FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

DESİGN OF IoT BASED VİOLENCE DETECTİON SYSTEMS USİNG LİGHTWEİGHT VİRTUALİZATİON MECHANİSM

Waraz Mustafa Sadeeq ALDUHOKI

Master's Thesis

DEPARTMENT OF COMPUTER ENGINEERING

Program of Hardware Science

JULY 2022

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Computer Engineering
Hardware Science

Master's Thesis

DESİGN OF IoT BASED VİOLENCE DETECTİON SYSTEMS USİNG LİGHTWEİGHT VİRTUALİZATİON MECHANİSM

Author
Waraz Mustafa Sadeeq ALDUHOKI

Supervisor
Assistant Prof. Güngör YILDIRIM

JULY 2022
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Computer Engineering
Hardware Science

Master's Thesis

Title:	Design of IoT based Violence Detection Systems Using Lightweight Virtualization Mechanism
Author:	Waraz Mustafa Sadeeq ALDUHOKI
Submission Date:	27 May 2022
Defense Date:	01 July 2022

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Signature

Supervisor:	Assistant Prof. Güngör YILDIRIM Firat University, Faculty of Engineering	Approved
Chair:	Assoc. Prof. Ilhan AYDIN Firat University, Faculty of Engineering	Approved
Member:	Assistant Prof. Alper KILIÇ Konya Technical University, Faculty of Engineering	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Prof. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled “Design of IoT based Violence Detection Systems Using Lightweight Virtualization Mechanism ” in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Fırat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

01 July 2022

Waraz Mustafa Sadeeq ALDUHOKI

PREFACE

The nature of human beings always has led to unwanted violent incidents. Violent incidents may not be completely prevented, but their effect can be reduced with auxiliary systems, or at least the perpetrators of the violence can be found. This thesis study focuses on the search for solutions for the detection of violent events. For this, it has benefited from today's deep learning, cloud, and edge computing technologies. The combination of these technologies naturally brings along sustainability and cost problems. With the system model proposed in the thesis, the expected problems that will arise have been tried to be minimized and it has been paid attention that the proposed system model is practically applicable.

I would like to dedicate this thesis to my family. A special dedication to my parents who encouraged me to further my education and career, as well as taught me to set new and bigger goals and never stop reaching them. In addition, I also dedicate it to my dear supervisor Dr. Güngör YILDIRIM.

Waraz Mustafa Sadeeq ALDUHOKI
ELAZIG, 2022

TABLE OF CONTENTS

Page

ABSTRACT

Design of IoT based Violence Detection Systems Using Lightweight Virtualization Mechanism

Waraz Mustafa Sadeeq ALDUHOKI

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Computer Engineering
Hardware Science
July 2022, Page: xii + 53

Today, violent events have become more common in public areas, schools, and sports arenas. The early detection of these events, as well as catching the perpetrators, is vital for social order. Today's Internet of Things (IoT) technologies help to overcome these problems, and for this purpose, this thesis proposes a cloud computing-based IoT violence detection system, which is able to help security forces and related institutions. This violence detection system uses edge computing technologies, and in addition, the edge devices use Docker, which is a lightweight virtualization technology, for both software execution and system sustainability. The artificial intelligence (AI) module, which detects violence, uses deep learning models that are suggested in the literature. The deep learning module includes (CNN) with LSTM, BiLSTM, Gated Recurrent Unit (GRU) and Recurrent Neural Network (RNN) models that use a pre-trained MobileNet. The images taken by the edge devices are locally evaluated, and the urgent cases are forwarded to the relevant institutions, or to the cloud environment. Computing at the edge leads to a great reduction in the increase of cloud load, and the Docker technology allows the installation and management of related AI modules on edge devices in a fast and flexible way. The training dataset consists of violent and non-violent images. In the thesis, AI modules were trained outside the edge layer, and then the trained modules were loaded on edge devices, using container technologies. The experiments were done in local environments and the performance of the system was then analysed. In addition, the resources consumed by the edge devices were examined, with an additional discussion regarding edge device performances.

In practice, conventional violence detection systems may have disadvantages in terms of installation and execution costs. Moreover, these costs can increase even more with the use of IoT technologies such as cloud computing. Attention should always be paid to optimizing the cloud system load and customer cost, which was proven in this work, since the proposed IoT system model has several gains and advantages, with a clear practical and commercial potential.

Keywords: Violence Detection; Edge Computing; Deep learning; Convolutional Neural Networks; Cloud Computing, IoT

ÖZET

Hafif Sanallaştırma Mekanizması Kullanan IoT Temelli Şiddet Algılama Sistemi Tasaramı

Waraz Mustafa Sadeeq ALDUHOKI

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı
Donanım Bilim Dalı
Temmuz 2022, Sayfa: xii + 53

Şiddet olayları bugün kamusal alanlarda, okullarda ve çeşitli spor alanlarında sık karşılaşılan durumlar haline gelmiştir. Bu olaylarının tespit edilmesi ve faillerinin yakalanması toplum düzeni açısından hayatidir. Günümüz IoT teknolojileri burada da insanlığın yardımına yetişmektedir. Bu amaçla bu tez çalışması da şiddet olaylarının algılanmasında güvenlik güçlerine ve ilgili kurumlara yardımcı olabilecek bulut hesaplama temelli bir IoT şiddet algılama sistemini önermiştir. Şiddet algılama sistemi uçta hesaplama teknolojilerinden faydalanmaktadır. Ayrıca yazılım yürütümü ve sistem sürdürülebilirliği için uç cihazlar hafif sanallaştırma teknolojisi olan Docker'ı kullanmaktadır. Şiddet algılamayı gerçekleştiren yapay zekâ modülü, literatürde önerilmiş olan derin öğrenme modellerini içermektedir. Derin öğrenme modülü CNN, LSTM, BiLSTM, GRU ve RNN modellerinden oluşmakta ve önceden eğitilmiş MobileNet ile beraber çalışmaktadır. Uç cihazlar üzerinden alınan görüntüler lokal olarak değerlendirilmekte ve acil olanlar, ilgili kurumlarla veya bulut ortama iletilmektedir. Uçta yapılan hesaplama, bulut yükünün artmasını büyük ölçüde azaltmaktadır. Docker teknolojisi, ilgili yapay zeka modüllerinin hızlı ve esnek bir şekilde uç sistemlere kurulmasına ve yönetilmesine olanak sağlamaktadır. Eğitim için kullanılan veri seti şiddet içeren ve içermeyen görüntülerden oluşmaktadır. Tez çalışmasında ilk olarak yapay zeka modülleri eğitilmiş ve sonrasında bu eğitilmiş modüller gerçek zamanlı çalışabilen uç cihazlara konteyner teknolojileri kullanılarak yüklenmiştir. Yapılan deneyler lokal ortamlar için test edilmiş ve sistemin başarımı gözlemlenmiştir. Ayrıca uç cihazlarda tüketilen kaynak miktarı ve iletişim trafiği de incelenerek sistemin uç cihaz performansları tartışılmıştır.

Pratikte şiddet algılama sistemleri kurulum ve yürütüm maliyetleri bakımından dezavantajlara sahip olabilmektedir. Dahası bu tip sistemlerde bulut hesaplama gibi IoT teknolojilerinin kullanılması ile bu maliyetler daha da artabilmektedir. Bulut sistem yükünün ve müşteri maliyetinin optimize edilmesine her zaman dikkat edilmelidir. Bu bakımdan bu tez çalışmasında önerilen IoT sistem modeli önemli kazanımlara sahip olup aynı zamanda uygulanabilir ve ticari potansiyel de taşımaktadır.

Anahtar Kelimeler: Şiddet Algılama; Uçta Hesaplama; Derin Öğrenme; Evrişimli Sinir Ağları; Bulut Bilişim; Nesnelerin İnterneti.

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS

DL : Deep Learning

TL : Transfer Learning

AI : Artificial Intelligence

ML : Machine Learning

NN : Neural Network

CCTV : Closed-Circuit Television

KARD : Kinect Activity Recognition Dataset

VID : Violent Interaction Dataset

SVM : Support Vector Machine

CNN : Convolutional Neural Network

RNN : Recurrent Neural Network

LSTM : Long Short-Term Memory

SOM : Self-Organizing Maps

ReLU : Rectified Linear Unit

1. INTRODUCTION

Especially due to their usefulness in locating detailed materials for numerous apps and objectives such as search, summaries, and recognition action, image and video processing has seen unparalleled technical evolution. Because of the growth of violent crimes from terror organizations to single or multiple person assaults, there was the need of an increase of the development of methods for detection of events from video streams over the years. As an outcome, surveillance cameras are now widely used across the world. Human visual inspection of these videos to fins those violent behaviours, is an extremely tedious task, and is not possible to perfectly do it considering a long term, due to the huge growth in this area. Furthermore, the procedure of recognizing such instances could be erroneous, since individuals make errors, and can miss a critical event while reviewing several streams simultaneously. Although many webcams have been installed across the globe for monitoring reasons, considering that mistake rates are modest, thousands of individuals are indeed at risk. Although being only a guess, this enable us to look at the current status of violence detection systems.

To overcome this problem there is the need of unique methods to achieve a rapid identification of violence, without any human assistance. In fact, scientific research has been carried in latest years, and several firms are attempting to develop a system that identifies violent acts remotely using video. As a result, this work resorts to deep learning techniques that can train on their own, which is necessary since it could be used to anticipate possible violent behaviour before anybody else. The technology for detecting objects and motions has seen significant advances, and in this work several of these techniques are combined, in order to build a system that might efficiently identify possible violent acts that occur in our daily lives. In this thesis it was developed a system that identifies violent behaviour without any need for manual detection or the sight of a humans, involving several technologies, including object detection, movement recognition, and video categorization. Employing deep learning approaches, it is presented several techniques for recognizing violent threats and actions. LSTM, CNN, and RNN, among others, were applied to evaluate this system's action detection algorithm. The developed system is able to detect violent acts either in real-time, or from stored video feeds.

1.1 Problem Definition

Due to life and technology improvement, as well as the increase of population in the world, each country has different processes and systems to keep the security and well-being of their people. The problem of violence can be difficult to overcome, especially in crowded places where the police has limited means to reduce violence cases. During the past years, technology was able to simplify

the efforts of police by installing closed-circuit televisions (CCTVs) in the streets. In this way, they were able to record many crowded places, where the possibility of violence occurrences is higher. These videos need to be watched (real-time or not), so it is determined if there is violence or not. This type of control needs a lot of time and a high number of policemen, to be able to correctly defines what is violence or non-violence. Automatically detecting violence in videos and livestream is important in both police departments and security camera processing, with the common goal of ensuring public safety. Furthermore, it a useful solution could be people to assist security forces in the identification of violence solutions in videos, and in this way making better decisions about what to use and what not to use. However, since the concept of violence is so wide, it is a difficult challenge to solve. As a result, developing such a system without human supervision is not only a technological problem, but also a philosophical one. So, it is clear there is the need of easier stronger way, with improved techniques to solve these issues, leading to solutions with lower cost, less time, and efforts to determinate and reduce the violence. By using deep learning to create the models (MobileNet and LSTM), which are then applied on videos and livestream, as well as several public and private datasets that were recorded by webcam and Raspberry pi.

With all of that in mind, one of the purposes of this research is to look at whether a CNN might better explain the concept of violence. To characterize violence, first it was divided into more objective and tangible related notions, such as fighting, and then combined them in a CNN and RNN. 'It was also studied how to express moment processes inside the system, though many violent behaviours are defined in terms of motion. Later, it was looked at how to pinpoint violent occurrences, because many video streams have a combination of violent and non-violent sequences. By creating this technique, results show that it was able to determine violence and non-violence, and then sending this information to cloud that will be showed in many platforms and anyone can access in a simple and prompt manner.

1.2 The Aim of the thesis

The purpose of this thesis is to detect violence and non-violence in CCTV, in both real-time (live stream) and recorded videos, and later sending information via a cloud-based Internet of Things (IoT) notice, especially in public places, sport clubs and crowded places. The violence is identified using well-known deep learning techniques, including CNN and RNN; and a binary classification model has been employed to distinguish violence and non-violence videos from each other. Several deep learning approaches were analysed for visual classification, and to estimate the violence and non-violence class. These models were proven to work when deployed on a variety of devices, and in addition, the system works with raspberry pi, which is helpful to work in different environments with a high quality and speed, as well as a low cost. The output file is then sent to the cloud.

1.3 Literature Review

Crime and violence are difficult to detect and manage, particularly in crowded environments, such as music concerts, sport events, and public gatherings. To address this problem, one solution is for the city administrations to put in place monitoring systems that can detect and analyse such events. The work that is presented here combines two approaches that implement a cost-effective solution tailored to this need. The first involves sensor networks, which have proven to be a cost-effective solution in the context of smart cities, taking advantage of existing surveillance infrastructure with a quick deploy. Deep learning techniques are used in the second approach, showing exceptional results in image and action classification using a prior learning process. It was proposed in here a real-time and cost-effective solution for urban surveillance networks, by combining these two approaches [1].

M. Baba et al. proposed a Deep learning-based sensor network technique to detect violence in a smart urban environment [1]. The authors used BEHAVE, ARENA, and UCSD datasets, that are publicly annotated datasets, created from static surveillance cameras, and that contain combat or violence activities. These databases are not vast, but they are sufficient to train smaller models, which is sufficient for this application. This paper suggests a sensor network with Deep Neural Network (DNN) for violence detection, which is optimized for the city monitoring and detects aggressive activity automatically. The approach's innovation is expressed by the use of motion features, derived from MPEG streams as the DNN's sole input. It has been shown that using features embedded in MPEG streams eliminates the need for the purpose of video frames computation. As an outcome, the strategy was to improve distributed computing with limited resources. It was created a modern architecture that includes in a crescendo, very deep CNN and a transfer function detector. As a result, time and space perception are separated, and it is possible to use this design in a modified embedded system using Raspberry PI, which can be interesting for several applications.

M. Ramzan et al, this research thesis examines a variety of cutting-edge aggression detection methods [2]. The detecting technique in this paper are splitted to three classes regarding the used strategies: detecting of aggression using ML, SVM, and DL. Each individual method's approaches for extracting the features and motion detection are also discussed. Furthermore, the movie elements and data being used the different methods are explored, as well as their importance in the identification process. The suggested approach employs three benchmark datasets: a hockey fight, a hostile audience, and a movie crime. The extension of the Lagrangian theory for the identification of aggression improved the categorization effectiveness compared to cutting-edge techniques, such as Violent Flow (ViF), HOG + Bag of Words (BoW), two stream CNN, and others in terms of AUC and precision. The primary aim of this study is to compile categories, as well as to identify the most effective available approaches or strategies for detecting aggression and anomalous behaviour in

videos, by using computer vision. A systematic analysis approach was performed, in order to assess the available literature in this domain, where the systematic review's population include academic articles on the identification of aggression. The accessible research is assess using predetermined parameters, and the results are sorted into categories, and based on the usefulness of the several methodologies. To extrapolate information that is pertinent and useful from the massive amount of data, a thorough and methodical investigation is carried out. The primary goal of this review process is to compile a list of current methods for detection of violence and anomalous activity in videos, as well as to determine the most effective method for detecting violence and anomalous events. Violence classifiers are categorized based on the used classification algorithm: classification of violence detection systems (VDT) using computer vision, SVM, or deep CNN. The methods order follow are in the sequence it have been in created. A technique to detect entities and a procedure are indeed explored for obtaining characteristics. in this review. Regarding deep learning, when using more convolution layers, this technique is used to distinguish Using the dataset, recognize aggressive behaviour collection and selected characteristics. In the rest of this section, methods for identifying aggression which make use of deep learning methods will be clearly covered.

T. Zhang et al. proposes a novel strategy for detecting violence situations in CCTV, a rapid and accurate approach for locating and recognizing hostility in CCTV [3]. Because of this, a GMOF is considered in order to identify potential aggressiveness zones, which are adaptively represented as deviations from the scene's population usual activity, each movie clip passes through a dense sample of possible violent regions, in order to detect violence activity. also propose the OHOF, that is a novel descriptor that is put it to a SVM, to distinguish violence from Non-violence occurrences. Testing on a range of datasets demonstrated that the suggested method exceeds the present cutting edge, with accordance to identification accurate and pre-processing efficiency, while in congested settings. The above approaches work well in close-up scenes such as BEHAVE, Hockey and CAVIAR datasets, but not in scenes like the BEHAV dataset or the CAVIAR dataset, where human activities are limited to a restricted area. There might not be enough interest points in such scenes to make an effective decision, because most ViF descriptor cells are simply nonviolent. This paper proposes a basic but effective algorithm for detecting crime, with the main contributions being mostly focused on the three areas below [23]:

- Recommend a GMOF approach for extracting candidate aggression areas, which are adaptively model as deviations from the scene's standard crowd activity. An anomaly motion detection approach was developed in cells, which not only prevents wasteful processing and rise up the device, while increases the recognition accuracy.
- In the densely sampled candidate conflict zones, a multi-scale scanning approach is used violent cases. This eliminates the use of multi-scale spatiotemporal features

with minimal processing, as well as overcoming the challenges with varying sizes and camera distances.

- Propose a new Orientation Histogram of Optical Flow (OHOF) descriptor to differentiate aggression from these potential regions of violence. It is made by rearranging the histogram, inserting contextual data, and normalizing it. The new OHOF descriptor is very successful since it is both scale and rotation invariant.

Q. Meng et al. presents action recognition, using form and motion modalities [4]. In this article, they proposed a deep learning model focused on 3D CNN, to encode texture dependencies that does not depend on hand-crafted features or RNN architectures. The improved internal designs use the DenseNet architecture to facilitate feature reusing and channel interaction, which was shown to be more capable of collecting spatio-temporal features, needing less parameters. 1000 recordings from hockey tournaments make up the Hockey Fights Dataset, with each video being composed of 50 frames with a resolution of 720 x 576. All of the videos have shared history and human events, such as battles and everyday gaming. Regarding the Movies Dataset, it contains 200 video clips from action movies in various resolutions. In contrast to Hockey Fights, the content in the Movies Dataset varies a lot along the different videos. 246 images of crowd activity with frames resized to 320240 pixels make up the Violence-Flows Dataset. This dataset is easier to manage, when compared to the previous two, because of the multiple views, chaotic backdrop, and thick audience. Unlike conventional models, deep learning techniques employ easy to train DNN, which are able to create an end-to-end architecture that comprises edge detection, encoding, and classifications. The proposed approach aims to create a trainable end-to-end deep learning model to detect and identify battle clips in videos. A model capable of detecting violent scenes must be able to capture anomalies or patterns of intense motion in human bodies, where two of the most critical metrics are recognition performance and computational performance. As a result, creating accurate and reliable spatiotemporal modules is critical to complete this mission. In light of the above, this thesis refers to DenseNet's popular architecture before developing a deep learning model focused on 3D convolutional networks.

P. Zhou et al. proposed a violent interaction video detection system based on deep learning [5]. In this article, an image acceleration field is suggested as a new input modality for extracting motion attributes. Since each video is composed of frames that are RGB images, the consecutive frames are used to measure the optical flow field, which is used to calculate the acceleration field. Finally, the FightNet is trained using three different types of input modalities: RGB images for spatial networks, optical flow images for temporal networks, and acceleration images for temporal networks. It was determined whether a video depicts a violent incident or not by combining the outcomes of the various inputs. This article presents a multiple end-to-end strategy for identifying violence in a surveillance video feed. In the first step, people are monitoring to use an accurate

CNN model to reduce redundant frames, resulting in a reduction in the overall running time. After perceptual properties are retrieved, picture sequences and people are fed it into a 3D CNN model that is trained on different benchmark datasets, and then delivered to the Softmax function for open form. Lastly, an OPENVINO toolkit is used to augment the modelling, improve the reliability of the system. To offer researchers with a similar basis for comparison, the authors generated a violent interaction dataset (VID) including 2314 movies (1077 fights and 1237 no-fights). According the performed experiments, the proposed model is more accurate and resilient than existing techniques. FightNet is used, which is a ConvNet that models long-term temporal information to recognize violent encounters. Thanks to pre-training on the UCF101 dataset of action detection, the FightNet performs well on the suggested aggressive interaction dataset (VID), which is created from public datasets: "Hockey," "Movies," HMDB51, and UCF101. The proposed acceleration field proved to reach an increase of accuracy of around 0.3%. The FightNet delivers greater accuracy at a cheaper computing cost, when compared to existing violence detection systems.

Sh. Sumon et al. proposed a violence detection solution, by using pre-trained modules with different deep learning approaches [6]. In this article, they investigated various techniques to determine the importance of features from various pre-trained models in detecting violence in videos. A dataset containing violent and non-violent videos from various environments was built. To extract features from the frames of the videos, three ImageNet templates were used: VGG16, VGG19, and ResNet50. The extracted features were fed into a completely linked network that senses aggression at the frame level in one of the experiments. In a separate experiment, the authors fed the extracted features of 30 frames at a time to a LSTM network, while resizing the videos to 28 X 28 pixels for the CNN architecture, since they had different resolutions. To smooth the training of the model, the frames were down-sampled, where a total of 30 frames per second were derived from the images. The model of a CNN is composed of several layers, an input layer, a couple of convolutions and pooling layers, and an output layer. The CNN executes the role of a feature extractor, and the features are then transferred through a couple of completely connected layers that serve like a classifier, when combined. The proposed CNN architecture of this thesis, on the other hand, includes two convolution layers, three densely connected layers, and a sigmoid layer.

L. Ye et al. Proposed a physical violence detection system to prevent school bullying based on movement identification [7]. The physical violence detection system's architecture is composed of a Fuzzy Multi threshold classifier that detects physical bullying actions such as pulling, punching, and shaking. Importantly, the app is able tell the difference between these kinds of activity and commonplace behaviours like racing, jumping, sliding, or doing push-ups. The method employs acceleration and gyro signals to reach this purpose. The role of acting in school bullying situations and performing everyday tasks is used to collect experimental results. The simulations had a 92% classification accuracy, which is a positive finding for smartphone-based physical

bullying identification. The accelerometer and gyroscope are used to capture movement data, while RAM is used to store both data and functions, but the algorithm is executed on the CPU. When a physical bullying incident is observed, the short message system (SMS) module is used to deliver warning messages. Standing up, lying down, and walking are all daily tasks with predictable types and instructions; while physical bullying, on the other hand, is distinct since movements like punching, shaking, and pulling can come from either direction and appear to be highly unpredictable.

F. Min Ullah et al. proposed a violence detection solution that uses spatiotemporal features with a 3D CNN [8]. In this work it was developed a multi deep learning method for identifying violence that is end-to-end. People are initially noticed in the security camera footage stream, which uses a lightweight CNN model to reduce and solve the extensive encoding of meaningless frames. Secondly, a 16-frame series of humans is sent to 3D CNN, which extracts the patterns' spatiotemporal properties, sending them to the Softmax classifier. the 3D CNN model is refined, which turns the learnt model into an encoded format and adjusts it for optimum implementation at the end platform for the ultimate prediction of aggressive behaviour, when a criminal offense is detected, a warning is sent to the nearest police station or to the victim.

S. Accattoli et al. developed a violence detection in videos by combining 3D CNNs and SVM [9]. In both public and private contexts, video monitoring has long been an effective method for enforcing protection. Despite the fact that (smart) cameras are now increasingly popular and affordable, most surveillance systems are unsuccessful. Furthermore, there is no assurance that a human operator can track rare events in live video footages, requiring the use of those devices as a platform for prosecutions afhistenanted events have already occurred. Getting intelligent machines to conduct the role will allow video surveillance systems to achieve their maximum potential. In this article, a 3D Convolutional Neural Network-based approach is suggested, in order to detect battles, violent movements, and conflict scenes in live video streams. In comparison to state-of-the-art methods, this approach performed admirably on benchmark datasets: Hockey Battle, Movie Violence, and Crowd Violence. Table 1.1. shows the important details of some literature reviews, including the dataset and models used.

Table 1.1. Literature Review.

No.	Authors	Dataset	No. Videos	Used Models
1	M. Baba et al. [1]	BEHAVE, ARENA, UCSD	70 100 50	DNN
2	M. Ramzan et al. [2]	KARD	500	CNN + SVM
3	T. Zhang et al. [3]	BEHAVE, CAVIAR, Crowd Violence	80 Clips 24 Clips 246 Clips	SVM
4	Q. Meng et al. [4]	UCF	150 Clips	SVM
5	P. Zhou et al. [5]	VID	2314 Clips	FightNet
6	Sh. Sumon et al. [6]	YouTube, Twitter, Facebook,	220 Clips	(VGG16, VGG19 ,ResNet50 DenseNet,) + LSTM
7	L. Ye et al. [7]	-	(Accelerometer, Gyroscope) Sensors	Fuzzy Multithreshold classifier
8	F. Min Ullah et al. [8]	Hockey Fight, Crowd Violence, Movie Violence	500 Clips 100 videos 123 videos	MobileNet (CNN)
9	S. Accattoli et al. [9]	Hockey Battle, Crowd Violence, Movie Violence	1,000 Clips 246 videos 100 videos	CNN +SVM

1.4 Outline of the thesis

The thesis' subsequent sections are separated into four chapters. The thesis' theoretical foundation is outlined in Chapter two, including CNNs' architectures (e.g. MobileNet), RNNs' architectures (e.g. LSTM), as well as descriptions of deep learning and classification methods. In Chapter Three, the dataset and many of the assessment approaches used in the thesis will be presented in depth, as well as the software and hardware requirements for the thesis. The outcomes of the violence detection training and testing are discussed in the fourth chapter. The model's performance was assessed using a variety of measures and compared to that of other similar studies. In addition to a comparison to other existing approaches This chapter also looks at how successfully the network has tested the results.

2. THEORETICAL BACKGROUND

The following topics will be discussed in this section of the thesis: the detection and philosophy of violence; deep learning techniques; CNN architectures; and IoT.

2.1. Violence Detection

The majority of the studies in the domain of identifying violence is focused on the poor traits, e.g. retinal outflow, patterns, brightness, or other area properties are often extracted [10]. It is challenging to collect useful and discriminating characteristics in the field of video-based violence detection, due to human figure variances. Size, view point, shared opacity, or moving sceneries are indeed the major reasons of variances. Some approaches aimed at identifying violent situations via distinguishing violent systems like flame, blood, weapons, and audio [11, 12]. Nevertheless, the majority of approaches concentrated on identifying violent incidents. However, these approaches might not have been suitable to detect movies having poor picture quality. The limitations of this technology, for example poor detections rates and high false alarms rate, restrict the results. Additionally, these qualities are incompatible with common camera systems, which frequently don't acquire audio data. Another of the earliest efforts is Nam et al., where they suggested multisensory property limit values [13]. Volume or energy, and also abrupt variations in 2 of sum entropy, were considered for audio functionality. Devices use visual elements to compute active action in order to detect quick movements and pixel color levels for blood detection. Cheng et al. suggested a perceptual method for detecting fundamental acoustic happenings, which include shots, fires, motors, and automobile breakdowns [14]. HMMs was trained in audio activity detection and targeting, as well as simulating connections among several occurrences, using Gaussian mixtures to extract complicated semantics contexts.

The previous systems dealt with distinct occurrences and searched for each one separately. Bermejo et al. proposed an approach that seeks to classify, as well as to recognize, aggression via motion. They used a Bag of Visual Words (BoVW) technique, which includes an optic flow histogram to describe localized movement, and low-level characteristics like as Space-Time Interest Points (STIP) and the Motion Scale-Invariant Feature Transform (MoSIFT), an adaptation of a SIFT picture classifier to videos [15]. Using several STIP-identified summaries was difficult to deal with temporal knowledge to encode a bag of characteristics for every shot, and then a linear SVM was trained to classify movies, yielding comparatively superior results. This methods stresses on the relevance of detecting violence using both moving and space-time aspects. While characteristics derived from attention spots could capture much helpful info for detecting violent acts, it may not even be resistant to scene modifications in a CCTV footage situation. Furthermore,

the massive numbers of key points collected might directly harm the efficiency of the method. A MoSIFT technique was proposed by Xu et al. to extract low-level video description, where the MoSIFT classifiers were chosen using a non-parametric Kernel Density Estimation (KDE) and sparse coding, and the classification methods were processed to remove redundant data and to prevent racially prejudiced features [16].The LaSIFT characterization, together with a conventional BoW model for classification, was presented by Sense et al., as a model for categorization of violent video, by using visual data and hierarchical motion features. LaSIFT features are tested using the BoW platform, which outperformed the SIFT and MoSIFT classifiers on the Hockey Fight dataset and the Community Violence dataset. This method was able to describe a unique technique for detecting violence that could successfully characterize the evolving nature of violent films. The use of a SIFT classifier offers a new feature for evaluating aggression, by including the path Lagrange electronic logging calculation. All the features were then processed, by using an expanded BoW technique. Zhang et al [17]. created a novel description of violence detection called MoWLD, an identical descriptor as MoSIFT, which mixes a WLD histogram that describes the area, with a HOF that depicts point-of-interest activity. Afterwards, all the features were analyzed in the same ways as. The descriptions retrieved that are surrounding those attention sites can capture both appearance as motion pictures, which were confined to locations of the interest points, excluding helpful information from outside neighborhoods of key points. That are many other models for detecting violence in the literature that are not discussed in here [18].

To begin this work, it was recommended to use a GMOF to extract prospective violent zones. Furthermore, a new identifier named as the Optical Flow Orientation Histogram (OHOF) was suggested to be used in the potential locations. To distinguish between violent and non-violent incidents, the descriptors were finally loaded into a linear SVM. The GMOF algorithm, on the other hand, will have poor racist and discriminatory effectiveness, if the backdrop is cluttered or moving. Data et al. discovered aggression by analysing the trajectory of motion and alignment of such a user's limbs [4]. Precision profiles is necessary to determine the location of the limbs, however due to the severe obscurity, division of the image is rather challenging. Several studies use statistical properties that are retrieved in spatio activity patterns, such as mean, variance, standard deviation, centroid, area, etc., to characterize violent videos [19, 20]. Algorithms that use these attributes benefit from minimal computer cost, but their classification accuracy is restricted. Deniz suggested a new strategy that took advantage of its most significant trait of aggressive behaviour: high acceleration patterns. Severe velocity characteristics are efficiently calculated, by applying the Radon transform to the power spectrum of subsequent images. Its static environment has an impact on severe accelerating patterns, leading to a high false alarm. Regarding the identification of violence at overcrowded scenarios, it is more difficult due to severe concealment of the moving throng. Some statistics show considerable changes, in the case of an aggressive swarm behaviour

was considered. The ViF descriptor, which is gathered for small frame sequences, is used to describe these data. ViF characteristics are classified using a linear SVM, which is an approach that detected mob aggression in a statistically efficient manner. Inside the data list of non-behaviour, nevertheless, the performance of the ViF-based technique was considerably worse. A novel feature called as Oriented Violent Flows (OViF) can recognize non-crowded violence in videos based on a ViF 7 description [21, 22]. Depending on motion orientation statistics, OViF characteristics are able to explain movement size changes however this strategy might not have a good function in congested situations. Huang et al. uses a data approach, based on optical how elds, to identify aggressive mob behaviour, which uses statistical features and optical ow descriptor (SCOF) statistical characteristics to describe the video frames. Those SCOF characteristics were classified between normal or violent incidents with a linear SVM. Furthermore, this method is confined to the SCOF characterization, that can only describe frames and cannot capture appearance aspects.

The goal of this thesis is to create a system that can efficiently detect violent actions in both generalized and packed scenarios. The surprising development of deep convolutional video movement networks, like Temporary Segment Networks, has led to the possibility of using these methods to apply for this purpose [23]. Ding et al. collected spatial and temporal parameters from 3D convolution layers and classified them, however, neither of proposed approaches supports the changeable duration of films. Due to the depth of the temporal axis, the computing time for 3D convolution increases at a high pace. Dong et al. advocated employing LSTM as data consolidation inside the detection of violence, by using a CNN and later fusion, extracting features from the raw pixels, optical ow pictures, and acceleration ow maps. Zhou et al. created a FightNet to depict the complicated interaction between visual violence and three different inputs modalities, which includes Image pixel saturation for space networking, optically ow, and short term acceleration pictures for the network. The identification of violence showed positive outcomes in the performed experiments. To avoid overfitting, UCF101 was used to pre-train CNNs, together with a video-based detection and tracking [24]. Unfortunately, networks trained on targeted datasets might not always work well, particularly when the datasets are substantially different from of the pre-trained dataset, such as the Crowd Violence dataset. In this respect, deep learning-based algorithms are hampered by the absence of a suitably big training dataset for violence. Furthermore, DNNs are show higher computational expenses, needing the use of increasingly powerful devices. However, this thesis proposes a different deep learning technique to a detection of video-based aggression, driven either by enormous success of DNNs in computer vision [25].

2.2. Violence Philosophy

The first stage in the thesis is to find out exactly what the authors have been looking for in the literature. With studying the notion of violence until this, but what really defines violence and

non-violence? In this chapter, it has been discussed classification of videos and live stream, as violent or non-violent, and how it might help with development's experimental stage.

2.3. Deep Learning

Deep learning is a subset of machine learning, being a relatively new discipline that has already made significant contributions to artificial intelligence (AI) in a variety of fields. Deep learning, which has an architecture composed of several layers of neural networks, produces incredibly good outcomes, due to the large amount of analysed data. The depth of these solutions is proportional to the number of layers, and while a traditional network has 2-3 layers, a deep learning network may have up to 150 layers. The data is handled in these levels throughout the learning process, which usually needs a GPU processing capability. AI is being applied in several fields, which include item categorization, object discovery, face recognition, voice recognition, violence detection, among others [26].

While deep learning falls well short of the capabilities of the human intellect, it allows algorithms to classify data and make incredibly accurate predictions. The DNNs attempt to mimic human brain function by letting it to "learn" from massive volumes of data, but they fall well short of its capabilities. While a single-layer neural network can still make approximations, adding more and more hidden layers, can help with an increase of accuracy. Deep learning is used by many AI systems and services, in order to increase the automation, by performing analytical and physical tasks without the need for human intervention [27].

Deep learning technology is used in everyday products and services, as well as anticipated developments (such as voice assistants, sound TV remotes, card fraud detection, self-driving cars, diseases detection, and so on). DNNs are a type of neural networks that attempt to replicate the human mind by combining sophisticated inputs, weights, and bias. These components work together to effectively recognize, categorize, and describe dataset elements. DNNs are composed of several layers of linked nodes, each enhancing and refining the prediction or classification, where the advancement of calculations through the network is known as forward propagation. The input and output layers are the visible layers of a deep network [28]. While the input layer is a collection of input data composed of values that are unrelated to one another and corresponding to the input type; the output layer is linked to the relevant income scenario, such as the predicted weather (clear/rainy) or a patient's health status (healthy/sick). The term "deep" in deep learning refers to a neural network's several hidden layers, and deep learning models are trained using a huge amount of input data, as well as intermediate information calculated in the several hidden layers of the network, learning to interpret the features directly from the data without requiring human participation. These systems are designed to be flexible and adapt to changing conditions, and once a neural network has assessed the outcomes of a certain item, it can discern if the thing is the same

the next time it appears. Other interesting characteristic is that a neural network does not sense objects in the same manner that humans do; rather, it distinguishes objects based on their own set of properties. Figure 2.1 depicts the link between deep learning, machine learning, and AI in a graphical manner.

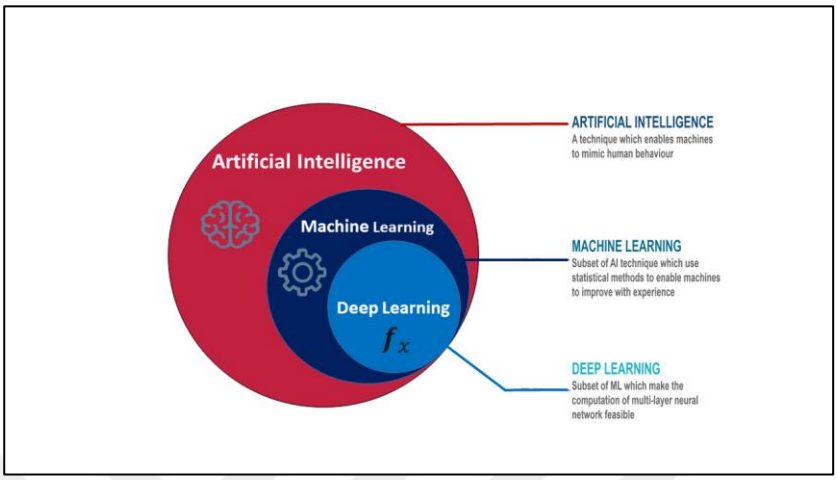

Figure 2.1. Deep Learning [28].

2.4. Techniques of Deep Learning

A CNN is a form of DNN that is especially intended to analyse pixels as input, and has been widely used in pattern recognition systems. CNNs are used in image enhancement, image and video recognition, or recommendation systems, including natural language processing (NLP). Conventional neural networks aren't designed towards image recognition, and therefore must be given pictures in smaller chunks. CNN's "neurons" are structured in the same way as those in the prefrontal cortex, which is the region within humans and animals required to process visual inputs. Pre-trained neural networks' piecemeal image processing difficulty is avoided, by rearranging the multiple units (neurons) in such a manner that they span the whole field of vision.

2.4.1. Neural Networks

Multilayer perceptron are neural networks where the neurons are linked to the continuous layer, being widely used to identify Fully Connected Neural Networks. Fran Rosenblatt, an American psychologist, developed this type of network in 1958, entailing the transformation of the model into basic binary data inputs. The following two functions are included in this model:

- Linear function, which shows a line that multiplies the input values with a constant, as the name implies.
- Non-Linear function, which can be divided into those sections:

- Sigmoid Curve, which is a function that has a range between 0 to 1, being perceived as an S-shaped curve.
- Hyperbolic tangent (tanh), which is a S-shaped curve with a range between -1 to 1.
- Rectified Linear Unit (ReLU): The Rectified Linear Unit (ReLU) is a single-point function that returns 0 when the input value is less than the set value, and the linear multiple, when the input value is greater than the set value

2.4.2. Convolutional Neural Networks-CNNs

CNNs are a sort of traditional artificial neural network, which are intended to mix biological with computational features. When it comes to machine vision, it is widely used and extremely successful technique. In a CNN, there are unique elements that make a picture unique, which are employed to recognize images, and it functions in a similar fashion to how the human brain does. It is possible to define an object in a human brain that automatically identifies pictures. In a CNN, a similar discrimination is carried out, since it separates an object or picture from its attributes and provides an abstract definition. If the desired outcome is not attained or the received values do not fit the system expectations, more than one element can be adjusted or evaluated when running the CNN algorithm. One of the issues encountered is that there is no clear picture of what should be configured, in order to get the best machine learning results. The first step is to ensure that the algorithm is compatible with the training data, and because of that, the training data results should be assessed for acceptability. This expectation may be close to human perception for some systems; however, this varies depending on the application. It is expected to get a good result on the validation dataset and test dataset, after getting a good result on the training dataset, and the cost function appears after the results of all the datasets. If it is determined to make a correction following the cost function result, the algorithm can be updated so test results could match the training set [27].

CNNs are one of the most adaptable models for both image and non-image data analysis.

The layers in these models can be separated into four categories:

- a single input layer, which is typically a two-dimensional arrangement of neurons for interpreting primary visual data comparable to picture pixels.
- a single-dimensional output layer of neurons, which in certain CNNs analyse pictures on their inputs, via the distributed linked convolutional layers.
- A third layer, known as the sample layer, that is present in CNNs to limit the amount of neurons engaged in the respective network levels.
- one or more linked layers connecting the sample and output layers.

Deep learning techniques rely on neural networks. Similarly, they consist of an input layer, hidden layer(s), and an output layer. Each node has a connection to the other nodes in the next layer. If a node's output provides a predefined limit value, that node is activated, and data is sent to its next layer of the network. Neural networks come in a variety of sizes and forms, and they're used for different purposes. RNN, for example, are frequently employed in speech recognition and NLP, although CNN are more frequently used in classification and computer vision applications. Detecting entities in images before CNNs required laborious, and time-consuming feature extraction approaches. CNNs, on the other hand, use concepts from matrix multiplication to discover patterns inside an image, making them more scalable for image classification and object recognition applications. They can, however, be computationally intensive, necessitating the use of graphics processing units (GPUs) to train algorithms. The convolution layer shares all the weights by applying the convolution operation to the given input image. In reality, the convolutional layer's task is to detect and recognize features at various locations in the input image/feature map. One of the application areas where CNNs are mostly used is the area of diagnostic purposes in the field of medical image analysis.

2.4.2.1 Input Layer

The input layer is the first layer of a CNN, as was already explained in previous sections. This is the layer where raw data enters into the network. For the success of the model, the magnitude of the input data is a critical factor, since if a high-dimensional picture is used, a lot of memory, training time, and testing time are required. However, more data has the potential to boost networking success. The memory need and length are decreased correspondingly if a low-dimensional picture is used, however the depth or effectiveness of the created network may be quite reduced. To achieve the best results, a proper size must be chosen, which can be a difficult task. To offer a sample of a model that might be regarded as a fundamental structure, the deep learning Mobilenetv2 filter, which is another common model when using 4x4 size 7x7, has been used as a drift net ZF 11x11 and 2x2 shift the filter size. A first layer containing lower filter input is selected in varied sizes to ensure the retention of numerous original image pixels.

2.4.2.2 Convolutional layer

This can be considered as the most important component of a CNN. By applying filters to the pictures, a features layer is obtained. The majority of such filters are multidimensional and have pixel values, and the picture data of the specified lower flow rates is used in conversion. By applying a filter to the data from a previous layer, it is possible to build an output which looks like the original picture Filters of various sizes can be made for this matter, and in this thesis, sizes of

5x5x3 were used, where the height and width of the matrix are 5x5, and the depth is 3. Assuming the input data is 256x256 pixels, the picture is first processed (1x1 scrolling (stride)) with the given filter starting from the top left corner. When the first line is finished, a pixel is sent to the bottom line, where the identical action is done. The outputs matrix is formed by scanning the entire picture (all pixels). The values between the original picture and the picture after filtration are multiple among each, whose multiplier value is gathered and stored, to construct an output matrix. In Figure 2.2 it can be visualized a convolution of a 3 x 3 filter around an input [29].

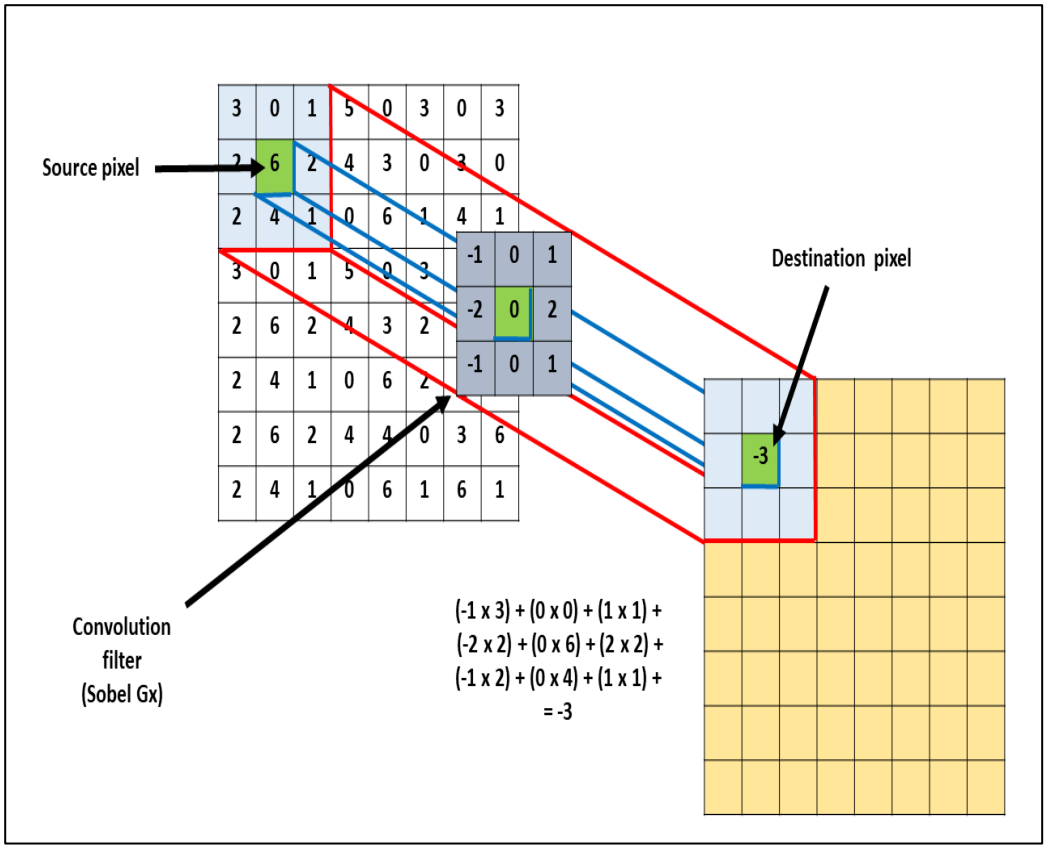

Figure 2.2. Visualization of 3 x 3 filter convolving around an input.

2.4.2.3 Activation Layer

Inside a neural network, an activation function specifies how the weighted sum of an input is turned into an output. An activation function in a deep CNN explains how the counted weight inputs is translated into an output from a unit or units in a layer.

A "Transfer function" is a term used to describe its activation layer function. If the output range of the activation function is limited, it is called as a "Squashing function." Since most activation functions are non-linear, "non-linearity" within a layer and network setup is referred to, as shown in a Table 2.1 below. The activation function in a neural network explains how the linear combination of an input is transformed into an output, from a neuron or network in a layer. Looking

at a CNN, its activation function illustrates how the recorded weighted outputs are transformed into such an output from a unit or cells in a layer [30]. As shown in a Table 2.1 below.

Table 2.1. The Transfer functions.

Squashing function	Activation function
$\boldsymbol{\sigma(X) = \frac{1}{1+e^{-x}}}$	Sigmoid
$\boldsymbol{\max(0.1x, x)}$	Leaky ReLU
$\boldsymbol{\tanh(x)}$	tanh
$\boldsymbol{\max(W_1^T X + b1,\quad W_2^T X + b2)}$	Maxout
$\boldsymbol{\max\{0, X\}}$	ReLU
$\begin{cases} X & X \geq 0 \\ \alpha(e^x - 1) & X < 0 \end{cases}$	ELU

2.4.2.4 Pooling layer

Following the convolutional layer, adding a pooling mechanisms is a frequent approach of layer placement in a CNN. Particularly when a non-linearity (e.g., ReLU) is applied towards the extracted features created by one convolutional layer. In a given model, each pattern could be used one or even more times. This layer operates independently on each feature map to make a new collection of pooled feature maps with nearly identical feature sets.

Pooling is comparable to applying a filter on feature maps, since it requires a pooling technique to be chosen. This pooling layer operation of filter is likely less expensive than the feature map; as example, it's often 22 pixels with 2 pixel strides applied. This demonstrates how well the pooling layer has been used to reduce the size of each feature map by a factor of two, resulting in a convolution layer by one with the number of values or pixels. The effect of applying a pooling layer and creating down sampled or aggregated feature maps is a summary version of characteristics gathered in the input. Pooling has the ability of detecting minimal details inside the area of the features, when compared with the convolutional, which makes that layer useful. Pooling was used to introduce variance to the model for local translations [30].

2.4.2.5 Fully connected layer

Even as the title implies, two-phase neurons in a layer are entirely coupled to each and every activation in the previous layer. Fully linked layers are always found at the network endpoint, and that layer takes the input area (so what outcome of the previous conv, ReLU, or pool layer was) and produces an N-dimensional vector, with N being the number of classes the algorithm should classify. The number of classes N in our test is equal to 2, then the first outcome would be violence, and the next would be non-violence [30].

2.4.2.6 Dropout

A small total sample size in a training dataset is expected to quickly generalize DNNs. Subsets of neural networks with different model mixes are known to reduce the fitting problem, but train and developing multiple models increase both the computing time and cost. Any model may be used to mimic a large number of different network models via randomly dropping off nodes during training. A dropout is a normalization approach for DNN of various sorts that lowers the probability of overfitting and improves generalization error, being at the same time computationally inexpensive, as well as extremely effective. Giant neural networks trained on small datasets show a high probability towards overfitting, when in the training stage. When this happens the model cannot be flexible enough to be evaluated on new data, including a test dataset, resulting to poor results performance of the model, and increasing the inaccuracy in predictions. Problem may be reduced by training all possible unique neural networks along the same dataset, at the same time predictions are combined out of each design. In reality, this is not achievable, but a small set of unique models known as an ensemble can be used to simulate it. Dropout may be implemented in a single layer inside a neural network, and it is possible to apply it in a variety of layers, such as dense fully connected layers, convolutional layers, and recurrent layers such as the LSTM network layer. Dropout can indeed be employed across partially partial hidden levels of the network, as well as the visible or input layer. However, dropout is not an option. This dropout approach is displayed in Figure 2.3 just on output layer [30].

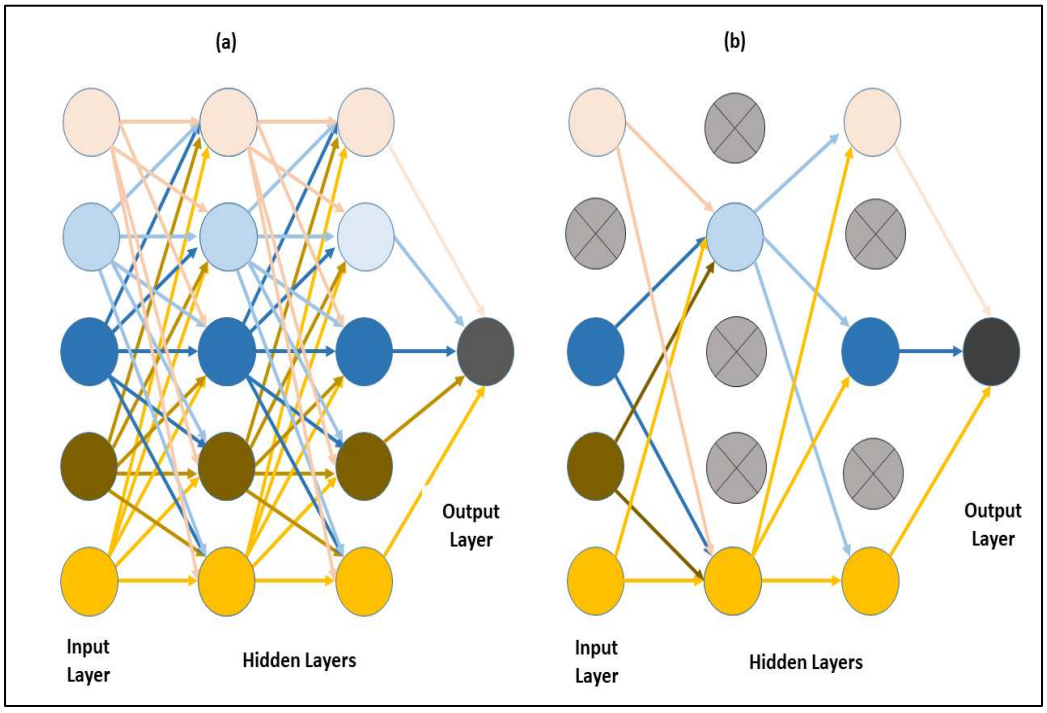

Figure 2.3. Dropout Prevention Plan (a) Traditional neural network (b) after applying dropout.

2.4.2.7 Batch Normalization

Because DNN models are usually big, it is difficult to train tens of layers that are susceptible to the training system's randomized initialization settings and activation. When weights are altered after each mini-batch, the balance of inputs for layers in the below network may shift, which might be one source of the training difficulties. It might lead the learning process to indefinitely track a moving object, which is called as an external covariate shift. Normally, the values are adjusted to a balanced scale, before feeding them together into machine learning or deep learning algorithm. Part of the reason for normalization is to guarantee that the used algorithms can be appropriately extended.

Batch normalization is indeed the adding of layers to a DNN, in order to enhance effectiveness and stability. This new layer only performs the standardizing and normalizing processes on inputs from the previous layer [31]. Batch normalization uses the term "Batch" for a specific reason, since Batch data is a collection of data input, and is used to train most neural network models. The normalizing technique is also executed in batches rather than as a single input, when using batch normalization. The normalizing technique is also performed in batches, rather than as a single input with batch normalization. The unique algorithm of the normalizing technique is also done in batches, rather than as a single input for batch normalization. Batch Normalization's unique algorithm is provided below, where the given data batch input data (BatchNormInference) conducts the basic steps [32].

Input:

$$B = \{x_1 \ldots . m\}; \tag{2.1}$$

Parameters to be learned:

$$\gamma, \beta \tag{2.2}$$

Output:

$$\{\Upsilon_i = \mathrm{BN}_{\gamma,\beta}(x_i)\} \tag{2.3}$$

mini-batch mean

$$\mu\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2.4}$$

mini-batch variance

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu\mathcal{B})^2 \tag{2.5}$$

normalize

$$\hat{x}_i \leftarrow \frac{x_i - \mu\mathcal{B}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{2.6}$$

scale and shift

$$\hat{y}_i \leftarrow \gamma\hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(\chi_i) \tag{2.7}$$

The The standardization of fully connected layers is performed by normalizing all the potential actions in a mini-batch over all the locations. The pair of variables *y* and *B*, for example, are learned per feature map rather than in each stimulation. The frequencies of input features to each layer are normalized into another range via batch normalization, which lowers training algorithm fluctuations as the lowest point is approached, and convergence is sped up. Batch normalization also introduces modest noise to every layer, since every tech is blended with other data in a mini-batch, which decreases the effect of the classifier. In reality, any network employing batch normalization requires a smaller dropout rates.

2.4.3. Recurrent Neural Networks (RNNs)

RNNs were introduces to help in the prediction of sequences; for example, the LSTM algorithm. RNNs only run with data sequences. The RNN make use of the previous knowledge as an input value, and a result, it can help in the implementation of short-term memory, allowing for the efficient time-based data systems. As aforementioned, there have been three sorts of RNN designs that may be used to help with problem solving:

- The One-to-One, which has only one input and output. This is a standard NN with set sender and receiver lengths. Classification of images is usually performed using the One-to-One application.
- One-to-many: One input is linked to more than one output sequences, for example picture captioning, which uses multiple words from a single image. Whenever produced a single input, a one-to-many RNN produces several outputs. It accepts a set input size and produces a series of data.
- Many to one: uses a series of inputs to produce a single outcome. When a single output is produced from many input units or a succession of those, many-to-one is utilized. To show a fixed output, a series more inputs are required. This form of RNN is commonly used in Trend Analysis.
- Many to many: leads to a series of outputs, such as video classification. Many to many seems to be a technique for creating a series of expected output from a series of more input units, as shown in Figure 2.4.

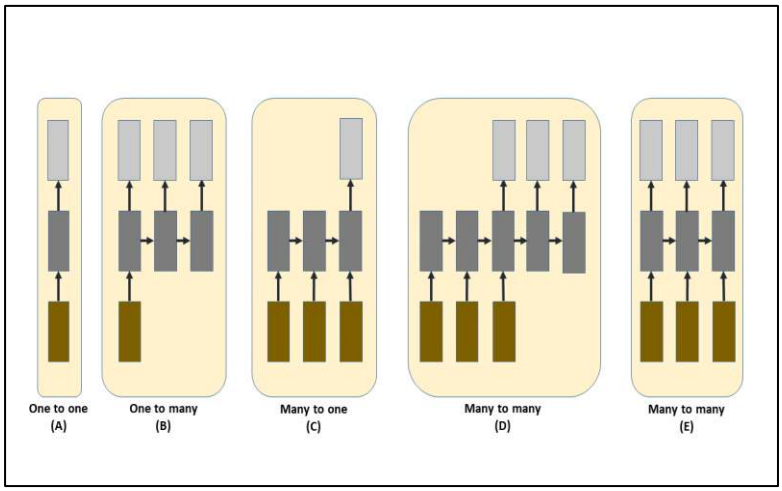

Figure 2.4. Types of RNN.

2.4.3.1 Long Short-Term Memory (LSTM)

LSTM networks have data modification, in contrast to more traditional machine learning techniques. This network is a RNN and is able to analyse not just scatter plots (like pictures), and moreover complete data batches (such as speech or video). Memory-based algorithms that have been able to successfully forecast results over a number of time periods. The names "Input," "Output," and "Forget" were also given to each of the three gates. Whether interactions moving into rather than returning out of another cell is facilitated either by three gates, or even the cell itself would be capable of holding information over a series of different lengths which that operator may choose. All of this is shown in the Figure 2.5.

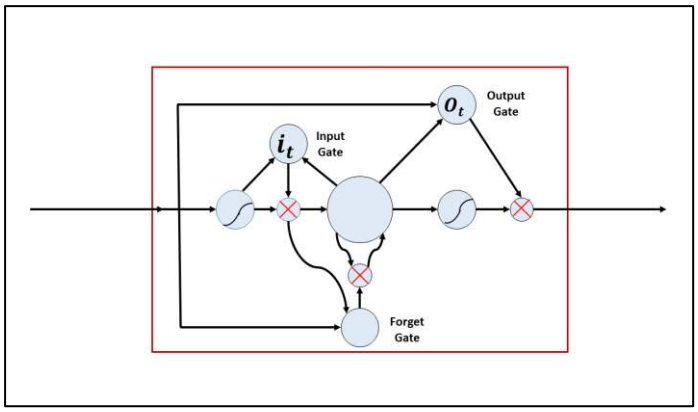

Figure 2.5. LSTM with three gates.

In an LSTM, there are three different gates: input, forgetting, and output. This gates control where fresh input should be allowed (input gate), if the data should be deleted (forget gate), or should it affect the output just at time - step (output gate). a representation of an RNN including three gates it has LSTM equations:

h(t-1): the preceding LSTM block's output (at timestamp t - 1)

w (x): Weight for such gate(x) neurons

σ: functions of sigmoid

b (t): input at the present time

x(t): biases for the gates(x)

i(t) input gate

$$i(t) = \sigma\,(w(i)\,[h(t-1), x(t)]\, + b(i)) \tag{2.8}$$

f (t) forget gate

$$f(t) = \sigma\,(w(f)[h(t-1), x(t)] + b(f)) \tag{2.9}$$

o (t) output gate

$$o(t) = \sigma\,(w(o)[h(t-1), x(t)] + \; b(o)) \tag{2.10}$$

2.4.3.2 Gated Recurrent Unit (GRU)

The GRU belongs to the most recent generation of RNNs, and is analogous to the LSTM in its capabilities. GRU removed the cell state from the equation, shifting its data transmission to the hidden state. In addition to this, there are just two present gates, namely an update gate and a reset gate. The update gate performs operations that are analogous to those of the forget and input gates of an LSTM. It decides what information should be removed as well as what new information should be included. Another gate that decides how much past data to forget about is known as the reset gate. [33]. GRUs need less tensor operations than LSTMs, which means that they can be trained more rapidly than the latter, which is shown in the Figure 2.6.

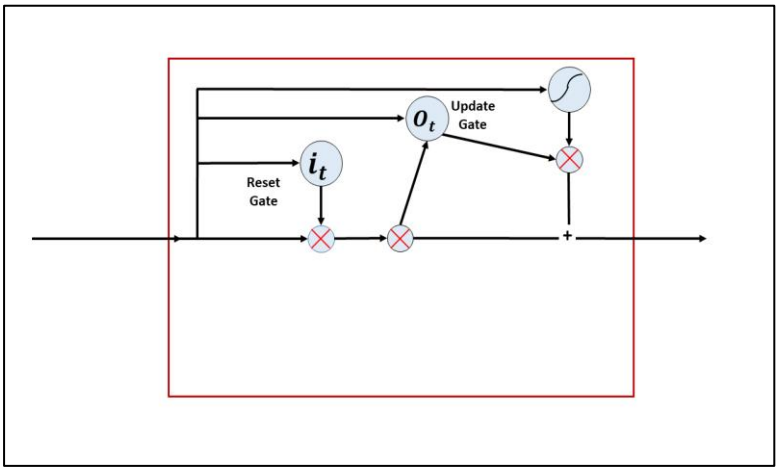

Figure 2.6. GRU model.

2.4.3.3 Bidirectional Long Short-Term Memory (BiLSTM)

Bidirectional Long Short-Term Memory, or BiLSTM for short, is a type of RNN that is applied in the natural language processing field. In contrast to a conventional LSTM, input is allowed to go in either direction, and the data collected from either side may be utilized. In addition to this, it is a powerful tool for modelling the sequential dependencies between words and sentences, and this may be done in either way. In a nutshell, BiLSTM consists of an extra layer of LSTM that inverts the direction in which information is processed, revealed that the additional LSTM layer processes the input sequence backwards rather than forwards. After that, the results of the two LSTM layers are merged using a number of different procedures, such as taking the mean, the total, the product, or concatenating the results [34]. A BiLSTM is shown in the Figure 2.7.

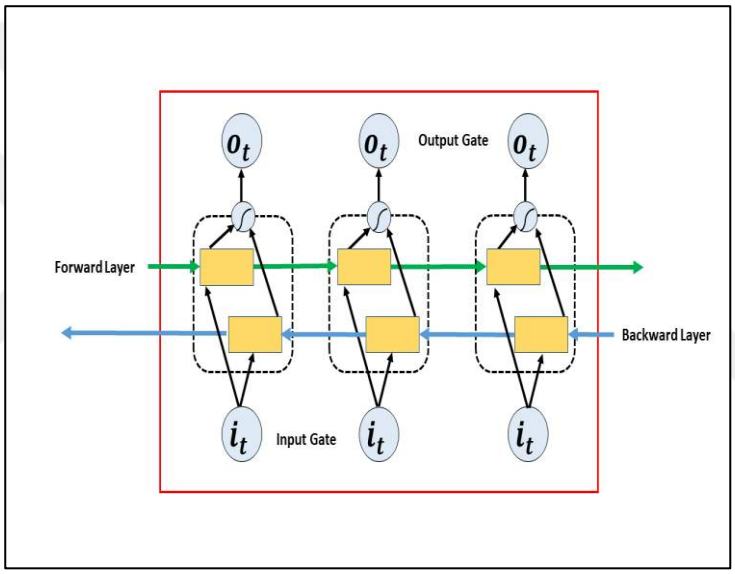

Figure 2.7. Bidirectional LSTM.

2.4.4. Generative Adversarial Networks

It's a hybrid of two DNN techniques: a generator and a discriminator. The Discriminator helps to distinguish between true and fake data, while the Generator Network generates fictional data. Both networks are successful, since this Generators keeps making fake data that is equal to genuine data, while the Classifier continues to identify real and unreal data. In the event that a picture library is necessary, this Generator network would generate simulation result for the real photos. A deconvolution neural network would subsequently be created.

2.4.5. Self-Organizing Maps

Self-Organizing Maps (SOM) run with the reduced number of random variables, and unsupervised data. As each synapse links to its input and output nodes, the output dimension is set

as a two-dimensional model in this sort of deep learning approach. The SOM adjusts the weight of the nearest nodes or Best Matching Units as each data point competes for its model representation (BMUs). The weights vary according to a BMU. Since weights are node characteristics, the value indicates the node's location in the model.

2.4.6. Boltzmann Machines

Because there is no set for orientation in this network model, the nodes are connected in a circular pattern. This deep learning approach is utilized to generate model parameters because of its uniqueness. The Boltzmann Machines model is stochastic, in contrast to all the preceding deterministic network models.

2.4.7. Deep Reinforcement Learning

Before going into the details of Deep Reinforcement Learning, it's important to understand what reinforcement learning is. This type of learning is the process through which an agent interacts with its environment to change its state. The agent may monitor and respond appropriately, assist a network in achieving its goal by engaging with the circumstance. This proposed system has an input layer, another output layer, and countless hidden multiple layers, with both the state of the environment serving as the input layer. The method is based on making many efforts to predict the future pay out of each action taken in a given circumstance.

2.4.8. Auto encoders

This This model, one of the well-known deep learning algorithms, functions autonomously depending on its inputs before applying an activation function and decoding the final output. As a result of the bottleneck, fewer types of data are produced and the underlying data structures are utilized. Auto encoders are classified as:

- Sparse – The generalization strategy is used when the hidden layers outnumber the input layer to reduce overfitting. It restricts the loss function and stops the auto encoder from using all of its nodes at the same time.
- Denoising - Here, a modified version of the inputs is randomly turned into 0.
- Contractive - When the hidden layer outnumbers the input layer, a penalty factor is added to the loss function to prevent overfitting and data duplication.
- Stacked - When another hidden layer is added to an auto encoder, resulting in two stages of encoding and one phase of decoding.

2.4.9. Backpropagation

The backpropagation (back-prop) is the essential mechanism enabling neural networks to learn from any mistakes when performing data prediction in deep learning. Propagation represents the transfer of data across a channels. In the time of choice, the complete system can work according to signal propagation in the forward direction, transmitting any data about network flaws back in reverse. First, the network evaluates the parameters and selects the data. Second, Backpropagation is weighted using a loss function. Third, the detected error is transmitted backwards to rectify any inaccurate settings.

2.4.10. Gradient Descent

Gradient is a slop that may be expressed as a connection between features in mathematics. The link between the expected errors and the parameters could be expressed as "x" and "y" in the model. Because the attributes in the model are dynamic, the error can be increased or lowered with little adjustments.

2.5. CNN architectures

A CNN architecture was made up of a series of separate layers using a spatially separated algorithm to turn an input size into more of an output volume (for example, containing class scores). Layers are often employed in a variety of ways. Several different CNN architectures were created throughout the years to meet real-world challenges. Yann Lecun created LeNet, the first practical application of CNNs, in the 1990s. It was used to read zip codes, numbers, and other data. LeNet-5, a 5-layer CNN that achieves 99.2%on insulated text detection, is the most recent study.

2.5.1.1 MobileNet

Howard Sandler (2018) developed the MobileNet approach [35], which provides a good balance of performance with computation time. MobileNet reduced computing time values by using occurring and Depth-wise convolution, despite maintaining good classification performance. In MobileNet, persistent constraint block is proposed for recovering convolution layer. As a consequence, MobileNet may achieve excellent accuracy of classification while reducing training time. The depth-wise separates the convolutions used in MobileNet design as a sort of normalized combination that separates a normal convolution it into depth-wise convolution and a 1x1 pointwise convolution. MobileNet's depth-wise convolution provides a separate filter for every input. Even via the pointwise convolution, the outputs of a depth-wise convolution are joined using a 1x1 convolution. A traditional convolution process combines inputs in one step to produce a new set of

results. This can be divided into two layers by the depth-wise separable convolution: one for filtration and the other for combination.

The MobileNet design is constructed on depth-wise separable convolutions, except from its first layer, which would be a complete convolution. By describing a network in such fundamental parameters, we may easily examine network architectures to find an effective network. All layers of the model were proceeded by batch-normalization and ReLU non-linearity, with the exception of the last convolution layer, which has no non-linearity and feeds this onto the softmax layer for classification. Depth-wise convolution, 1x1 pointwise convolution, batch-normalization, and ReLU after each convolutional layer contribute to pass information from non-linear to the normalized layer. In the depth-wise convolutions and the first layer, down sampling is accomplished via stride convolution. Before the entirely connected layer, a last pooling layer decreases the pixel density to 1, [35].

2.6. Internet of Things

IoT is critical to changing threes and continuing urban development. Machine learning and IoT are critical to enhance livelihoods and sustaining urban development. They're believed to be the most effective methods for detecting environmental violence, since it is quite challenging to use IoT technologies for real-time applications. The smart city community has shown interest in AI and the IoT in smart city contexts in prior years. An information ology is a vision for information and communication, such as the IoT, which enhances quality of life while also assuring cities' economic and long-term growth. Many automation issues are thought to be solved by combining IoT with deep learning. IoT systems like the Jetson Nano are extremely fast and efficient, and they can process data quicker than any other edge device currently on the market. Deep learning models may be applied in real-time with the aid of an edge device, although processing a huge number of sensors that produce millions of readings per second presents major challenges.

In 1999, a patron of a Radio Frequency Identification (RFID) planning profession developed the word "internet of things," but only recently this technology has become useful in the real world, especially thanks to the rise of smartphones, embedded and powerful information exchange, virtualization, and high-performance computing [36]. Considering a society in which billions of items can detect, communicate, and exchange data over public or private Internet Protocol (IP) networks; data needs to be gathered, evaluated, used to start action on these networked items on a continuous basis, offering a wealth of insight for management, monitoring, including decision-making. In fact, this would be a world where IoT would be present in everyday life.

The is a network of physical items, according to the most prevalent definition. Technology has changed into a network of devices of all shapes, which include cars, smartphones, home appliances, toys, cameras, and so on hospital equipment and factory automation, people, and

buildings, all connected, all communicating and providing information based on predefined protocols in accordance with appropriate detections, positioning, tracing, safe and control, and even violence detection real actual stream-based monitoring, upgrade, and analytic capabilities. We divide the IoT into three different categories:

- Humans to Humans.
- Humans to things.
- Things, machines to things, and machines.

IoT is an idea and a framework that recognizes the increase of the environment of a range of things that could communicate with each other and cooperate other things to develop new software solutions, using wireless and wired connections and determines systems. In this situation, the research and development barriers to building a smart environment are enormous. An environment in which real, digital, and interactive experiences are combined to create smart environments that improve energy, maritime transport, cities, and some other areas [36].

As depicted in the Figure 2.8, the term "Internet of Things" refers to a broad concept of things, particularly daily items, that are readable, identifiable, locatable, accessible via media sensing devices, or controllable over the Internet, regardless of effective communication (whether via RFID, wireless LAN, wide area networks, or other means) [36].

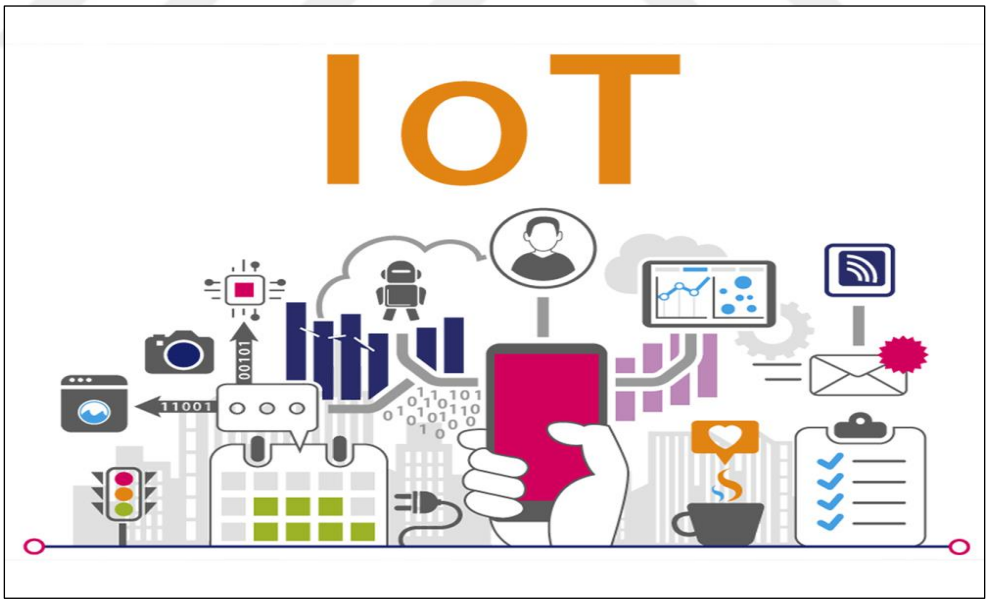

Figure 2.8. Internet of Things[36].

3. MATERIAL AND METHOD

This section includes the material and methods used in this thesis, the material such as data collected for the work as well as the datasets used in the work, followed by the models description that were used in the proposed system, as well as the hardware and software used.

3.1. Data collection

To address our violence detection issue, we had explored and advocated for both the usage of machine learning and deep learning architectures. These techniques are supervised learning models that require a lot of training data to gain knowledge on. The used datasets in the violence detection operations are crucial, since the reliability of the training data has a significant impact on the classification outputs. As a result, collecting this dataset is a critical stage in our procedure. It was gathered the growth of real Violence (RWF) dataset from different platforms, which is comprised of 2,000 video clips taken by CCTV in real-world scenarios, to make violence identification more feasible in modern applications. The data collecting process was performed by searching on the website a set of terms connected to violence (for example, genuine fights, violence under video, and violent occurrences) to get a collection of videos [37].

3.2. Dataset

Since the purpose of this project is to identify violence, RWF2000 dataset has several types of violence were chosen to be included: sidewalk fighting, social rioting, and sports field clashes. This dataset also included films of non-violent political protests, gallery celebrations after playing sports events, and non-violent street marches for environmental issues, to represent non-violent actions. This dataset is composed of videos from several streaming video websites, such as social media platforms, such as Facebook and Twitter. The videos have varied quality and durations according to the various platforms. There were 400 movies gathered for each class, giving a total of 800 recordings. The gathered films were modified in such a way that the non-violent sections of the violent videos were chopped off.

The UCF-Crime dataset consists of 128 hours of videos. It comprises of 1900 raw and unedited CCTV recordings from the actual environment, comprising genuine occurrences such as assault, shooting, arson, and so on. These anomalies were chosen due to their substantial influence on community risk, some samples of the UCF-Crime dataset. The dataset can serve two major purposes. First, the universal recognition of violence, with all aberrations (violence) in one category and all normal actions (nonviolence) in another. Second, for the identification of each uncharacteristic behaviour [38]. Some sample images of this data set are given Figure 3.1.

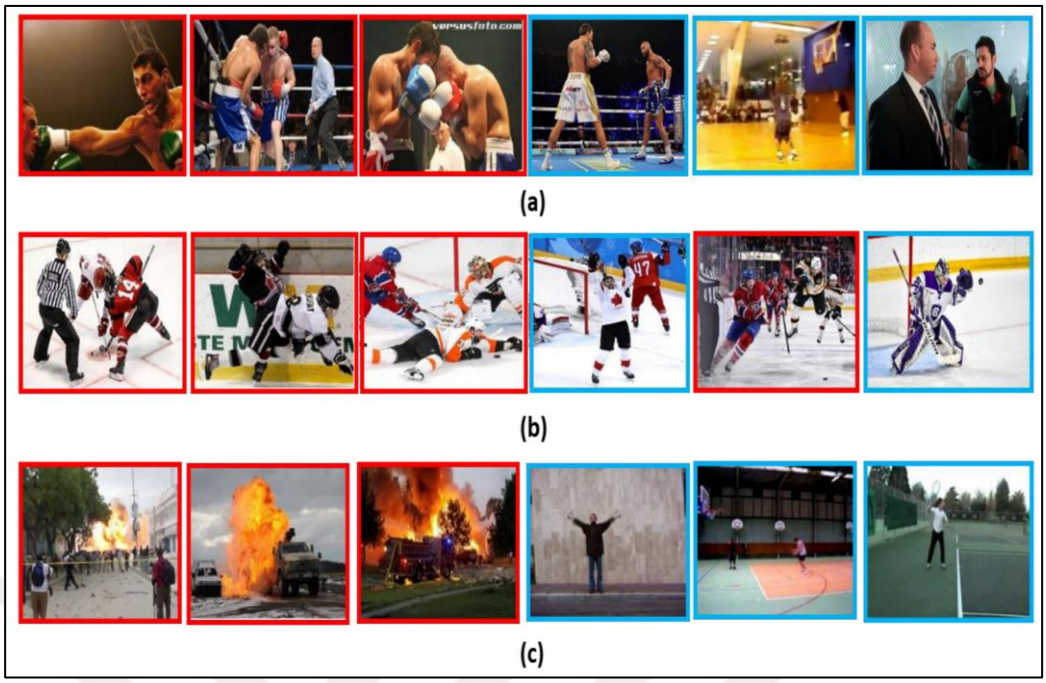

Figure 3.1. UCF-Crime Real life violence samples.

Datasets used in this work can be explained as shown below in the Table 3.1: the first dataset is RWF2000 [37] which consist of 2000 videos with 30 frames for each video the total used in our work is 800 videos according to our hardware resources capability. The second one is UCF-Crime [38] that include 1900 videos with 60 frames per each video which 400 videos used in this thesis as hardware resource capability. Finally, the personal videos used that contains 40 videos with 30 frames per each video.

Table 3.1. Datasets Table.

Number	Author	Dataset name	Contents	Video Frames	Used in Model
1	M. Cheng [37]	RWF2000	2000	30 Frames	800
2	Ulla [38]	UCF-Crime	1900	60 Frames	400
3	Own videos	Personal Videos	40	30 Frames	40

A private dataset was also gathered, and many sources were used, like Duhok government local dataset, personal accounts, and local channels, reaching a total of 40 videos, which were already recorded inside Duhok City. Two family members were also asked to act as volunteers for 20 more videos, to use as test of the solution. And all of the videos used in the thesis are such an example for those videos shown in Figure 3.2.

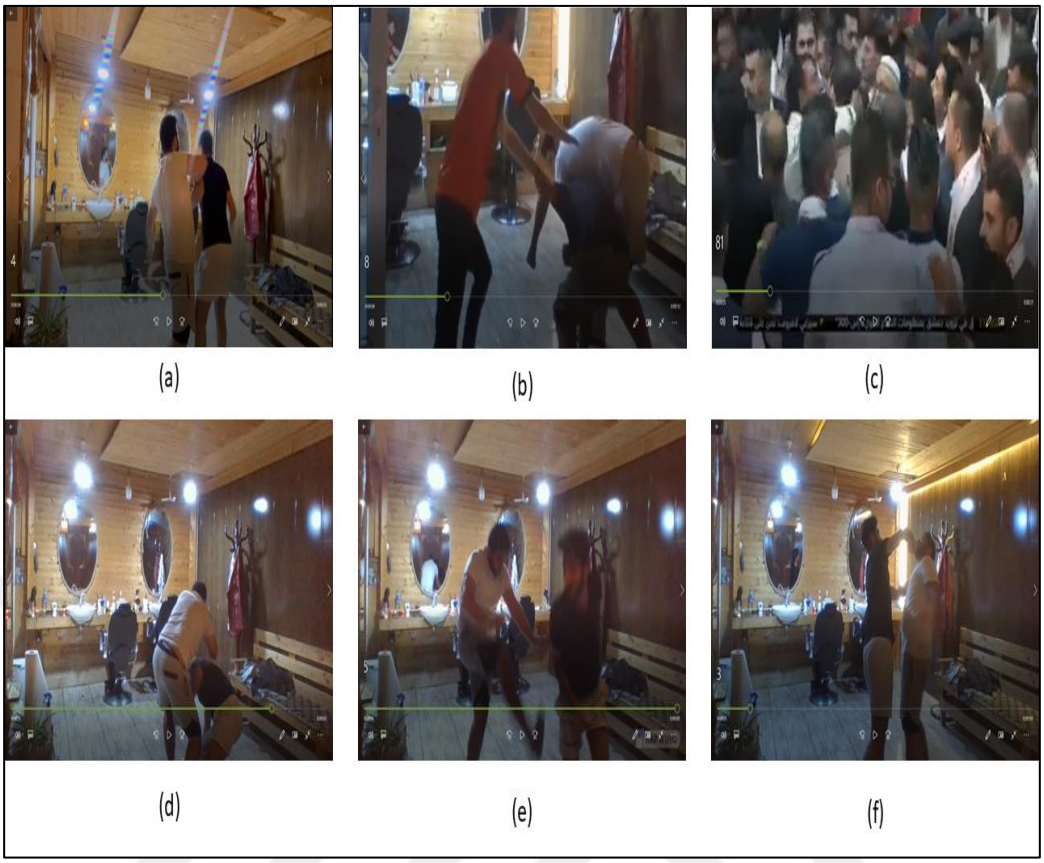

Figure 3.2. Own videos

3.3. Hardware and software configuration

In this section the hardware and software configuration will demonstrate each of them separately to define all requirement used by this work, the resources of the hardware can affect the result of the work, so all the used hardware is declared, as well as the software.

3.3.1. Hardware configuration

When A PC with 32 GB of RAM was used, with a disk of 100 GB. This was able to upload 400 videos for training and 400 videos for testing the solution. Table 3.2 shows all of the other information regarding the PC, the hardware configuration. The test environment includes the personal computer, processor (Intel Core i7-10230 CPU @ 2.60 GHz), graphic cards (Intel® HD Graphics 4000) and the Windows 11-64 bits.

Table 3.2. Hardware Identifications.

Hardware	Characteristics
Memory	32 GB
Processor	Intel Core i7-10230 CPU @ 2.60 GHz
Graphics	Intel® HD Graphics 4000
Operating system	Windows 11, 64 bits

For performance, the Raspberry Pi 4 used is close to entry-level x86 PC computers. A 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decodes at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability are some of the key features of this product (via a separate PoE HAT add-on). The board's dual-band wireless LAN and Bluetooth compliance certification is modular, allowing it to be designed into end products with significantly reduced compliance testing, reducing both cost and time to market [39].

3.3.1.1 Raspberry Pi

Raspberry Pi, like other single-board computers, is small – about the size of a credit card, but it can accomplish everything a larger, more strength computer can do, just not as rapidly.

The Raspberry Pi series has evolved over time, from the Pi 1 to the Pi 4, and even a Pi 400. In most generations, there has been a Model A and a Model B. Model A is a less costly variation with less RAM and fewer ports (such as USB and Ethernet). The Pi Zero is a smaller and less expensive version of the original (Pi 1) generation. as the line-up of the different versions is shown in Figure 3.3.

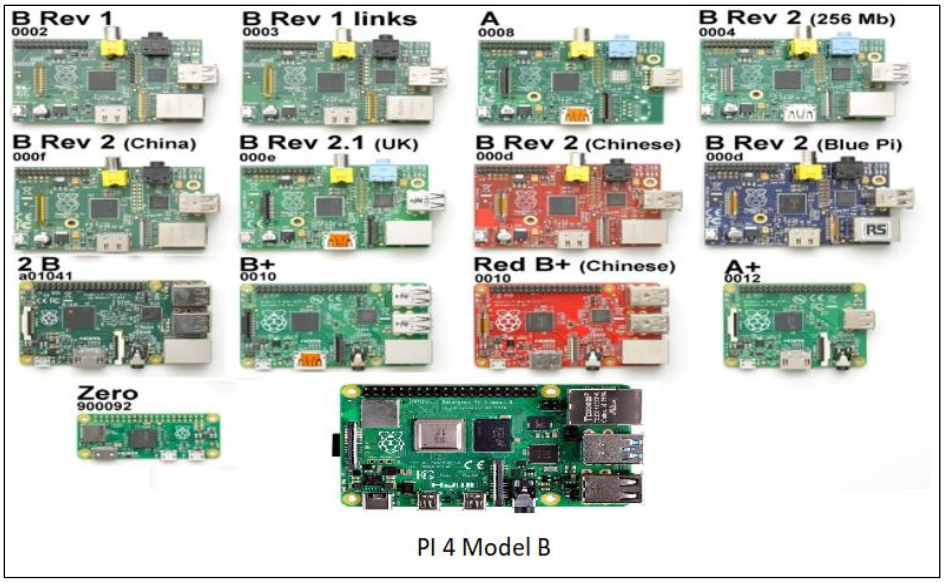

Figure 3.3. Types of Raspberry Pi [40].

The Raspberry Pi 4 Model B is the most recent addition to the popular Raspberry Pi computer line. When compared to the previous-generation Raspberry Pi 3 Model B+, it offers ground-breaking improvements in processor speed, multimedia performance, memory, and connectivity while maintaining backwards compatibility and similar power consumption [40], The used Raspberry Pi 4 Model specifications is given in Table 3.3.

Table 3.3. Raspberry Pi 4 Specifications.

Hardware	Characteristics
Memory	LPDDR4 1GB, 2GB, or 4GB (depending on model)
Processor	Broadcom BCM2711 is a quad-core Cortex-A72 (ARM v8) 64-bit SoC with a clock speed of 1.5GHz.
Interconnection	IEEE 802.11b/g/n/ac wireless LAN at 2.4 and 5.0 GHz, Bluetooth 5.0, BLE Gigabit Ethernet There are two USB 3.0 ports on this device. There are Two USB 2.0 Ports.
GPIO	A 40-pin GPIO header is standard (fully backwards-compatible with previous boards).
Sound and video	There are two micro HDMI connectors on this device (up to 4Kp60 supported) MIPI DSI display port with two lanes MIPI CSI camera port with two lanes Ports 4-pole stereo audio and composite video.
Multimedia	OpenGL ES 3.0 graphics; H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode).
Support for SD memory cards	Micro SD card slot is used to load the operating system and store data.
Power of input	5V DC (minimum 3A1) through USB-C connection 5V DC (minimum 3A1) through GPIO header Enabled for Power over Ethernet (PoE) (requires separate PoE HAT).
Environment	Temperature range: 0–50°C.

3.3.1.2 Pi camera

The Pi camera module shown in Figure 3.4 may be used to capture high-definition films and photographs. It is simple to use; skilled users may supply a wealth of information for you to learn from. The cameras are 5 megapixels as shown in the Figure 3.4. 1080p30, 720p60, with a fixed focus camera and a capture. It moreover includes a wired connection that really should be connected to that same Raspberry Pi's CSI port Using Simple Python programs, it is possible to have the Pi

snap photographs or record movies. The camera module is widely used in home security and in recognizing objects, image processing, and machine learning [39].

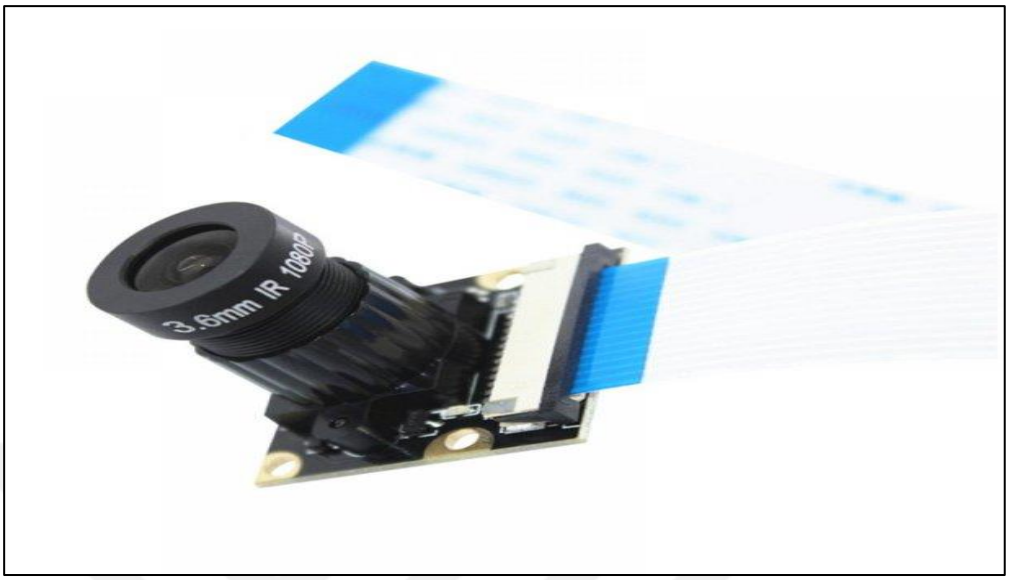

Figure 3.4. Pi Camera.

3.3.2. Software configuration

Regarding DNN parameters, it is possible to use the standard values, or try to improve them. Standard improvement procedures are mainly enhanced in the following aspects: while learning rate decrease and GDO are both defined as adjusting the variable changing step size, learning rate attenuate is defined as changing the informed directions. These improved optimization methodologies could potentially help batch or probabilistic gradient descent approaches. In this case, the learning rate was 0.0001, the maximum number of epochs was 100, the Validation Frequency was 3, and the dataset was shuffled for each epoch. Finally, all of the CNN models were trained with a 32 batch size. The general meta-parameters that the CNN has used are shown in Table 3.4.

Table 3.4. CNN network's training option hyper-parameters and values.

Training option Parameters	Value
Optimization algorithm	Adam
Learning rate	0.0001
Epochs	100
Shuffle	Every epoch
Batch size	32
Validation Frequency	3

3.3.2.1 Google Colaboratory

Deep learning apps can be applied in a range of places, including web search engines, social network recommendations, natural language recognition, and ecommerce insights. Complex operations on large datasets are common in this type of application. As a result, parallel computing has historically been thought of as a way to speed up the training operation. GPUs are prime options for performing such a parallel task. This type of accelerator is widely available, easy to use, and has a high GFlops/Dollar rate. NVIDIA GPUs are also supported by the major deep learning architectures. Google Colaboratory, often known as "Google Colab" or simply "Colab" is a gathering and processing that allows machine learning models to be prototyped using powerful machines including such GPUs and TPUs. It creates an open Jupyter environment without the need for a server. Google Colab, like the rest of the G Suite, is completely free to use [41].

3.3.2.2 Docker Container

Docker is an open-source platform that simplifies its deployment using software containers for apps. Even though they may function isolated from one another because of the operating server, some application containers resemble analogous to lightweight virtual machines. Docker requires functionality found in newer Linux kernels to effectively operate correctly, and meanwhile, on Mac OSX and Windows hosts, a virtual machine running Linux is necessary. Today, one primary meaning of configuring such VM is through Docker Toolbox, which uses Virtual Box directly, even though there are actions to merge this capability within Docker itself, by using the OS system's native virtualization capabilities. Docker runs directly upon that client on Linux platforms.
Because although container technology has indeed been present for over a decade, Docker, a relatively new contender, is now a standout among the top breakthroughs, since it provides additional capacities than previous technologies lacked. It provides the ability to build and manipulate containers at first, and aside from that, developers may simply bundle programs inside lightweight Docker. Those virtualized programs may be used without modification everywhere. Furthermore, somewhat on identical hardware, Docker can express more virtual scenarios than alternative innovations.

To summarize, Docker could easily collaborate alongside third-party tools that make it easier to create and deploy Docker containers. Docker containers are simple to set up in a cloud system. Whenever apps are deployed inside Containers, Docker comes with a feature to streamline the process, and it also offers an extra layer of deployment engine on top of a Container environment, where programs are virtualized and then run. Docker is meant to get a speedy as well as lightweight framework in which code may be executed quickly, as well as an additional resource of the competent development process to remove the software from the desktop during testing prior to

construction. Docker Client and Server, Docker Images, Docker Registries, and Docker Containers seem to be the four primary major characteristics of Docker [42], as seen in Figure 3.5.

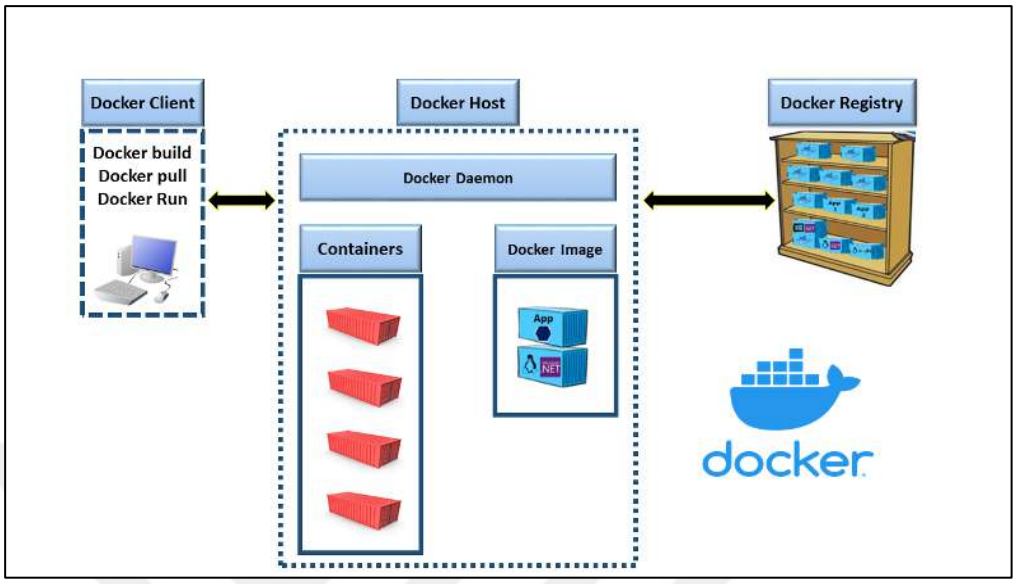

Figure 3.5. The four primary major characteristics of Docker[42].

3.3.2.3 Zabbix

To operate with a high-performance computer cluster, there is already an online system that is widely used in a diversity number of studies, which is Zabbix. This tool makes the production of the starting script easier and faster, and can monitor many resources shown in the Figure 3.6. The existing web system includes information on High-Performance Computing Cluster (HPCC), a tool for register, a listing of research initiatives, a collection of software, HPCC user guides, and the ZABBIX networking and deployment monitoring tool. It also keeps track of the amount of processing time and resources utilized for the execution of each job. Every administrator, as shown in Figures 1 and 2, has access to information regarding the cluster's utilization, where the use of processors in the HPCC is depicted. The graph represents the number of available CPUs, whereas the line represents the number of processors still being used. This number of used CPUs is represented on the vertical axis, while the days in the calendar interval are represented on the horizontal axis.

There seem to be a variety of detection and control solutions available, ranging by the most basic towards the most sophisticated. Several of them are free, while others are quite costly. Most contemporary network monitoring solutions gather information via network nodes using the Simple SNMP and store it in a database for later use. These systems can monitor a variety of metrics (loss of connections, network speed limits, interface failures, CPU and memory utilization, among others) and are supported by various devices SNMP works based on a request-response architecture. To make the proposal independent of the Raspberry pi or other devices, an SNMP

aware network switch was used in this investigation. The testing facility sends a request to all endpoints for their network monitoring information, as well as the devices respond. These replies are stored in a database, and some characteristic thresholds can be set to define critical conditions that need the activation of an alert. Server Orion software, as described in [43], is utilized as performance monitoring in this investigations.

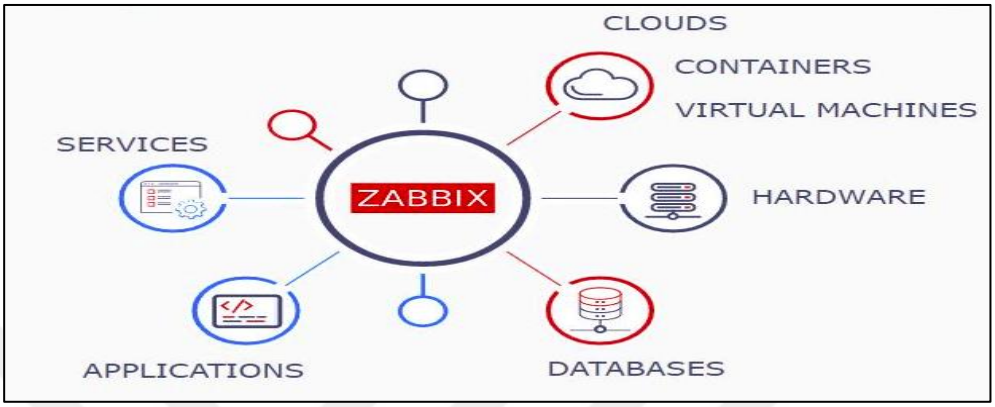

Figure 3.6. Zabbix[43].

3.4. Proposed system

The proposed system has been developed to improve deep learning technique, where it was merged the following: MobileNet and LSTM, BiLSTM and GRU; and the main purpose was to identify violence and nonviolence. It has been trained and tested as previously mentioned, starting with receiving data from Raspberry Pi camera live videos, recorded or archived videos as shown in the Figure 3.7. In the end, all videos that have been labelled as if there is violence or nonviolence will be saved and transferred to the Docker container. While the model is operating, the Zabbix tool is used to measure the raspberry's resources, including CPU, RAM, bandwidth read/write rates, raspberry loading system, disk utilization and queue, disk read/write rates, and disk space consumption.

MobileNet [35] was combined with three distinct models, which are referred to as MobileNet with LSTM, MobileNet with BiLSTM, and MobileNet with GRU. The vast majority of the available resources were merged using MobileNet2 with LSTM, VGG16 [6], VGG19 [6], and DenseNet [6]; their models utilized videos, and their outcomes were shown as visuals portraying either violence or calm. The proposed approach is distinct from the models that have been described in the past because the model in makes use of live streams, videos, and local movies. It then transfers the result to a Docker container in the form of a cloud result before transmitting it to Zabbix, and in order to monitor the model's performances; it is assessed the percentage of videos that were labelled violence or non-violence.

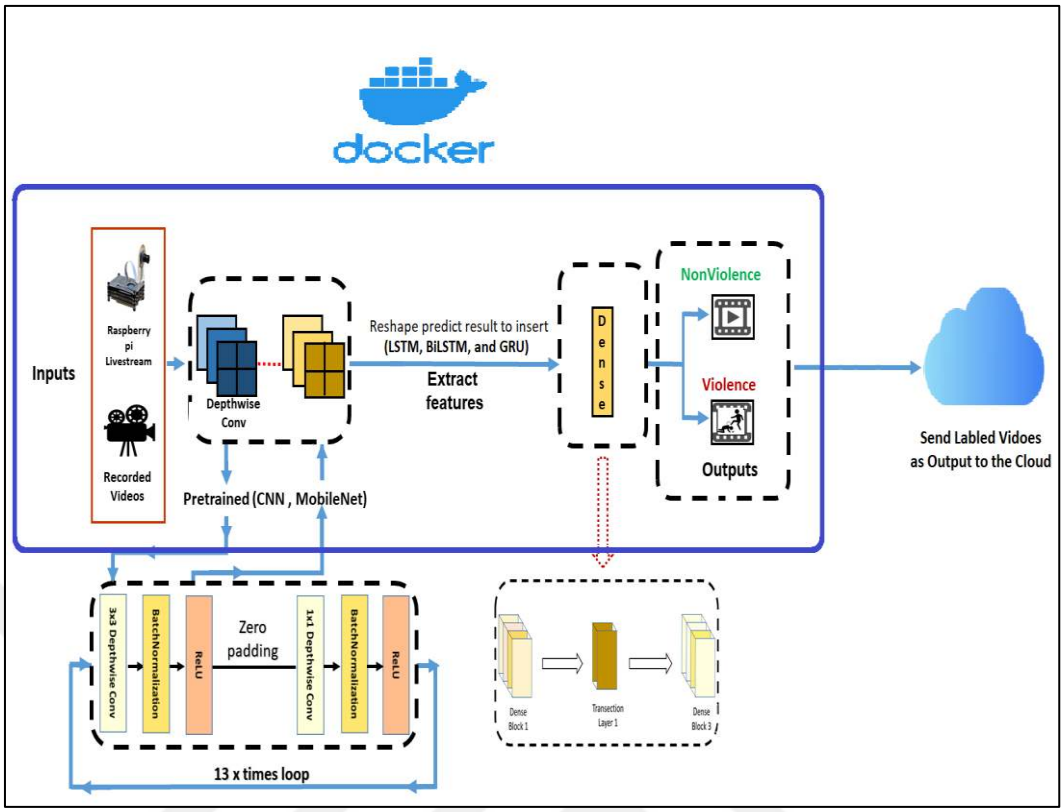

Figure 3.7. The design of the proposed system

3.4.1. Data pre-processing

The The videos are based on dataset resources, such as: raspberry pi, pc device camera, and recorded local videos, are reduced with Numpy to 160x160 pixels, and videos will later be saved as pickle file This is shown below in the Figure 3.8.

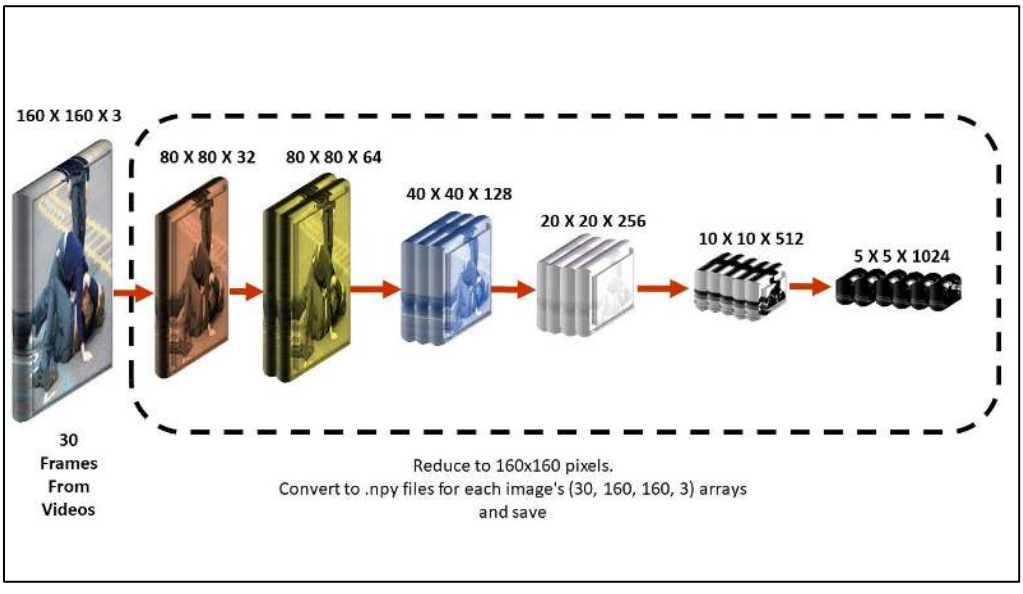

Figure 3.8. Numpy array transformation process [44].

3.4.2. Creating the Train and Test set.

After the the pre-processing stage, this stage will load violence and non-violence video pickle files, then combine data and shuffle them randomly, which will combine the violence and non-violence datasets. Afterwards, the combined dataset is shuffled, and saved it as a pickle file. The training and test sets are splitted, based on the defined percentage of each set. The data labelled as (training, testing) is set to save both the training and test sets as Numpy arrays and then the sets will be trained and then tested.

3.4.3. LSTM Model Training and Evaluation

After the third stage reshaped output, it becomes the input for the fourth stage for the LSTM Model to be trained and evaluated, which will begin with Feature datasets being loaded to make the LSTM model an establishes function. The LSTM Classification Models will examine the LSTM Model that has been trained and use it as a test-set to evaluate the LSTM Model. Then the trained LSTM Model accuracy and loss will be plotted.

3.4.4. Recorded videos and real-time (Pc webcam and Raspberry Pi camera) violence detection.

MobileNet will be used to load the model files, and afterwards MobileNet's output will be used as an input for the trained LSTM model. After that, check each of the characteristics that are provided, add a caption, and finally save the completed video clip. The input and output routes are configured, which add a label to the video clip and decide whether or not the violence is real.

3.5. Google cloud MEASURES

At the beginning we have worked on computer to develop model, due to our computer storage is 32 GB so we are able to uploaded 400 videos to be used in the train. It was decided to work on the personal computer and then upload it to Google cloud. The process map can be shown in Figure 3.9. below.

The Raspbianos application is deployed on the micro SD card and conducts the necessary tasks. The micro SD card (128Gigabytes) is maintained in the Raspberry Pi and started up, allowing the raspberry pi to run Raspbianos. The V.N.C viewer program or Putty would be used to control a Raspberry Pi from a laptop and Mobile. The Raspberry Pi has a MobileNet SSD to be deployed and loaded pre-trained models.

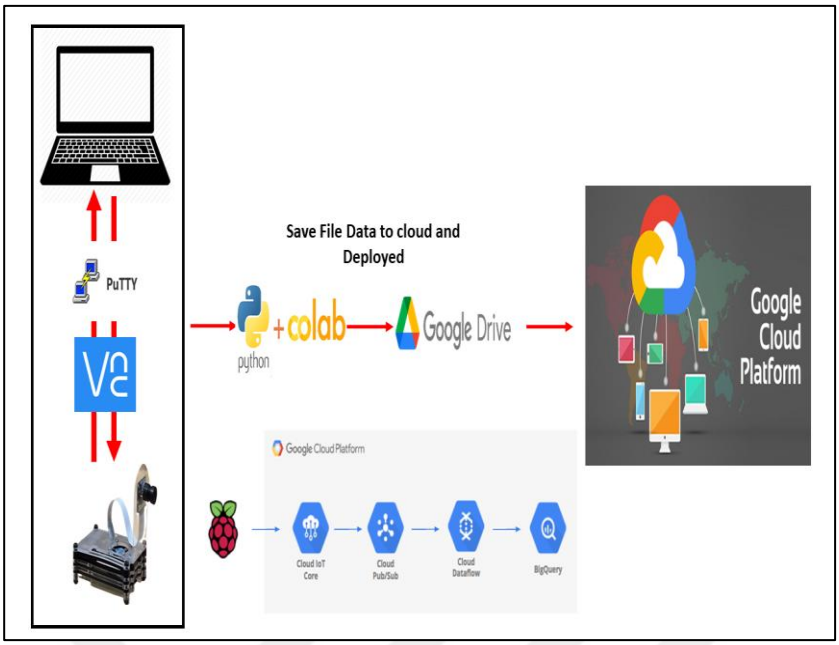

Figure 3.9. The process of map.

The proposed system can categorize directly in the Raspberry Pi as Violence and Non-Violence and save the environmental event record. Afterwards, the results are transferred to the cloud server. In this way the PC was used as a server and raspberry pi as a client.

3.6. Zabbix Resources Measures

The Zabbix tool is used to measure the raspberry resources while running the model; these measures are applied to (CPU usage, memory usage, bandwidth read/write rates, raspberry load system, disk utilization and queue, disk read/write rates, disk space usage) resources and the process of Zabbix setup can be explained as:

- install the Zabbix repository on the raspberry pi.
- install the Zabbix server, Zabbix browser, and Zabbix Client when the package has been installed.
- create the zabbix database and importing the initial zabbix schema.
- starting the zabbix Server.
- accessing zabbix Server.
- Import the first scheme.

The network was used as a local network, which means that using the same internet with specifying the IP address for raspberry mac address through forwarding it in the router and enabling Simple Network Management Protocol (SNMP) in PC. This will measure the resources through

Zabbix server on raspberry, by reaching the IP address specified for it then run the Zabbix server to measure resources usage and rates while running the model raspberry.

3.7. Docker Resources Measures

Docker is a container-based application development, deployment, and execution tool. The program is technologically advance since it streamlines overall application development while consuming less resources. Docker containers are Lightweight Virtualization Mechanism in comparison to virtualization technologies.

Docker Raspberry Pi is a container-based development, deployment, plus execution platform. When tried to compare to virtual machines, it would be the most lightweight yet powerful. Inside the Raspberry Pi technology differs from other competing systems and it being revolutionary.

3.8. Docker Image

The The Docker image is indeed a package that a Docker container uses to run programs. Docker images, like a pattern, serve as a collection of rules for constructing a Docker container. Whenever utilizing a Container, Docker images also form a basis. Within virtual machine systems, an image seems similar to something like a snapshot.

4. RESULT AND DISCUSSION

This This section will finalize the work of thesis, where the different results and discussion will be presented. So, it will be presented the way of loading the model into Raspberry pi, which is denoted as client of image in Docker container; the way that Google Cloud will get the image to measure it, in other way the model classifies the video from camera and live through Raspberry as violence and non-violence. It will be also presented the video resources used as local video recorded, which also identifies the video as violence or nonviolence. Results retrieved from Zabbix will be also presented, and used to measure components of PC, for instance disk, Memory, CPU, and network bandwidth.

For the purpose of this thesis, pretrained neural networks were used to conduct an analysis of the difficulty of recognizing instances of violence within a large number of datasets consisting of recorded videos and livestreams. This research was conducted with the intention of determining how accurate LSTM, BILSTM, and GRU models are. The results are presented in the table located at 4.1. However, the BiLSTM architecture higher level of efficiency in the UCF-Crime dataset, and the GRU architecture higher level of efficiency in personal movies. The LSTM model demonstrates more precision in the RWF2000 dataset than in other datasets (UCF-Crime, Personal films), but the LSTM model demonstrates more precision in the RWF2000 dataset than in other datasets (UCF-Crime, Personal films). The LSTM model's performance is, on the whole, significantly higher than that of its rivals (BiLSTM and GRU). In Table 4.1 it is shown the accuracy, where for dataset RWF2000 is 0.835 with LSTM model, which is better than the BiLSTM and GRU, while LSTM show a lower accuracy in UCF-Crime dataset with 0.810 and Personal Videos with 0.805. BiLSTM model reaches a better result with personal videos datasets, with 0.780, when compared with the other two datasets, showing values of 0.761 and 0.750 for RWF2000 UCF-Crime datasets, respectively. Last, GRU model shows the best performance with UCF-Crime with an accuracy value of 0.730, while in other datasets it showed a value of 0.705 and 0.710 for RWF2000 personal videos, respectively. Overall LSTM show best result accuracy in all datasets as described above and as table shows result below.

Table 4.1. The comparison of the Pre-Trained Models results.

No.	Model	Dataset	Accuracy
1	**LSTM**	**RWF2000**	**0.835**
2	BiLSTM	RWF2000	0.761
3	GRU	RWF2000	0.705
4	LSTM	UCF-Crime	0.810
5	BiLSTM	UCF-Crime	0.750
6	GRU	UCF-Crime	0.730
7	LSTM	Personal Videos	0.805
8	BiLSTM	Personal Videos	0.780
9	GRU	Personal Videos	0.710

A learning curve is a graph of models learning efficiency, as time-dependent or knowledge-dependent. In deep learning architectures, learning curves are a common assessment measure for methods that learn progressively from data training. After every iteration throughout training, the models could be tested on the training dataset and a hang validating dataset, and graphs of the evaluation metrics are being constructed to display learning curves. As shown in Figure 4.1. the Learning Curve of the proposed model.

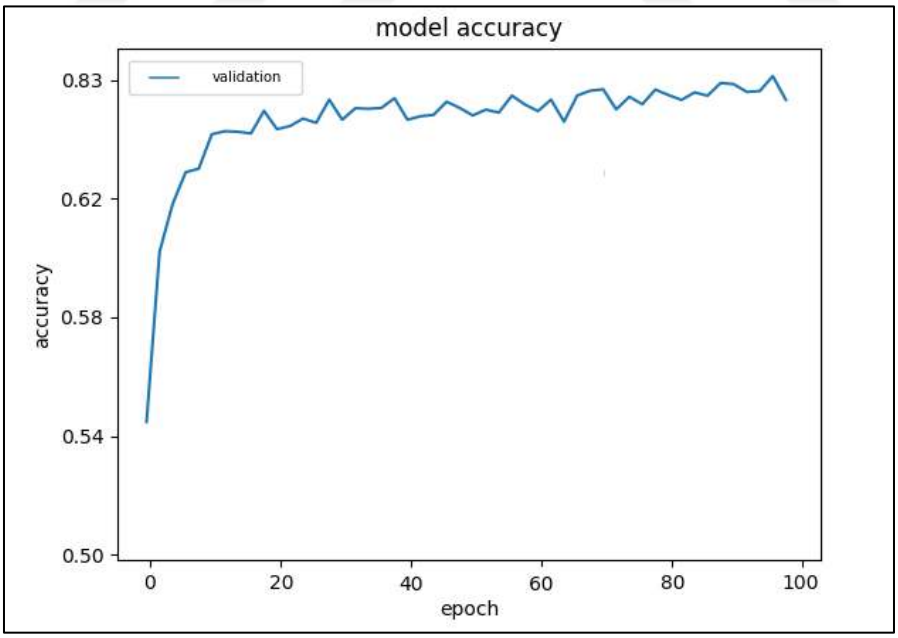

Figure 4.1. The Learning Curve of the model.

To reduce deep learning models' performance error, neural networks employ optimization algorithms such as stochastic gradient descent (SGD). A Loss Formula is used to determine this

error, being used to assess how well or poorly the algorithm is performing. the Loss Function of model is shown below in Figure 4.2.

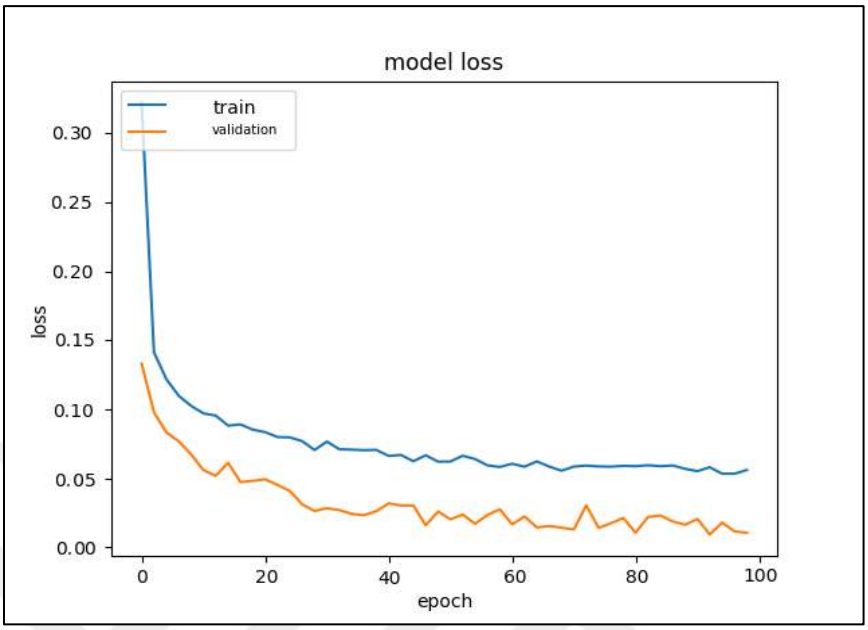

Figure 4.2. The Loss Function of model.

4.1. Result of Zabbix measures

The results of Zabbix server used as measurement of raspberry pi resources (CPU, RAM, DISK) while the pre-testing model is running in period of time from 17:41 pm to 18:01 pm and can be describes as follow:

4.1.1. CPU USAGE

Everything that occurs on the server is a system activity, and each activity is broken down into tasks which the server runs. Procedures vary in complexity and the speed which they can be accomplished. The amount of period which the CPU is used to execute its activities is referred to as CPU utilization. The CPU usage measured in Zabbix server when the model run in period time from 17:55 until 18:01 as Figure 4.3 below show up.

The CPU performance shows that, while the proposed model is running to determine which detection of violence or action happened in live stream or videos and CPU usage can be explained as follow; the CPU performance minimum start with 0% then reach 57.7214% with maximum usage of Raspberry CPU while the average rate is 20.4408% of CPU Raspberry usage.

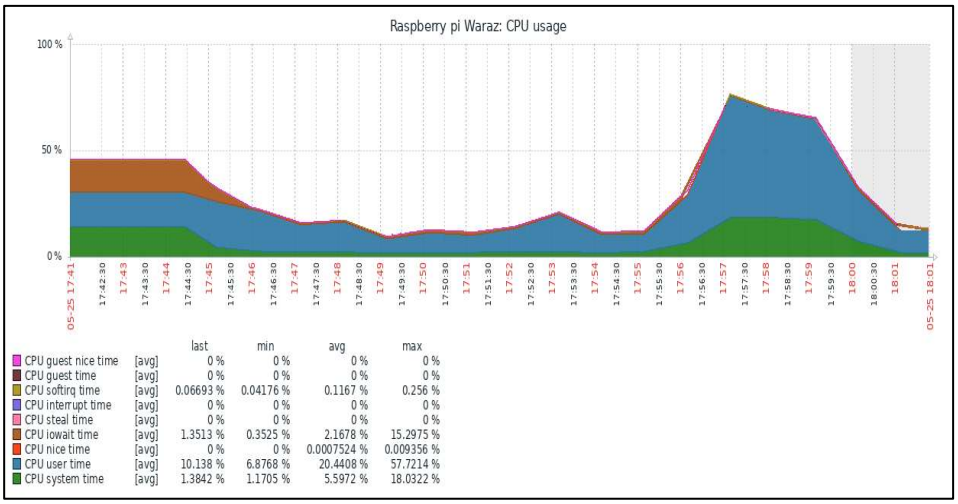

Figure 4.3. CPU usage.

4.1.2. SYSTEM LOAD

System load is a measure for CPU over or underutilization; i.e., the list of activities running on the CPU or even in the pending state. The system load of raspberry measure in Zabbix server as shown in Figure 4.4. The system with 1-minute average shows 0.47 for minimum one, 4.18 as maximum, and 1.835 for average one. In 5-minute average shows 0.7 as minimum one, 2.49 as maximum one, and 1.3317 as average. The 15-minute: 0.24 as minimum, 1.35 as maximum, and an of 0.7289. as figure 4.4 shows below.

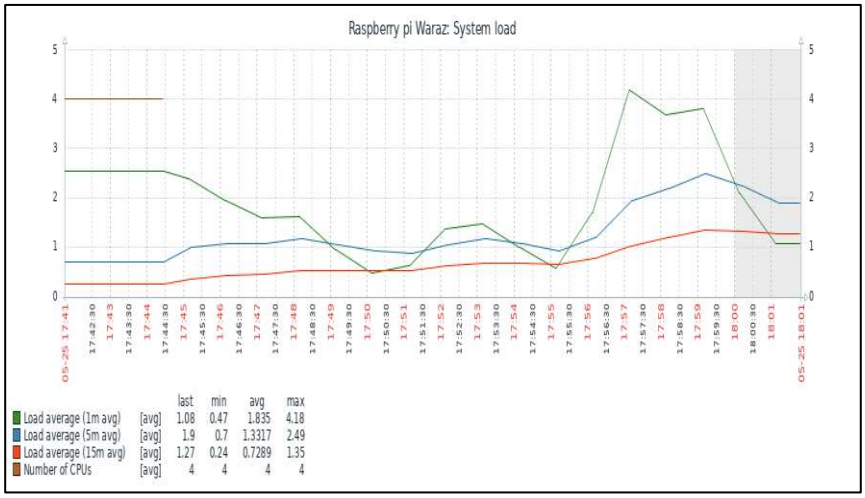

Figure 4.4. System load.

4.1.3. Disk read/write rates

Read/write speed are metrics to assess the efficiency of data storage. The reading rate indicates what amount of time it is required to open a file on a medium, whereas the writing rate

indicates how time it takes to store that to a storage medium. The disk of raspberry measure in Zabbix server as shown in Figure 4.5. The disk read rates for minimum, maximum and average is as follows 0 r/s, 20.5778 r/s and 2.0016 r/s, respectively; while the disk write rates is as follows 0.1267 w/s, 27.4142 w/s and 21.3587 w/s for the minimum, maximum and average rates, respectively.

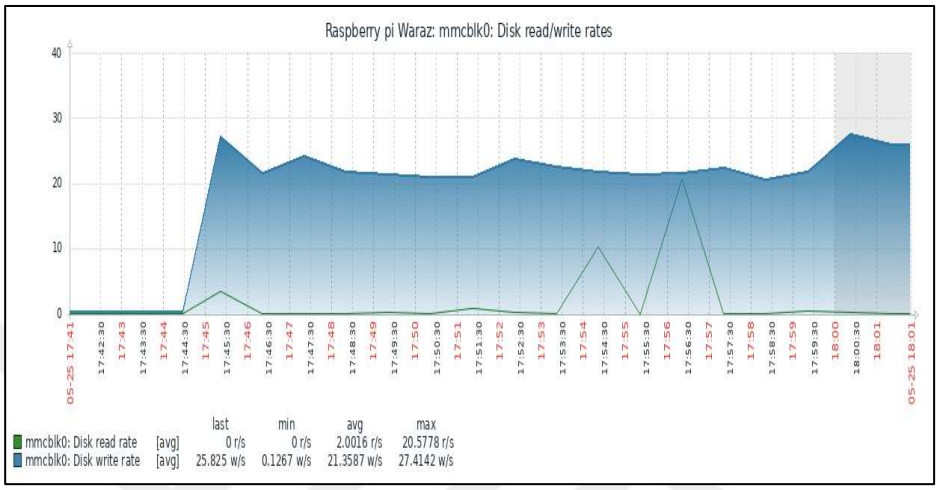

Figure 4.5. Disk read/write rates.

4.1.4. Memory Usage

The Memory Usage screen shows users how much memory is accessible on a particular device and also how much memory is presently being used by all apps, including Windows. Memory capacity measured in Zabbix server with range for total memory is 3.7 GB for minimum, 3.7 GB for maximum and 3.7 GB for average one. Regarding available memory, minimum one shows 3.14 GB, maximum is 3.16 GB while in average is 3.15 GB as figure 4.6 below shows.

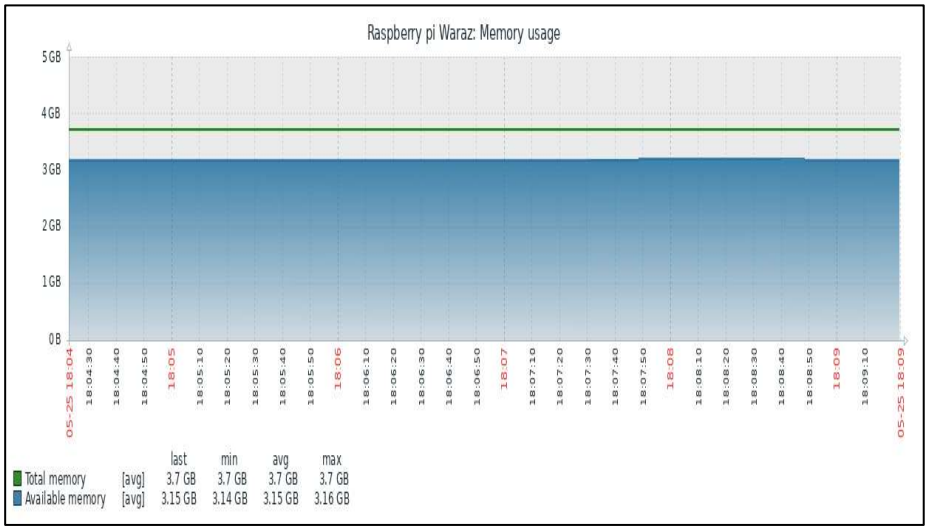

Figure 4.6. Memory usage.

4.1.5. Disk Utilization

The disk space utilization in the raspberry refers to that same amount of storage media that is actively being used, measured as a proportion. It differs from storage space or capability, which indicates the extent quantity of data that even a disk can store. Disk utilization is shown in Figure 4.7 below, showing a value retrieved from Zabbix server of 116.46 GB for a conventional system, while 80.87 GB for the proposed system.

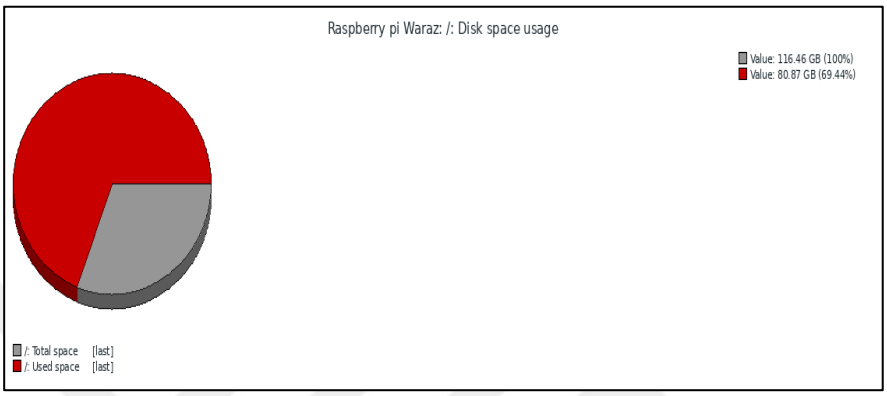

Figure 4.7. Disk Utilization.

4.2. Result of google cloud measures

The results of the graphics and analytics will be displayed here with a short explanation. As shown in Figure 4.8., the files of the output have been saved in Google cloud.

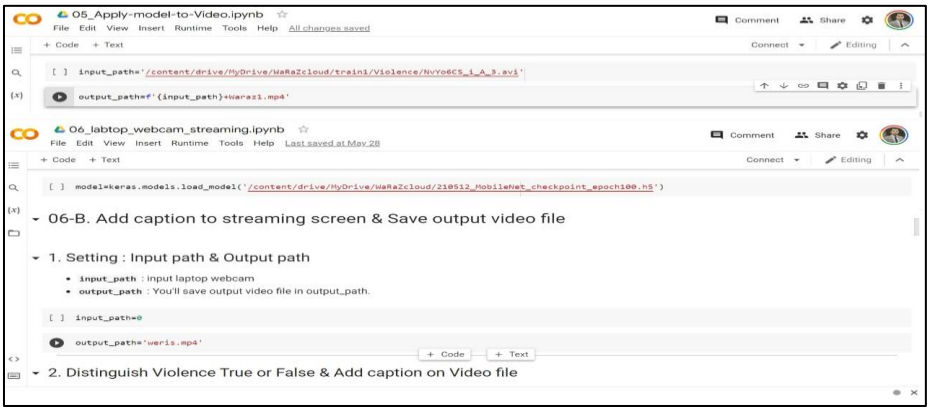

Figure 4.8. The output files in the google cloud.

Some examples of our thesis violence and non-violence situations, (a) The video that was recorded in a public place in Duhok city, and the output is violence in the detection process, (b) The second video was recorded in a public place that was got from a RWF-2000 dataset, and the

output is violence in the detection process, (c)The third video was recorded in a public place in Erbil city that was retrieved from a CCTV home camera, and the output is non-violence in the detection process, the (d) video was recorded in the author's house, by using a raspberry pi camera to test our system, and the output is violence in the detection process. At the end those results will have notified by Docker container, as shown in the Figure 4.9.

Figure 4.9. Livestream outputs labeled.

Some examples of thesis on violence and non-violence situations, (a) This video was recorded on the night of a football match between Duhok Club and Zakho Club during violence between the fans of the two clubs, and the output is violence in the detection process, (b) This video was recorded during a football match in Duhok governorate during crowds among the fans during the game, and the violence was detected during the crowds, and the output is violence in the detection process, (C) The third video was recorded in a public place at night that was acquired from a CCTV home camera, and the output is non-violence in the detection process, the (d) video was recorded in a house by using a raspberry pi camera to test the proposed system, and the output is non-violence in the detection process. This is shown below in the Figure 4.10.

Figure 4.10. Livestream outputs labeled.

4.3. Result of Docker cloud measures

The video we captured in the house using a raspberry pi camera to test the thesis proposed system is included below as just an instance of the thesis on scenarios of violence and non-violence. The result is violence in the identification procedure. The identified violent video output, which has been classified by the model being used for surveillance, was sent to the cloud through the Docker container, as shown in Figure 4.11.

Figure 4.11. Livestream outputs labeled.

5. CONCLUSION

In the thesis, a well-known CNN, together with a LSTM used to identify violence in videos. Converting picture frames to RGB format and image restoration of the video frame to the size (160x160x3) are used to pre-process videos. The test dataset picture frames are supplied to the convolutional neural network, which employs the MobileNet architecture with global average pooling. The characteristics collected by CNN are used to learn from input data, and the extracted features are sent into the LSTM, which determines the video sequence's frames. For the RWF2000 dataset, the LSTM model perform preferable than any of the other datasets (UCF-Crime, Personal Videos), whereas the BiLSTM model performs better than the GRU model in the Personal Videos dataset. Overall, the LSTM model does preferable than the other models (BiLSTM and GRU), with an accuracy of 0.83 percent.

To train and merge the LSTM into this thesis, we employed the CNN (MobileNet) model, where data is standardized and supplemented for improved performance training. The best categorization accuracy was recorded by the Violence and Non-Violence Detection System, which achieve about 83 %. Also put our system through real-time prediction testing, which it passed with flying colours. By choosing a time window on the online or mobile user interface, you can use the reported events from the database to keep an eye on the area for safety and other reasons.

The findings of this study introduce a unique computational framework that can detect violent behaviour in a variety of settings. The capability of the approach to capture pedestrian dynamics based on the spatial-temporal properties of multiple derivatives received a lot of focus throughout development of the methodology. This study brought to light the significance of information in demonstrating the intricate dynamics of pedestrian behaviour in densely populated areas. According to the findings of the study, the performance of classifiers is significantly improved when spatial and temporal motion patterns are combined. Last but not least, the example gives an illustration of how the descriptor may be used, not just in a variety of violent scenarios, but also in the context of panic. It is offered a fresh collection of data, as well as a way for determining aggressive behaviour from surveillance video. The RWF-2000 collection is the biggest dataset of security camera film that has ever been assembled for the purpose of identifying instances of real-world violence. Additionally, movement of the camera offers a novel pooling strategy that can be used to give time-based features as opposed to live objects. This may be accomplished by using the movement of the camera, which demonstrated to be an excellent way for automatically locating instances of aggressive behaviour captured on webcams.

While past strategies concentrated solely on the shot standard of a media model, it was decided to create a more conceptual image system in this methodology. As a consequence, more video’s dynamical information may be employed for processing. MobileNet and LSTM, working

together, are able to discern between violent and peaceful scenarios based on the multiple factors that are collected through the scenario design. After that, the motion data is merged, in order to assess whether or not the action sequence contains violent material. According to the findings of the trials, the technique that was offered is successful in recognizing a wide variety of violent occurrences. When assessed against the conclusions of earlier investigations, this study approach looks to have promising results. Since it's hard to define exactly what violence is, it would be difficult to identify any in the worst-case situation.

REFERENCES

[1] Baba, M., Gui, V., Cernazanu, C., & Pescaru, D. (2019). A sensor network approach for violence detection in smart cities using deep learning. Sensors, 19(7), 1676.

[2] Ramzan, M., Abid, A., Khan, H. U., Awan, S. M., Ismail, A., Ahmed, M., ... & Mahmood, A. (2019). A review on state-of-the-art violence detection techniques. IEEE Access, 7, 107560-107575.

[3] Zhang, T., Yang, Z., Jia, W., Yang, B., Yang, J., & He, X. (2016). A new method for violence detection in surveillance scenes. Multimedia Tools and Applications, 75(12), 7327-7349.

[4] Li, J., Jiang, X., Sun, T., & Xu, K. (2019, September). Efficient violence detection using 3d convolutional neural networks. In 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-8). IEEE.

[5] Zhou, P., Ding, Q., Luo, H., & Hou, X. (2017, June). Violent interaction detection in video based on deep learning. In Journal of physics: conference series (Vol. 844, No. 1, p. 012044). IOP Publishing.

[6] Sumon, S. A., Goni, R., Hashem, N. B., Shahria, T., & Rahman, R. M. (2020). Violence detection by pretrained modules with different deep learning approaches. Vietnam Journal of Computer Science, 7(01), 19-40.

[7] Ye, L., Ferdinando, H., Seppänen, T., & Alasaarela, E. (2014). Physical violence detection for preventing school bullying. Advances in Artificial Intelligence, 2014(2014), 1-9.

[8] F Ullah, F. U. M., Ullah, A., Muhammad, K., Haq, I. U., & Baik, S. W. (2019). Violence detection using spatiotemporal features with 3D convolutional neural network. Sensors, 19(11), 2472.

[9] Accattoli, S., Sernani, P., Falcionelli, N., Mekuria, D. N., & Dragoni, A. F. (2020). Violence detection in videos by combining 3D convolutional neural networks and support vector machines. Applied Artificial Intelligence, 34(4), 329-344.

[10] Zhou, P., Ding, Q., Luo, H., & Hou, X. (2017, June). Violent interaction detection in video based on deep learning. In Journal of physics: conference series (Vol. 844, No. 1, p. 012044). IOP Publishing

[11] Yilmaz, A., & Shah, M. (2005, June). Actions sketch: A novel action representation. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 1, pp. 984-989). IEEE.

[12] Ammar, S. M., Anjum, M., Rounak, T., Islam, M., & Islam, T. (2019). Using deep learning algorithms to detect violent activities (Doctoral dissertation, BRAC University).

[13] Nam, J., Alghoniemy, M., & Tewfik, A. H. (1998, October). Audio-visual content-based violent scene characterization. In Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269) (Vol. 1, pp. 353-357). IEEE.

[14] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

[15] LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In Proceedings of 2010 IEEE international symposium on circuits and systems (pp. 253-256). IEEE.

[16] Xu, X., Liu, W. Q., & Li, L. (2014). Low resolution face recognition in surveillance systems. Journal of Computer and Communications, 2, 70-77.

[17] Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S., & Mustaqim, M. (2020). Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. Ieee Access, 8, 23022-23040.

[18] Rota, P., Conci, N., Sebe, N., & Rehg, J. M. (2015, September). Real-life violent social interaction detection. In 2015 IEEE international conference on image processing (ICIP) (pp. 3456-3460). IEEE.

[19] Serrano-Gracia, I. (2017). Contributions to the Problem of Fight Detection in Video. ELCVIA: electronic letters on computer vision and image analysis, 16(2), 0033-36.

[20] Senst, T., Eiselein, V., & Sikora, T. (2015, July). A local feature based on Lagrangian measures for violent video classification. In 6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15) (pp. 1-6). IET.

[21] Chen, L. H., Su, C. W., & Hsu, H. W. (2011). Violent scene detection in movies. International Journal of Pattern Recognition and Artificial Intelligence, 25(08), 1161-1172.

[22] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Gool, L. V. (2016, October). Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision (pp. 20-36). Springer, Cham.

[23] Zhang, T., Yang, Z., Jia, W., Yang, B., Yang, J., & He, X. (2016). A new method for violence detection in surveillance scenes. Multimedia Tools and Applications, 75(12), 7327-7349.

[24] Deniz, O., Serrano, I., Bueno, G., & Kim, T. K. (2014, January). Fast violence detection in video. In 2014 international conference on computer vision theory and applications (VISAPP) (Vol. 2, pp. 478-485). IEEE.

[25] Bermejo Nievas, E., Deniz Suarez, O., Bueno García, G., & Sukthankar, R. (2011, August). Violence detection in video using computer vision techniques. In International conference on Computer analysis of images and patterns (pp. 332-339). Springer, Berlin, Heidelberg.

[26] Tamilarasi, F. C., et al. (2020, June). convolutional neural network based autism classification. in 2020 5th international conference on communication and electronics systems (ICCES) (pp. 1208-1212). IEEE.

[27] Aghdam, M. A., et al. (2018). Combination of rs-fMRI and sMRI data to discriminate autism spectrum disorders in young children using deep belief network. J Digit Imaging 31, 895–903.

[28] Udayakumar, N. (2016). Facial expression recognition system for autistic children in virtual reality environment. Int. J. Sci. Res. Publ, 6(6), 613-622.

[29] Salman A. D., et al. (2020). ASD classification based on machine learning techniques. journal of xi'an university of architecture & technology, XII(IV).

[30] Le, H. S., Oparin, I., Allauzen, A., Gauvain, J. L., & Yvon, F. (2011, May). Structured output layer neural network language model. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5524-5527). IEEE.

[31] Rani, P. (2019, November). Emotion detection of autistic children using image processing. In 2019 Fifth International Conference on Image Information Processing (ICIIP) (pp. 532-535). IEEE.

[32] Bisong, E. (2019). Kubeflow and kubeflow pipelines. In Building Machine Learning and Deep Learning Models on Google Cloud Platform (pp. 671-685). Apress, Berkeley, CA.

[33] Fu, R., Zhang, Z., & Li, L. (2016, November). Using LSTM and GRU neural network methods for traffic flow prediction. In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 324-328). IEEE.

[34] Cornegruta, S., Bakewell, R., Withey, S., & Montana, G. (2016). Modelling radiological language with bidirectional long short-term memory networks. arXiv preprint arXiv:1609.08409.

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

[36] Li, S., Xu, L. D., & Zhao, S. (2015). The internet of things: a survey. Information systems frontiers, 17(2), 243-259.

[37] Cheng, M., Cai, K., & Li, M. (2021, January). RWF-2000: an open large scale video database for violence detection. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 4183-4190). IEEE.

[38] Ullah, W., Ullah, A., Hussain, T., Khan, Z. A., & Baik, S. W. (2021). An efficient anomaly recognition framework using an attention residual LSTM in surveillance videos. Sensors, 21(8), 2811.

[39] Huitron, V., León-Borges, J. A., Rodriguez-Mata, A. E., Amabilis-Sosa, L. E., Ramírez-Pereda, B., & Rodriguez, H. (2021). Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. Computers and Electronics in Agriculture, 181, 105951.

[40] Coburn, J. (2020). Raspberry Pi History. In Build Your Own Car Dashboard with a Raspberry Pi (pp. 1-12). Apress, Berkeley, CA.

[41] Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access, 6, 61677-61685.

[42] Raj, P., Chelladhurai, J. S., & Singh, V. (2015). Learning Docker. Packt Publishing Ltd.

[43] Lambacing, M. M. M., Apriliani, R., & Sakti, D. V. S. Y. (2020). Rancang Bangun Sistem Manajemen Jaringan dan Suhu untuk Data Center menggunakan Raspberry Pi dan Zabbix. Prosiding SISFOTEK, 4(1), 151-155.

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

CURRICULUM VITAE

Waraz Mustafa Sadeeq ALDUHOKI

[redacted]

[redacted]

[redacted]