has been primarily used due to its faster performance and less computational power compared to its successor Darknet 53.

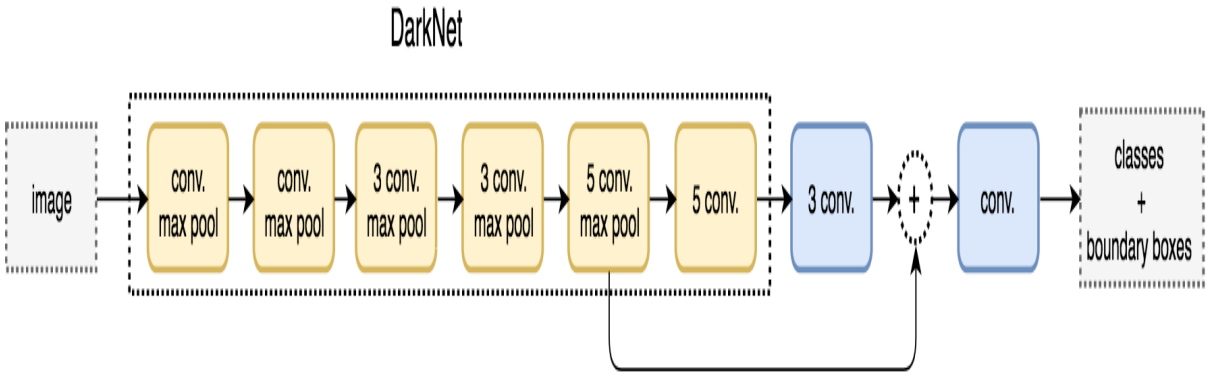

Figure 4.4: DarkNet 19.

4.3.3 Incorporating Temporal Information

We will take advantage of the temporal information available in videos for the second front. The setting in which they happen is likewise significant for violent scenes, and this temporal data is crucial to recognize it from non-violence. Various motion descriptors and other forms will be explored to represent time. Most recent approaches use a combination of features such as MFCC[33], [39] or time-related visual features such as STIP[33], Dense Trajectories[33], [39], MoSIFT[38], and Temporal Robust Features a.k.a. TRoF[54] to describe motion. However, those extracted by a CNN[33], [38], [39], [42] are the most prominent features. It is an incredible favorable position to have the capacity to process tremendous measures of data and consequently find a set of features that can be utilized to arrange this data. Although early outcomes are a long way from perfect, the improvement and potential is critical, particularly when temporal data is sustained to the neural network.

4.3.4 CNN + LSTM

To recognize a video as violent or non-violent, the network ought to have the capacity to encode localized spatial attributes and how they change over time. With the downside of having increased computational complexity, handcrafted features are able to achieve this. CNNs are capable of generating discriminating spatial features, but for temporal encoding using Long Short Term Memory (LSTM), existing methods use the features extracted from the fully connected layers. The fully-connected layers output represents the entire image's global descriptor. Therefore, the existing methods do not encode the spatial changes located. As a result, they use methods that involve adding more data streams such as optical flow images resulting in increased computational complexity. In this context, the use of LSTM becomes relevant as it encodes the convolutionary features of the CNN. In addition, the convolutional gates in the LSTM are trained to encode local regions ' temporal changes. This enables the entire network to encode localized spatiotemporal characteristics. Convolutional neural networks process information as separate data points, but other types of networks are also promising, such as Recurrent Neural Networks (RNN) or the (LSTM) networks, which both store and use information from previous frames to calculate features for the next frames, incorporating some temporal information within their own architecture. RNNs ' main attraction is that they not only take the current input they see as input, but also what they perceived a step back in time and combine it to determine how they respond to new data. This means the network can take advantage of the sequential information, and this has a clear purpose to use the sequence information itself to perform tasks that are impossible for feed forward networks. Sequential information is preserved in the hidden state of the recurring network, which spans many steps as it falls forward, affecting each new input processing. This architecture can be very useful as it is a sequence of events that happen over time.

An LSTM network works similarly, but instead of just using the information passed by one step in the past, they learn through many steps of time, allowing them to link causes and effects that occur over an extended period of time. They contain information in a gated cell outside the recurrent network's normal flow. This cell makes decisions on what to store, and when to allow reading, writing, and erasure,

through opening and closing gates. These types of networks open up opportunities for the detection of violence, as we can effectively use the video's temporal information as input. These types of neural networks point to a hidden transitional state between a non-violent and a violent state, and identifying this hidden state can be valuable information, not only to help identify when the violent scene begins and ends, but also to be able to predict if there will be a violent scene.

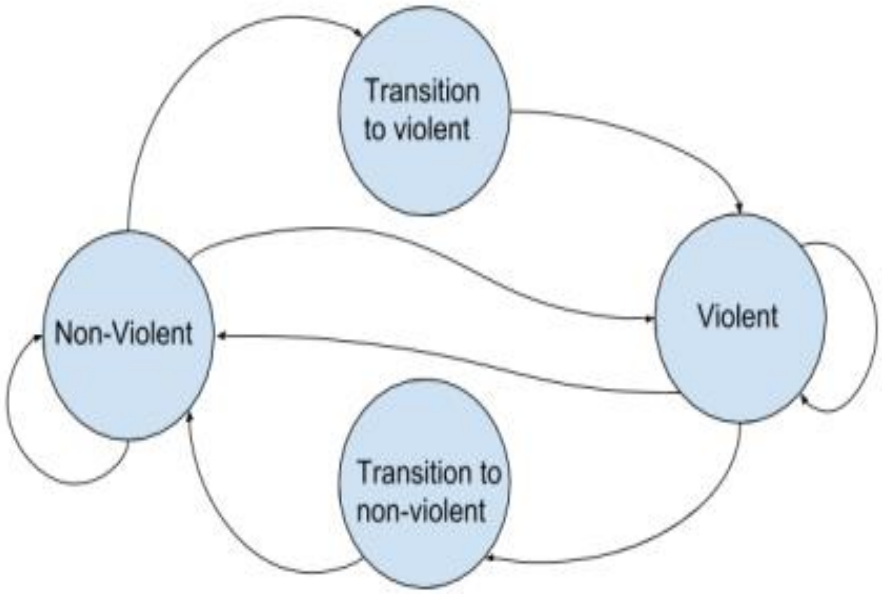

Figure 4.5: Overview of The Markov Violence Decision-making Process And It's Transitional States.

Designing a Markovian decision process from a neural network has been used for problem optimization[50], but it can be adapted to identify the boundaries of a violent scene and transfer context information to the classification process. A violent scene is full of subjective contexts, and depending on the nature of the video, a fighting scene, for example, can be intertwined with spectators ' close-ups, which alone do not indicate violence, but are placed in a violent context. Identifying these transitional states can help to determine whether or not a non-violent scene belongs to a violent scene, as illustrated by Figure 4.5.

4.3.5 Leaky Rectified Linear Unit

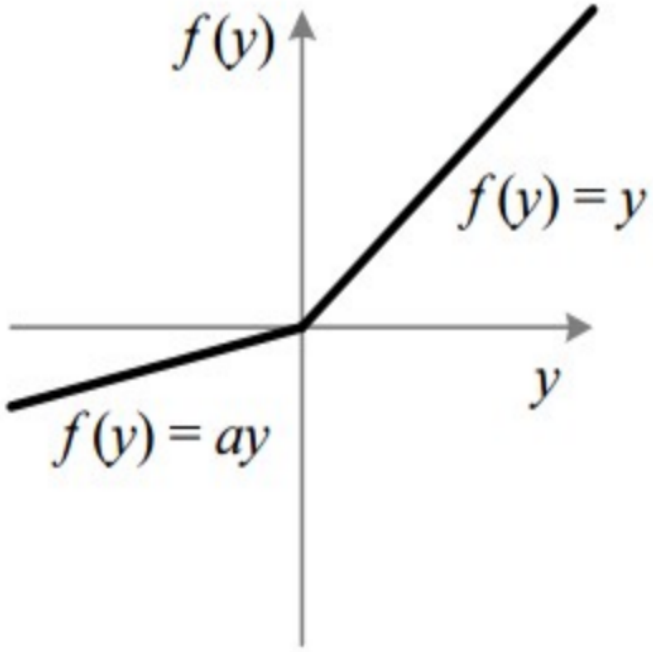

Figure 4.6: Leaky ReLU.

The Rectified Linear Unit is the most commonly used activation function in deep learning models with a Leaky ReLU (LReLU), we are overcoming the "dead ReLU" problem that happens when your ReLU always has values below 0-because of 0 gradients in the negative part, this completely blocks learning in the ReLU. The LReLU derivative is 1 in the positive part, and the negative part thus is a small fraction as mentioned in Figure 4.6.

Chapter 5

Implementation

5.1 Experiments and Optimization

In order to extract hierarchical characteristics from the video frames, the convolution layers are trained and then aggregated using the LSTM layer. The network functions as follows: sequentially applying the frames of the video under consideration to the model. Three benchmark datasets are used and the classification accuracy is reported to evaluate the effectiveness of the proposed approach in classifying violent videos which is further mentioned later in the paper.

5.1.1 Experimental Setup

Using the TensorFlow library, the network is implemented. For training, N number of frames that are equally spaced in time are extracted from each video and re-sized to a size of 224 to 224. Every 0.2 seconds, we extracted a frame and used this frame to make a prediction using the Darket19 model. This is to stay away from the excess calculations involved in the processing of all frames, as adjacent frames contain data that overlaps. The number of frames selected is based on each dataset's average video duration. The network is trained with 10 learning rates and 40 batch sizes using the RMSprop algorithm which is the number of frames in one iteration. Using the Xavier algorithm and variance scaling, the model weights are initialized. During the training stage, the network is running for 40 iterations. The video frames are re-sized to 224 / 224 at the evaluation stage and are applied to the network to be classified as violent or non-violent. To make their mean zero and variance unity, all the training video frames in a dataset are normalized.

Once all the frames are applied, in this final time step, the hidden state of the LSTM layer contains the representation of the applied input video frames. This video representation is then applied to a series of fully-connected classification layers in the hidden state of the LSTM. We used 128 filters in 2 gates in the LSTM and 64 filters in the second gate, conv2. This is with a filter size of 3/3 and step 1. Thus the LSTM's hidden state is made up of 128 feature maps. Before the first fully connected layer, a batch normalization layer is added. Before each of the convolutionary and fully connected layers, a leaky rectified linear unit (ReLU) non-linear activation is applied. In the network, the difference between adjacent frames is given as input instead of applying the input frames as such. Thusly, rather than

the frames themselves, the system is compelled to demonstrate the progressions occurring in adjacent frames. This is inspired by the technique suggested in[28] by Simonyan and Zisserman to use optical flow images as input for action recognition to a neural network. The distinction image can be considered as an unrefined and approximate adaptation of optical flow images. As a input, a distinction between adjacent video frames is applied frames in the proposed method. This avoids the computational complexity involved in the generation of optical flow images. The network is trained to minimize the loss of entropy in the binary cross.

5.1.2 Accuracy Evaluation

In this work, the proposed model can deliver the classified result per frame. The previous research, however, evaluates the video-level accuracy. The results of the frame-level are collected and processed by the following strategy in order to be able to compare with the previous work: the video is classified into a certain category if and only if the number of such category's continuous signals is greater than a certain threshold. This threshold can be derived by scanning the threshold from 0 to the video length and seeing which threshold provides the best accuracy in the validation set, as shown in Figure 4.3. If multiple thresholds can yield the same accuracy, choose the small one. Photograph. Figure 5.1. The validation set's threshold-accuracy curve. The horizontal axis is the threshold of the number of continuous frames with the positive signal. The vertical axis in the validation set represents the accuracy at such a threshold. Thresholds starting from 3 to 9 are all the most accurate in the Figure . The smallest threshold (i.e. threshold= 3) is selected so that this metric could reflect the continuous false positive in the test set.

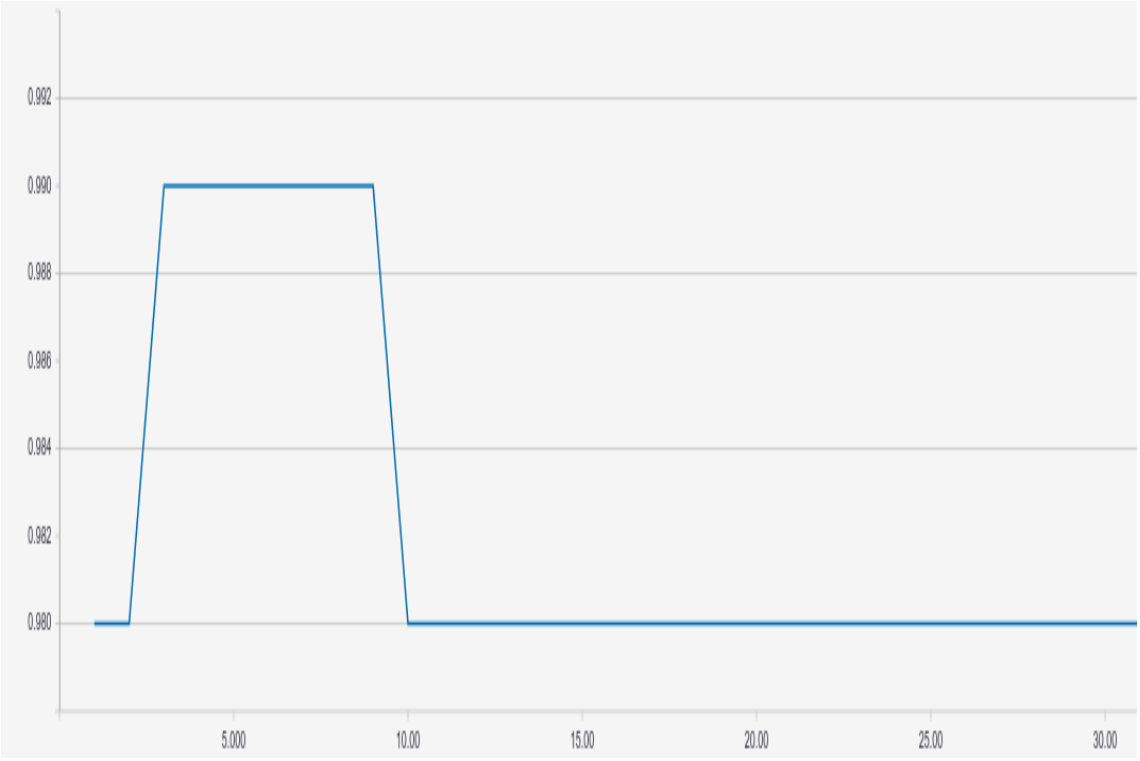

Figure 5.1: The threshold-accuracy curve in the validation set.

5.1.3 Gradient Clipping

Gradient Clipping Because of the long-term components[22], it is well known that the recurrent network gradient may increase rapidly. The normal way to handle the explosive gradient would be: truncate the gradient to keep it within a reasonable range. While several studies solve this problem through a different approach: start training with a few unrolls, then double the size of unrolls when the loss reaches plateaus[66]. They found it wasn't even necessary to clip the gradients in the second approach. They also state that the network may not even converge without starting from the small unrolls[66]. In this work, I found that even the initial unrolls are set to the length of the videos, the network can easily converge. The absence of gradient clipping, however, causes the loss curve to oscillate during training, even if the training starts with a small unroll. Therefore, the network's gradients are truncated between -5.0 and 5.0. The gradients were also tested in the smaller range (e.g. from -1.0 to 1.0). My experiment, however, shows that this will hardly cause the network to converge to the lower minimum.

Chapter 6

Results

To reiterate, the study proposes a methodology to detect violent and non-violent scene. In this paper, we used three types of algorithms: CNN+LSTM and RNN+LSTM. At the end of the research, the most accurate, convenient and efficient options were used from the variety of options available in each part. Bermejo et. al proposed the dataset for hockey. It has 500 fighting clips and 500 hockey game non-fighting clips[13]. Following Ding et Al. proposed experiment, our data set is further divided into 200 clips (including 100 fighting clips and 100 non-fighting clips) for testing, 800 training clips[25]. Table 6.1 shows the result. So the respective percentage is 80 % for training and 20 % for testing.

Table 6.1: Comparison Between Two Proposed Models.

Model	Accuracy
CNN	92%
CNN + LSTM	98%

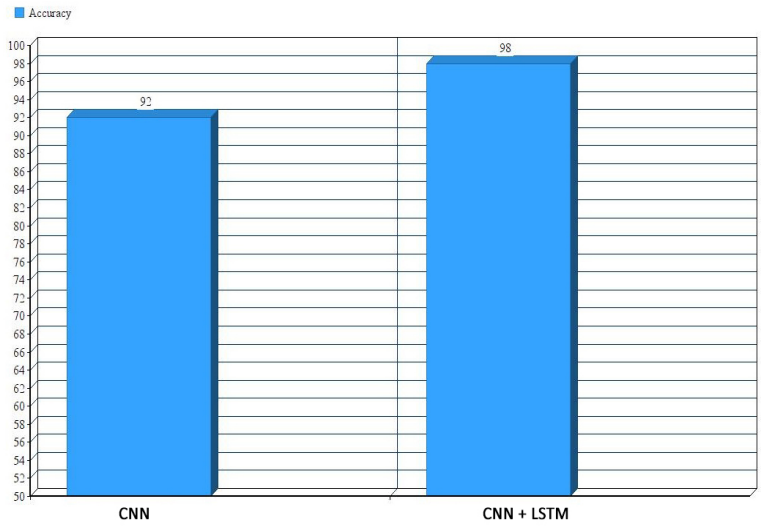

Figure 6.1: Comparison Between Two Proposed Models.

In table 6.1 and graph 6.1, we can see that CNN model gives less accuracy than CNN + LSTM. The main reason for giving better accuracy is RNN works on the

principle of saving the output of a layer and feeding this back to the input in order to predict the output of the layer. While CNN can do only image recognition and object classification. Furthermore, CNN only considers the current input while RNN considers the current input as well as the previously received inputs. Because of its internal memory, it can memorize previous inputs. RNN also handles sequential data and has a short term memory. However, as we are using LSTM, it has a long short term memory. Because of LSTM the training takes less time and also has high accuracy. Furthermore it solves the problem of gradients disappearing. As a result, RNN LSTM gives more accuracy than the other algorithms we used.

We followed the 5-folds cross validation technique for performance evaluation of our model. The model architecture selection was made by evaluating the performance of the various model dataset.

Table 6.2: Comparison Between Previous Methods and Proposed Method.

Methods	Violent-Flows Dataset	Hockey Dataset	VSD Bechmark
1. MoSIFT+HIK[15]		90.9%	89.5%
2. ViF[17]	81.3±0.21%	82.9±0.14%	
3. MoSIFT+KDE+ Sparse Coding[30]	89.05±3.26%	94.3±1.68%	
4. Three streams + LSTM[50]		93.9%	
5. Proposed	92.19±0.12%	98±0.55%	94.57±2.34%

Table 6.2 and graph 6.2 shows that the comparison between different method that has been applied before and the proposed method. One can see that the proposed method exceeds other state-of - the-art methods in this work. Moreover, it shows that the proposed method gives a better result in the case of Hockey Fight Dataset. The biggest problem to consider aggressive behavior as violent is in the case of sports. For example, the fight video includes players fighting each other in the hockey dataset. So checking if one player moves closer to another player is an easy way to detect violent scenes. However, non-violent video is also made up of hugging players. So, it becomes very tough to detect violent scene and non-violent scene. But this can be avoided by the proposed method which encodes motion of localized regions. In the case of Violent Flows Dataset, the proposed methodology does not give a good accuracy like Hockey Fight Dataset. The main reason is only a small part of the aggressive behavior is found in the violent videos while a large portion of the videos remain as spectators. As the model founds most of the people behaves normally, it marks it as a non-violent scene. Models that do not consider temporal information i.e. single frame models have been reported to have a strong performance. This could be due to the fact that the network does not necessarily have to learn the motion features of the moving objects and that several categories in the video classification task in multiple datasets like Sports-1M and UCF-101 can be easily recognized. The scene or the backgrounds in the videos help in that regard.

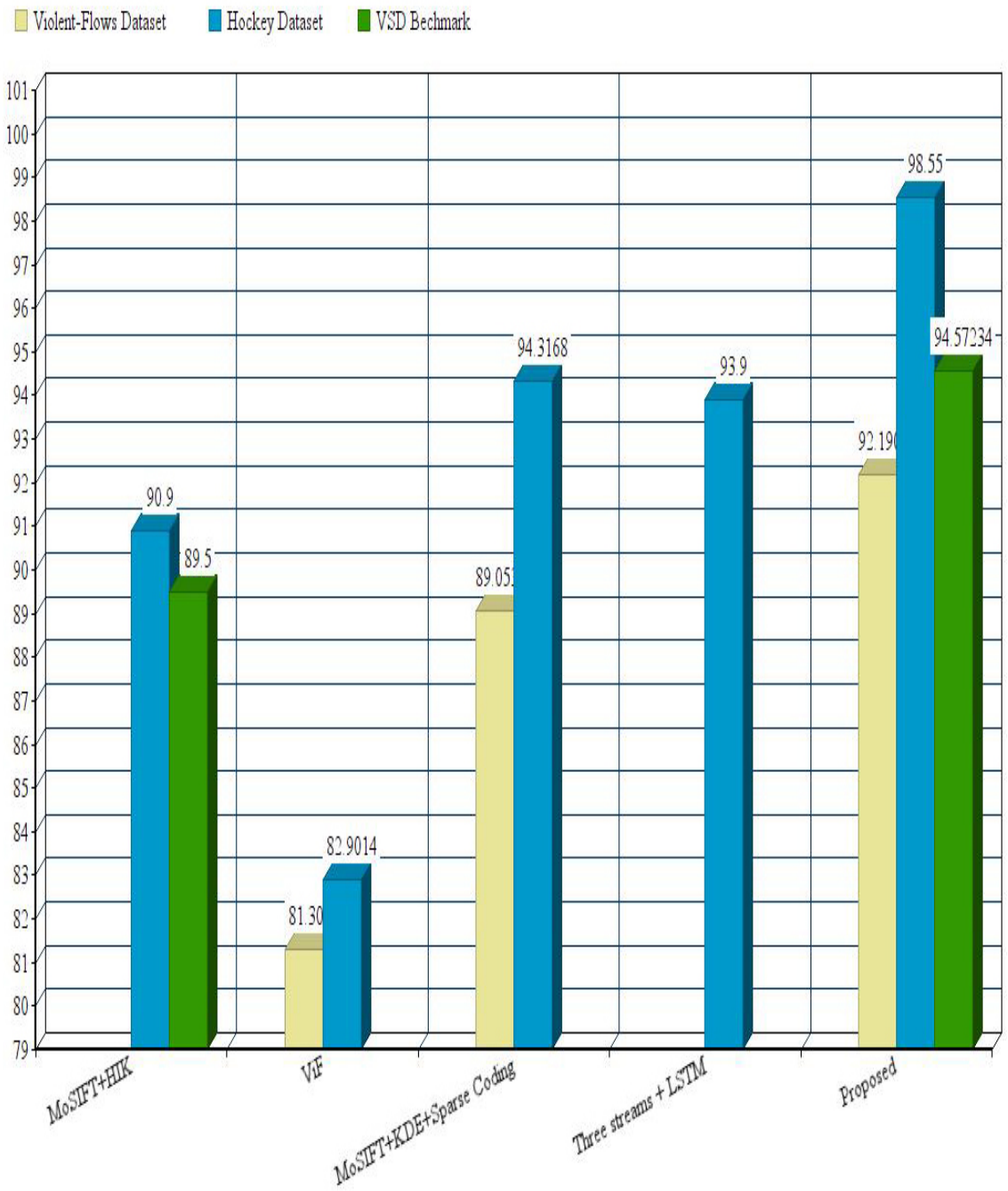

Figure 6.2: Comparison Between Previous Methods and Proposed Method.

However, when considering the Sports -1M dataset all the videos are shot in the hockey field in this work, and when examined by the human eyes, several frames are needed for better recognition. A simple single frame network was also proposed to compare the proposed method with the single frame model. As illustrated in Figure: 6.4, The output of the Darknet-19 goes to the single frame. After that, for input classification the output of the feature map goes to the three fully connected layers.

Table 6.3: Classification Accuracy Obtained with The Dataset for Different Models.

Input	Classification Accuracy
Video Frames (randominitialization)	94.1±2.9%
Video Frames (Darknet19 pre-trained)	96±0.35%
Difference of Video Frames (randominitialization)	95.5±0.5%
Difference of Video Frames (Darknet19 pre-trained)	98.5±0.55%

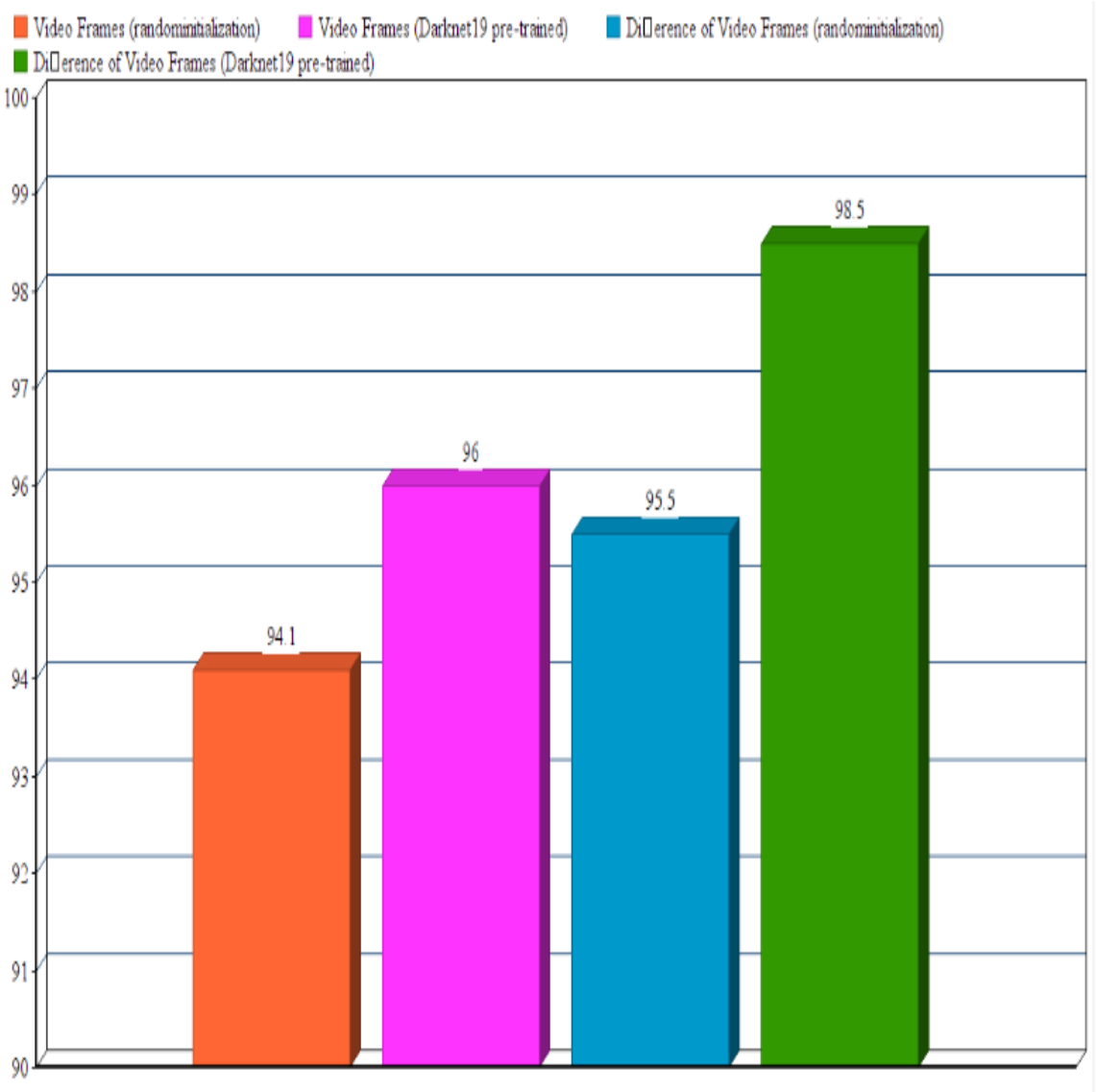

Figure 6.3: Classification Accuracy Obtained with The Dataset for Different Models.

Moreover, table 6.3 and graph 6.3 shows the accuracy of using two different inputs which are video frames and difference of video frames. The table also shows that Darknet19 pre-trained network model results which is almost 98.5% compared to

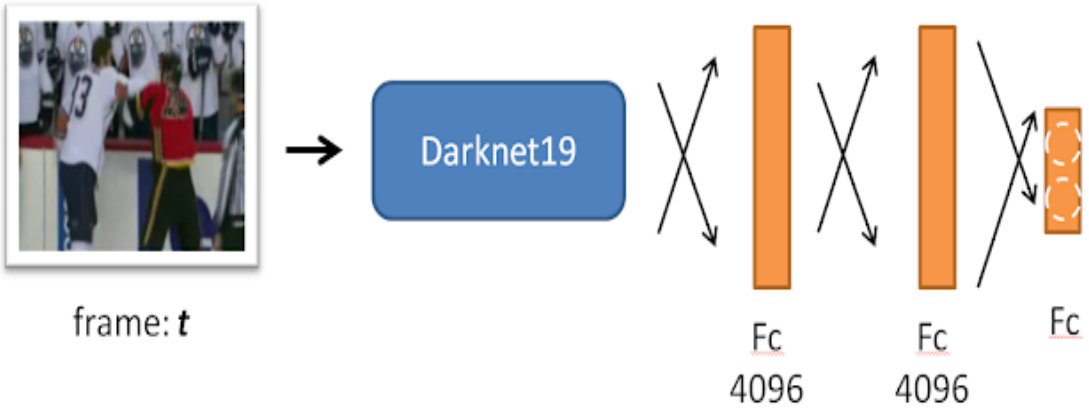

Figure 6.4: Single Frame Model.

using a randomly initialized network. We choose difference of video frames as input and use a pre-trained model which is Darket19. Darknet19 database are able to generalize better and lead to better performance for tasks such as action recognition. As we are using difference of video frames as input, the network is forced to model the adjacent frame changes rather than the frames themselves. As a result, differece of video framse which is tained on Darknet19 gives better performance than others.

Chapter 7

Future Work and Conclusion

7.1 Future Work

In future, we plan to design an online front-end application where we can upload videos to detect violent activities. Furthermore, we are planning to take our research into next step by detecting suspicious activity in real time. We will try to connect this prototype with surveillance camera and a device so that it can detect suspicious activity or criminal activity. The moment the system detects suspicious or criminal activity it could active an alarm or alert the police or guards. In our research we are only using three kinds of datasets: Violent-Flows Datasets, Hockey Datasets and VSD Bechmark. However, we can also train our network with other different datasets to detect different kinds of activity. For instance, we can train our network to detect bullying in real time with a camera located in school and college.

One of the challenges raised by violence's subjectivity is that one person can classify the same scene as violent and not violent by another. If violence is treated as a relative attribute[6], this ambiguity is lessened. We plan to show a couple of scenes instead of asking if a scene is violent and ask which one is more violent. Kovashka et. al have shown that when making relative statements vs. absolute statements, humans agree more[18]. With this in mind, we can label our data as a relative attribute with violence and use statistical models like the Bradley-Terry[1] to estimate and assign real-number scores for each perceived violence scene. We now have a perceived violence global ranking and can use it to define better thresholds for what is considered a violent scene. This method can also be used in conjunction with concepts of violence to help determine which are perceived to be more violent and therefore more important for the final classification. For instance, if scenes ranked higher in violence contain fights more frequently than fire, in a later classification fusion we can adjust the weight of these concepts. It is also a step towards a more robust study of the definition of violence.

In terms of more kinds of violent activity being detected. These particular concepts have their own characteristics, so the neural network must learn independently provided there is a good dataset available. In the future we plan to categorize violence further by adding more sections like suspicious, fire, explosions, gunshots, etc.

7.2 Conclusion

In this paper, we proposed a network system which can detect violent and non-violent scene. This network uses pre-trained model on ImageNet (Darknet19) dataset and also extracts global and local temporal features. CNN is used for frame level feature extraction. Convolutional long short term memory is being used for feature aggregation in the temporal domain. The proposed model is performed on three different datasets which are violent flows dataset, hockey fight dataset and VSD benchmark dataset. We created a system with a high accuracy in detecting violent activities from pre -recorded videos. To detect violence in real time processing speed or any sort of violent activities frame by frame we need higher processing speed. We also showed that if the network is trained on video frame difference as input, it gives better accuracy.

Bibliography

[1] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons", *Biometrika*, vol. 39, no. 3/4, p. 324, 1952. DOI: 10.2307/2334029.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] J. Nam, M. Alghoniemy, and A. H. Tewfik, "Audio-visual content-based violent scene characterization", in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, IEEE, vol. 1, 1998, pp. 353–357.

[4] A. Datta, M. Shah, and N. D. V. Lobo, "Person-on-person violence detection in video data", in *Object recognition supported by user interaction for service robots*, IEEE, vol. 1, 2002, pp. 433–438.

[5] W.-H. Cheng, W.-T. Chu, and J.-L. Wu, "Semantic context detection based on hierarchical audio models", in *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, ACM, 2003, pp. 109–115.

[6] ——, "Semantic context detection based on hierarchical audio models", *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval - MIR 03*, 2003. DOI: 10.1145/973264.973282.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features", in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, IEEE, 2005, pp. 65–72.

[9] I. Laptev, "On space-time interest points", *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[10] A. Yilmaz and M. Shah, "Actions as objects: A novel action representation", CVPR, 2005.

[11] M.-y. Chen and A. Hauptmann, "Mosift: Recognizing human actions in surveillance videos", 2009.

[12] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision", in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, IEEE, 2010, pp. 253–256.

[13] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar, "Violence detection in video using computer vision techniques", in *Computer Analysis of Images and Patterns*, P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, and W. Kropatsch, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 332–339.

[14] L.-H. Chen, C.-W. Su, and H.-W. Hsu, "Violent scene detection in movies", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 08, pp. 1161–1172, 2011.

[15] E. B. Nievas, O. D. Suarez, G. B. Garcıa, and R. Sukthankar, "Violence detection in video using computer vision techniques", in *International conference on Computer analysis of images and patterns*, Springer, 2011, pp. 332–339.

[16] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin, "Action recognition by dense trajectories", in *CVPR 2011-IEEE Conference on Computer Vision & Pattern Recognition*, IEEE, 2011, pp. 3169–3176.

[17] 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*, IEEE, 2012, pp. 1–6.

[18] A. Kovashka, D. Parikh, and K. Grauman, "Whittlesearch: Image search with relative attribute feedback", *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012. DOI: 10.1109/cvpr.2012.6248026.

[19] 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.

[20] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm", in *2013 IEEE workshop on automatic speech recognition and understanding*, IEEE, 2013, pp. 273–278.

[21] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[22] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks", in *International conference on machine learning*, 2013, pp. 1310–1318.

[23] 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 (VISAPP)*, IEEE, vol. 2, 2014, pp. 478–485.

[24] C. Ding, S. Fan, M. Zhu, W. Feng, and Jia, "Violence detection in video by using 3d convolutional neural networks", in *International Symposium on Visual Computing*, Springer, 2014, pp. 551–558.

[25] C. Ding, S. Fan, M. Zhu, W. Feng, and B. Jia, "Violence detection in video by using 3d convolutional neural networks", in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, R. McMahan, J. Jerald, H. Zhang, S. M. Drucker, C. Kambhamettu, M. El Choubassi, Z. Deng, and M. Carlson, Eds., Cham: Springer International Publishing, 2014, pp. 551–558.

[26] J.-F. Huang and S.-L. Chen, "Detection of violent crowd behavior based on statistical characteristics of the optical flow", in *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, IEEE, 2014, pp. 565–569.

[27] 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*, 2014, pp. 1725–1732.

[28] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos", in *Advances in neural information processing systems*, 2014, pp. 568–576.

[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks", in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[30] L. Xu, C. Gong, J. Yang, Q. Wu, and L. Yao, "Violent video detection based on mosift feature and sparse coding", in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 3538–3542.

[31] X. Xu, W.-Q. Liu, and L. Li, "Low resolution face recognition in surveillance systems", *Journal of Computer and Communications*, vol. 2, pp. 70–77, 2014.

[32] M. Bi, Y. Qian, and K. Yu, "Very deep convolutional neural networks for lvcsr", in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[33] Q. Dai, R.-W. Zhao, Z. Wu, X. Wang, Z. Gu, W. Wu, and Y.-G. Jiang, "Fudan-huawei at mediaeval 2015: Detecting violent scenes and affective impact in movies with deep learning.", in *MediaEval*, 2015.

[34] C.-H. Demarty, C. Penet, M. Soleymani, and G. Gravier, "Vsd, a public dataset for the detection of violent scenes in movies: Design, annotation, analysis and evaluation", *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 7379–7404, 2015.

[35] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[36] E. Y. Fu, H. V. Leong, G. Ngai, and S. Chan, "Automatic fight detection based on motion analysis", in *2015 IEEE International Symposium on Multimedia (ISM)*, IEEE, 2015, pp. 57–60.

[37] I. S. Gracia, O. D. Suarez, G. B. Garcia, and T.-K. Kim, "Fast fight detection", *PloS one*, vol. 10, no. 4, e0120448, 2015.

[38] Q. Jin, X. Li, H. Cao, Y. Huo, S. Liao, G. Yang, and J. Xu, "Rucmm at mediaeval 2015 affective impact of movies task: Fusion of audio and visual cues.", in *MediaEval*, 2015.

[39] V. Lam, S. P. Le, D.-D. Le, S. Satoh, and D. A. Duong, "Nii-uit at mediaeval 2015 affective impact of movies task.", in *MediaEval*, 2015.

[40] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory", *arXiv preprint arXiv:1511.06309*, 2015.

[41] T. Senst, V. Eiselein, and T. Sikora, "A local feature based on lagrangian measures for violent video classification", 2015.

[42] M. Sjöberg, Y. Baveye, H. Wang, V. L. Quang, B. Ionescu, E. Dellandréa, M. Schedl, C.-H. Demarty, and L. Chen, "The mediaeval 2015 affective impact of movies task.", in *MediaEval*, 2015.

[43] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms", in *International conference on machine learning*, 2015, pp. 843–852.

[44] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[45] 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*, 2015, pp. 802–810.

[46] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection", *arXiv preprint arXiv:1510.01553*, 2015.

[47] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention", in *International conference on machine learning*, 2015, pp. 2048–2057.

[48] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[49] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034–3042.

[50] 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*, Springer, 2016, pp. 517–531.

[51] Z. Fang, F. Fei, Y. Fang, C. Lee, N. Xiong, L. Shu, and S. Chen, "Abnormal event detection in crowded scenes based on deep learning", *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14 617–14 639, 2016.

[52] 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.

[53] J. R. Medel and A. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks", *arXiv preprint arXiv:1612.00390*, 2016.

[54] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Pornography classification: The hidden clues in video space–time", *Forensic science international*, vol. 268, pp. 46–61, 2016.

[55] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition", in *European conference on computer vision*, Springer, 2016, pp. 20–36.

[56] T. Zhang, Z. Yang, W. Jia, B. Yang, J. Yang, and X. He, "A new method for violence detection in surveillance scenes", *Multimedia Tools and Applications*, vol. 75, no. 12, pp. 7327–7349, 2016.

[57] E. Y. Fu, H. V. Leong, G. Ngai, and S. C. Chan, "Automatic fight detection in surveillance videos", *International Journal of Pervasive Computing and Communications*, vol. 13, no. 2, pp. 130–156, 2017.

[58] C. R. Gil, *Video analysis to detect suspicious activity based ... - dzone*, Oct. 2017. [Online]. Available: https://dzone.com/articles/video-analysis-to-detect-suspicious-activity-based.

[59] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[60] 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, no. 12, pp. 2945–2956, 2017.

[61] 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)*, IEEE, 2017, pp. 1–6.

[62] T. Zhang, W. Jia, B. Yang, J. Yang, X. He, and Z. Zheng, "Mowld: A robust motion image descriptor for violence detection", *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 1419–1438, 2017.

[63] P. Zhou, Q. Ding, H. Luo, and X. Hou, "Violent interaction detection in video based on deep learning", in *Journal of Physics: Conference Series*, IOP Publishing, vol. 844, 2017, p. 012 044.

[64] S. K. Choudhury, P. K. Sa, R. P. Padhy, S. Sharma, and S. Bakshi, "Improved pedestrian detection using motion segmentation and silhouette orientation", *Multimedia Tools and Applications*, pp. 1–40, 2018.

[65] N. Donges, *Recurrent neural networks and lstm*, Feb. 2018. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5.

[66] D. Gordon, A. Farhadi, and D. Fox, "Re $\hat{3}$: Re al-time recurrent regression networks for visual tracking of generic objects", *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 788–795, 2018.

[67] A. J. Hanson, K. Pnvr, S. Krishnagopal, and L. Davis, "Bidirectional convolutional lstm for the detection of violence in videos", in *ECCV Workshops*, 2018.

[68] V. Nigam, *Understanding neural networks. from neuron to rnn, cnn, and deep learning*, Sep. 2018. [Online]. Available: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90.

[69] B. M. Peixoto, "Violence detection through deep learning", Aug. 2018.

[70] Prabhu and Prabhu, *Understanding of convolutional neural network (cnn) - deep learning*, Mar. 2018. [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[71] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition", in *German Conference on Pattern Recognition*, Springer, 2018, pp. 281–297.

[72] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6479–6488.

[73] S. Yegulalp, *What is tensorflow? the machine learning library explained*, Jun. 2018. [Online]. Available: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html.

[74] P. Zhou, Q. Ding, H. Luo, and X. Hou, "Violence detection in surveillance video using low-level features", *PLoS one*, vol. 13, no. 10, e0203668, 2018.

[75] *Convolution retrieved from https://www.cs.cornell.edu/courses/cs1114/2013sp/sections/s06_convolution.pdf.*

[76] *Deep learning for videos: A 2018 guide to action recognition.* [Online]. Available: http://blog.qure.ai/notes/deep-learning-for-videos-action-recognition-review.

[77] *Introduction. (n.d.). retrieved from https://docs.opencv.org/master/d1/dfb/intro.html.*

[78] *Violent scenes dataset description, technicolor. (n.d.). retrieved from https://www.technicolor.co innovation/violent-scenes-dataset-description.*