

# **Vision-Based Violence Detection Through Deep Learning**

**KOH WEI ZHE**

**UNIVERSITI TUNKU ABDUL RAHMAN**

# **Vision-Based Violence Detection Through Deep Learning**

**KOH WEI ZHE**


**A project report submitted in partial fulfilment of the  
requirements for the award of Bachelor of Science  
(Honours) Software Engineering**

**Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman**

**September 2024**

## DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :   
\_\_\_\_\_

Name : Koh Wei Zhe  
\_\_\_\_\_

ID No. : 2004757  
\_\_\_\_\_

Date : 12 September 2024  
\_\_\_\_\_

**APPROVAL FOR SUBMISSION**

I certify that this project report entitled “**Vision-Based Violence Detection Through Deep Learning**” was prepared by **KOH WEI ZHE** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Software Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature :   
\_\_\_\_\_

Supervisor : Chua Sing Yee  
\_\_\_\_\_

Date : 13 September 2024  
\_\_\_\_\_

Signature : *Lau*  
\_\_\_\_\_

Co-Supervisor : Lau Kean Hong  
\_\_\_\_\_

Date : 13 September 2024  
\_\_\_\_\_

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2024, Koh Wei Zhe. All right reserved.

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. Chua Sing Yee, my supervisor, for her invaluable guidance, support, and encouragement throughout this Final Year Project research. Her expertise and insights were instrumental in shaping the direction of this project, and her constructive feedback significantly contributed to the quality of the work.

I also extend my heartfelt thanks to my family and friends for their unwavering support and understanding during this challenging journey. Their encouragement kept me motivated and focused.

My sincere appreciation goes to the lecturers at Universiti Tunku Abdul Rahman (UTAR) for imparting the knowledge and skills that were crucial in accomplishing this research. Additionally, I am grateful to the researchers whose published papers provided essential information and insights that informed and enriched this study.

Lastly, I would like to thank the Lee Kong Chian Faculty of Engineering and Science (LKC FES) and the Department of Computing at UTAR for providing the resources and a conducive environment for conducting this research. This project would not have been possible without the collective support of everyone involved.

## ABSTRACT

In the present society, video surveillance systems have continued to develop and incorporate more sophisticated video analysis to enhance security and public safety. With increasing demand, the need for accurate and efficient violence detection in video footage has become more critical. However, detecting violence in video footage remains challenging due to varying lighting conditions and data quality. While advancements in deep learning techniques can improve the accuracy and robustness of violence detection, they often require extensive datasets, leading to overloaded training processes. This research focuses on advancing and utilizing deep learning models for violence detection in surveillance videos, with particular emphasis on varying lighting conditions. A dataset of 2,000 videos mostly in normal lighting conditions is used to train a hybrid deep learning model combining MobileNet-v2, a lightweight Convolutional Neural Network (CNN), with BiLSTM (Bidirectional Long Short-Term Memory). This hybrid model seeks to employ MobileNet-v2 for feature extraction and BiLSTM for temporal analysis in video datasets. To enhance detection accuracy under different lighting conditions, histogram equalization is integrated into the video prediction process alongside the trained base model. The approach is designed to optimize video-based violence detection without overwhelming the model with large datasets and excessive training times. The base model (MobileNet-v2 and BiLSTM) performed well in normal light conditions (96.33%). While the base model with histogram equalization achieved higher accuracy (98.91%) and the model trained on varying lighting conditions further improved to (99.15%). On the other hand, the base model performed poorly in very dark conditions (24.89%) but showed significant improvement with histogram equalization (92.21%), nearly matching the performance of the base model trained on varying lighting conditions (99.97%). This result highlights the benefit of the proposed histogram equalization method, which achieves high detection accuracy without relying on extensive datasets and overloaded training resources, making it a potential solution for real-time violence detection in diverse lighting scenarios.

## TABLE OF CONTENTS

<b>DECLARATION</b>		<b>i</b>
<b>APPROVAL FOR SUBMISSION</b>		<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>		<b>iv</b>
<b>ABSTRACT</b>		<b>v</b>
<b>TABLE OF CONTENTS</b>		<b>vi</b>
<b>LIST OF TABLES</b>		<b>ix</b>
<b>LIST OF FIGURES</b>		<b>x</b>
<b>LIST OF SYMBOLS / ABBREVIATIONS</b>		<b>xii</b>
<b>LIST OF APPENDICES</b>		<b>xiii</b>
<b>CHAPTER</b>		
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	General Introduction	1
1.2	Problem Statement	2
1.2.1	Rise in Violence Incidents in Malaysia	2
1.2.2	Challenges in Acquiring Appropriate Datasets	2
1.2.3	Challenges in Training Large and Complex Datasets	3
1.3	Aim and Objectives	4
1.4	Research Questions	4
1.5	Research Hypotheses	4
1.6	Research Activities	4
1.7	Scope and Limitation of the Study	5
1.8	Contribution of the Study	6
1.9	Outline of the Report	6
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>7</b>
2.1	Introduction	7
2.2	Dataset Preparation	7



2.3	Deep Learning	8
2.3.1	Overview of Deep Learning	9
2.3.2	Convolutional Neural Network (CNN)	11
2.3.3	Recurrent Neural Network Architecture (RNN)	19
2.3.4	Long Short-Term Memory (LSTM)	21
2.3.5	Hybrid CNN + LSTM Architecture	23
2.4	Summary	26
<b>3</b>	<b>METHODOLOGY AND WORK PLAN</b>	<b>28</b>
3.1	Introduction	28
3.2	Experimental Setup	28
3.3	Workflow plan	29
3.3.1	Data Collection	30
3.3.2	Data Preprocessing	31
3.3.3	Data Splitting	31
3.3.4	Fine Tuning	32
3.3.5	Model Architecture	32
3.3.6	Model Training	33
3.3.7	Model Evaluation	33
3.3.8	Model Testing	33
3.3.9	Histogram Equalization	36
3.3.10	Video Prediction	36
3.4	Violence Detection Models	38
3.4.1	Base Model	38
3.4.2	Model Trained on Videos with Varying Lighting Conditions	38
3.4.3	Base Model with Histogram Equalization Enhancement	38
3.5	Gantt Chart	38
3.6	Summary	40
<b>4</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>41</b>
4.1	Introduction	41
4.2	Performance Evaluation of the Base Model: MobileNet-v2 with BiLSTM	41

4.2.1	Confusion Matrix of Base Model	42
4.2.2	Evaluation of the Receiver Operating Characteristic (ROC) Curve for the Base Model	43
4.2.3	Example Results of Base Model Prediction	43
4.3	Comparison of Violence Detection Models Across Different Lighting Conditions	46
4.4	Analysis of the Proposed Flow of Base Model with Histogram Equalization	47
4.4.1	Performance of Different Video Frame Rates	47
4.4.2	Processing Time of Histogram Equalization Across Different Video Lengths	49
4.4.3	Performance Improvement Through Reduction of Video Resolution	50
<b>5</b>	<b>CONCLUSIONS AND RECOMMENDATIONS</b>	<b>53</b>
5.1	Conclusions	53
5.2	Recommendations for future work	54
	<b>REFERENCES</b>	<b>55</b>
	<b>APPENDICES</b>	<b>59</b>

**LIST OF TABLES**

Table 2.1: Comparison of different deep learning models.	26
Table 3.1: Hardware and Software Specifications.	28
Table 3.2 Summary of Dataset Used	30
Table 4.1 Evaluation Metrics for Base Model (MobileNet-v2 with BiLSTM)	41

## LIST OF FIGURES

Figure 2.1: ReLU activation function graph.	10
Figure 2.2: Softmax activation function graph.	11
Figure 2.3: LeNet-5 network.	12
Figure 2.4: Convolutional Layer.	13
Figure 2.5: Max pooling.	14
Figure 2.6: Fully connected layer.	16
Figure 2.7: SPIL overview model pipeline.	17
Figure 2.8: The squeezed module.	18
Figure 2.9: RNN architecture.	20
Figure 2.10: BiLSTM architecture.	22
Figure 2.11: VGG19 with LSTM model.	24
Figure 2.12: CNN-BiLSTM model.	25
Figure 3.1: Flowchart for the Entire Workflow.	29
Figure 3.2: ROC and AUC graph	35
Figure 4.1 Confusion Matrix of Base Model	42
Figure 4.2 ROC Curve Graph	43
Figure 4.3 Prediction as Frame by Frame in Normal Condition	44
Figure 4.4 Prediction as Entire Video Sequence in Normal Condition	44
Figure 4.5 Prediction as Frame by Frame in Dark Condition	45
Figure 4.6 Prediction as Entire Video Sequence in Dark Condition	45
Figure 4.7 Prediction as Frame by Frame in Very Dark Condition	45
Figure 4.8 Prediction as Entire Video Sequence in Very Dark Condition	45
Figure 4.9 Accuracy Comparison Across Three Models Across Different Lighting Conditions	46

Figure 4.10 Accuracy Across Different Video Frame Rates	48
Figure 4.11 Processing Time vs. Video Length for Video 1 (Very Dark)	49
Figure 4.12 Processing Time vs. Video Length for Video 2 (Dark)	50
Figure 4.13 Histogram Equalization Processing Time for One-Second Videos under Different Lighting Conditions	51
Figure 4.14 Comparison of Accuracy and Histogram Equalization Processing Time with and without Reduced Video Resolution	51

## LIST OF SYMBOLS / ABBREVIATIONS

$f$	Softmax
$x_i$	Input vector
$e^{x_i}$	Standard exponential function for input vector
$K$	Number of classes in the multi-class classifier
$e^{x_j}$	Standard exponential function for output vector
1D	1 Dimensional
2D	2 Dimensional
3D	3 Dimensional
AI	Artificial Intelligence
AUC	Area Under the Curve
BiLSTM	Bi-directional Long Short-Term Memory
BiRNN	Bi-directional Recurrent Neural Network
CEC	Constant Error Carousel
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
FPS	Frame Per Second
I3D	Inflated 3D ConvNet
IoT	Internet of Things
LSTM	Long Short-Term Memory
R(2+1)	ResNet (2+1)D
ReLU	Rectifier Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RWF	Real World Footage
SPIL	Skeleton Points Interaction Learning
STCNN	Spatio-temporal Convolutional Network

**LIST OF APPENDICES**

Appendix A: Accuracy Comparison Across Three Models for Videos Categorized into Four Categories Tables	59
---	----

## CHAPTER 1

### INTRODUCTION

#### 1.1 General Introduction

Security and social peace form necessary conditions in a modern society, thus strong mechanisms for monitoring, and taking actions are needed to prevent conflict situations. The classic methods of surveillance such as human observation carried out by security personnel and closed-circuit television (CCTV) system types, have been used to detect and curb incidences of violence and criminal tendencies. Nevertheless, these methods have some inherent constraints, which might stand as a barrier in case of ensuring public safety.

Nowadays, with the exponential growth in the amount of video data, surveillance and anomaly detection have become increasingly crucial. (Feng et al., 2021). Hence, surveillance techniques are crucial to effectively respond to the security problems. Therefore, the challenge nowadays is further compounded especially because surveillance systems deal with a growing amount and complexity of footage. A large set of data which is required by deep learning models results in longer training time and requires more computational power. These challenges, therefore, underscore the need for enhanced approaches that would address issues such as large data handling and the detection of better accuracies.

Thus, this research study seeks to overcome the above challenges by proposing a novel automatic violence detection system based on deep learning methodology. To improve violence detection under changing lighting conditions, this project aims to increase the efficiency and effectiveness of identifying violent scenes using hybrid models i.e. MobileNet-v2 with Bidirectional Long Short-Term Memory (BiLSTM) networks, The use of advanced image processing techniques in histogram equalization in the vision pipeline is adopted. The ultimate goal is to enhance the reliability of surveillance systems and to define violent actions in real life circumstances and thus make a contribution to improving security and reducing crime.



## **1.2 Problem Statement**

This section discusses the problem statement of this project.

### **1.2.1 Rise in Violence Incidents in Malaysia**

With the increasing occurrence of violent episodes across the country, including thefts, domestic abuse and bullying, the present methods for identifying and responding to such crimes are becoming insufficient. Conventional systems increase the hazards to public safety since they primarily rely on human interaction, which will cause delays in detection and reaction. Furthermore, recent reports reveal a concerning surge in crime rates in Selangor, with 13,740 cases reported in 2023, indicating a 5.84% increase from the previous year (Khandelwal, 2024). There is a noticeable sense of anxiety among the public, as seen by this statistical increase. While property-related offenses have increased by 5.8%, violent crimes have increased by 5.92% (Khandelwal, 2024).

However, even with concentrated efforts to address violence in Malaysia, the existing mechanisms are unable to stop the increasing threat of violent occurrences. The safety and well-being of residents are still in danger due to the ongoing threat of violence, even with the enforcement measures already in place.

Therefore, to overcome the limitation of manual monitoring and the increased number of violent incidents in Malaysia, a sophisticated deep-learning model will be constructed. Thus, the model will aim to enhance the surveillance camera capabilities by automatically detecting violent activity by every frame per second.

### **1.2.2 Challenges in Acquiring Appropriate Datasets**

One of the critical challenges in constructing reliable deep learning models for automated violence detection is the lack of suitable datasets that are representative of variability in real-world conditions. Most of the current datasets are small in quantity and lack variation in lighting, different perspectives of the camera, and resolution which is very essential when it comes to training models that are capable of operating effectively in different environments. Deep learning models highly depend on the quality and amount of training data (Alzubaidi et al., 2021a); however, when datasets are

insufficient, unbalanced, or contain errors, the models might tend to overfit or even underfit, the models learn these peculiarities and are not very good at predicting under different conditions (Prudhomme et al., 2023). This usually results in the creation of models that do very well on paper, and in lab conditions, but poorly when implemented in real-life scenarios such as the recognition of violence where there is poor lighting or background noise. The lack of an inclusive and diverse dataset also impacts the development of models that can identify violent incidents in a variety of environments with reasonable accuracy, thus limiting the feasibility and efficacy of automated violence identification schemes.

### **1.2.3 Challenges in Training Large and Complex Datasets**

One of the major concerns for implementing a deep learning model is the fact that the training duration escalates when training with large datasets. When these datasets become larger and varied, the time required to train the model adequately rises. These diverse datasets are required to teach the model to identify violence under different conditions, although training the program on millions of pieces of data is considerably time-consuming and requires significant computational power (Radiuk, 2017). However, if the performance of the model on the initial data is not satisfactory, then a retraining process is required, which also takes time and effort. The problem with retraining is that it entails rehearsing the model on the entire dataset every round, a process that consumes time and requires a lot of computing power. Such a drawn-out training and retraining process hamper the efficiency of the model by protracting the time taken to deploy it also increases the cost of development which is a major barrier to developing an ideal violence detection system.

### **1.3 Aim and Objectives**

This project aims to develop a deep learning-based solution for automated violence detection to enhance public safety.

1. To implement a deep learning-based model capable of accurately detecting violent incidents in video footage, even under varying lighting conditions.
2. To investigate and apply preprocessing techniques to enhance the model's effectiveness in detecting violent incidents.
3. To evaluate the performance of the implemented violence detection model in various lighting conditions.

### **1.4 Research Questions**

Based on the research objectives, there are two research questions for this study. The first research question is how accurately that is a deep learning model detects violent behaviour in video footage. Secondly, what effect do preprocessing techniques such as histogram equalization have on the deep learning model performance in detecting violent behaviour across varying light conditions?

### **1.5 Research Hypotheses**

Based on the research question, there are two research hypotheses for this study. The first research hypothesis is a deep learning model can accurately detect violent behaviour in video footage. Secondly, preprocessing such as histogram equalization can help improve the ability of deep learning models to detect violent behaviour in video footage under varying light conditions.

### **1.6 Research Activities**

Based on the research hypothesis, there are four research activities for this study. The first research activity is to conduct a literature review to identify and choose existing deep learning model architectures used in similar projects which suitable for detecting violent behaviour. Secondly, implement and train a deep learning model to detect violent behaviour in video footage. Thirdly, applied histogram equalization in the test video that is under the video prediction phase where the model is trained. Lastly, analyze and compare the model's accuracy

and reliability based on the base model with additional datasets, with and without histogram equalization.

### **1.7 Scope and Limitation of the Study**

This research project aims to investigate and develop deep learning-based models for violence detection using videos or surveillance footage videos, to reduce violence cases in Malaysia, and address the challenges of finding suitable datasets that reflect the diversity of real conditions and the impact of dataset size on the training time. Histogram equalization is performed on test data to improve the models gathered in this context as a significant part of this work. The project focuses on implementing a deep learning model, a hybrid MobileNet-v2 with BiLSTM model, to analyze the video data and identify the patterns to indicate violent behavior. This model incorporates histogram equalization and identifies the patterns of violent behavior. Therefore, the deep learning model will be experimented with to optimize the accuracy of the violence detection performance result, and also manage training time effectively in making the model suitable for practical deployment in real-world surveillance systems. Moreover, this research seeks to explore the model that will allow for future modification and integration into a real system. Hence, the integration of the trained models into the current surveillance is presented in the subsequent implementation phase.

Besides, this research has several limitations which must be acknowledged while undergoing the project. Firstly, this project is focused on improving violence detection under varying lighting conditions through the use of histogram equalization. Thus, the model may not be able to address other sources of noise in the video data, such as a damaged video with a glitch or extremely blurry. Moreover, the model training process will require high-quality and diverse training datasets containing examples of violent and non-violent behaviors. It will be challenging to obtain high-quality and diverse datasets due to data availability and ethical considerations. Lastly, the datasets may not cover all possible violent actions. Thus, it will limit the deep learning model's ability to generalize to unseen or uncommon violent action which will affect the violence detection accuracy in real-world situations.

## **1.8 Contribution of the Study**

The research enhances the existing knowledge and practice in using deep learning models for video-based violence detection, especially where the context is complex, such as during low light conditions. The contributions include a proposed approach that improves deep learning model accuracy utilizing histogram equalization during the prediction stage and the investigation of the effects of training datasets on the model accuracy and time. These provide important information that will be useful for researchers and practitioners, who want to enhance automated violence detection systems.

## **1.9 Outline of the Report**

This report is separated into 5 chapters as follows:

### Chapter 1: Introduction

This chapter provides an overview of the research project which includes an introduction, a problem statement, aim and objectives, a research question, research hypotheses, and the scope and limitations of the research.

### Chapter 2: Literature Review

This chapter summarises the study on many theses that have been put out by other researchers and are relevant to this undertaking.

### Chapter 3: Methodology

This chapter explains the details of the workflow plan and the procedure to carry out this research.

### Chapter 4: Results and Discussion

This chapter explains and discusses the findings of the research, including the performance evaluation of different models under various lighting conditions.

### Chapter 5: Conclusion

This chapter summarizes the important findings and recommendations for future work.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

Violence carries significant implications for public safety. Violence can occur in different forms such as public violence, family abuse, sports violence, and even school bullying cases. As violence continues to escalate across diverse domains, there is an urgent need for sophisticated technologies to facilitate the early recognition and prevention of these outbreaks of violence. In recent years, deep learning technology which is a part of artificial intelligence emerged as a potential approach to change the complex task of violence detection.

The literature presents the current state of the research conducted using deep learning techniques to detect violent behaviour in public areas. This research focuses on summarizing studies with datasets used, preprocessing methods, different activation functions, and deep learning architectures as well.

Hence, in this literature review, the main point is to research the basic concept of deep learning models to understand how they work. Moreover, reviews existing research emphasizing the performance and architecture of current models which gives more understanding to seek opportunities to push the boundaries of the field with a new and distinctive research contribution.

#### 2.2 Dataset Preparation

Before putting any video datasets into any deep learning models, preprocessing steps are carried out. Thus, a sequence of keyframes is extracted from video clips this way (Jahlan and Elrefaei, 2022). The process is deduced by separating the video frames and computing the absolute difference in every two consecutive frames. This difference frame is compared with the set threshold value. If the number of non-zero elements of the difference image is greater than the threshold value, the current frame is considered significantly different from the frame before and is included. After this, the image is resized to 224x224 pixels randomly by chopping a portion of the input image with this size.

Besides, another way for preprocessing the video is to extract the frame from the video datasets and then resize every frame of the video into a fixed

height and width. Then, the frames will then be loaded into a Numpy array, with each row indicating a sequence or pattern in the video (Halder and Chatterjee, 2020). Hence, this sequence can comprise many movements and activities such as punching, patting, shaking hands, etc.

Moreover, the video frames can be further processed by normalization to reduce the time to train in the deep learning model. This process is to ensure the input data for the deep learning model are on the same scale, so the model can process a lower range of data which greatly decreases the training time. Hence, video normalization is the process of normalizing all the frames in a video such that the RGB values of each pixel fall between 0 and 255 (Thakkar and Lo, 2023). Fundamentally, the aforementioned image normalization equation is applied to every frame of the video leading to video normalization. This process is meant to guarantee that the intensity readings are consistent throughout the video segments which makes it easier to standardize the processing and analysis of the video data.

### **2.3 Deep Learning**

Deep learning also lies within the scope of machine learning which itself is a subset of artificial intelligence. Artificial intelligence (AI) is one of the emerging areas that involves a wide array of techniques used to make computers behave like humans. Machine learning is about the algorithms being trained with the data in the process of enhancing this function. The particular branch of machine learning called Deep Learning essentially imitates the complex mechanism of the human brain.

In 2006, Hinton together with Salakhutdinov published the most influential article in the Science journal, which was the beginning of the deep learning era (Hinton and Salakhutdinov, 2006). The study logically confirmed the fundamental role of fully connected neural networks with hidden layers in improving feature learning skills. Such algorithms have great chances to enhance precision performance not only of cancer data but of many disparate data sources. Hence, this project mainly focuses on the deep learning model that can predict violence or nonviolence by image which is the frame extracted from the video.

### **2.3.1 Overview of Deep Learning**

The fundamental blueprint of the deep learning model, especially if used in the framework of CNNs, contains several layers that process data and make the necessary predictions. It always consists of convolutional layers (Conv2D layers) filters that scan and identify structures within the input data, e.g. images (Li et al., 2024). These layers have learnable filters known as kernels that move over the input to construct feature maps that detect features including edges, textures, and other complex patterns.

Next to the convolutional layers, there are often pooling layers that are used to decrease the spatial size of the feature maps but at the same time to keep the most significant values (Gichoya et al., 2022). Such steps are referred to as sub-sampling or down-sampling, which is important for shrinking the size of the model and reducing the risk of over-learning. The two fundamental types of pooling are the max pooling, which keeps the largest value of the patch of the feature map, and the average pooling which calculates the average of the patch. These operations make the model less sensitive to small translations in the input data.

The result of convolution and pooling works is the dimensionality reduction of the output feature maps which are then flattened into 1D vectors before being fed to one or more fully connected layers (Du et al., 2016). In these layers, all neurons in one layer are connected to all neurons in the previous layer and this enables the model to integrate the feature learned by each of the convolutional layers before making the final decision. In classification, this final fully connected layer uses the activation function to transform the output in the form of a probability ensemble for each of the classes, with the probability values ranging from zero up to one, with the maximum being the predicted class of the model.

Besides, there are different activation functions to determine the output of each neuron in a deep learning model.

#### **2.3.1.1 ReLU Activation Function**

ReLU is one of the activations functions that are commonly used even in the ImageNetILSVRC competition in the year 2012 (Krizhevsky et al., 2017). ReLU works by always flattening the input values below a certain threshold, which



are commonly set to zero, and then increasing linearly for any input value larger than this certain threshold. This means the ReLU will activate a node when the input value exceeds a certain threshold level which will exhibit a linear relationship with the dependent variable, whereas if the input value is below zero, the output value also becomes zero. Although it seems like the ReLU has a simple operation, the ReLU activation function introduces key nonlinear transformations.

Although ReLU has excellent performance, it also has unique limitations. As a result, the slope of ReLU often is either zero or a constant value, which may cause the phenomenon termed "dying ReLU", when the neurons stop updating during the training (Montesinos López et al., 2022). However, experimental data shows that ReLU activation functions on average obtain better results than sigmoid activation functions in the field of practice. ReLU is frequently used in hidden layers and output layers, particularly when the response variables are continuous and positive.

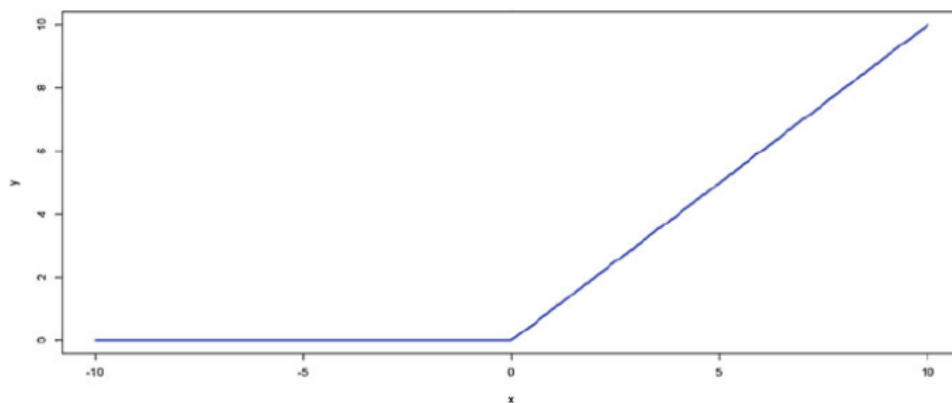


Figure 2.1: ReLU activation function graph.

Source: (Montesinos López et al., 2022)

### 2.3.1.2 Softmax Activation Function

The softmax activation function is often used at the CNN fully connected layer to provide a dependent probability output, which corresponds to the chance that each class fits the input (Emanuel et al., 2024). Thus, the softmax activation function will convert the integer values delivered by the fully connected layer into probabilities and the resulting values lie within the range  $[0, 1]$ . Furthermore, it makes sure the probabilities of all the events sum up to 1 and

facilitates interpretation as the real probability distribution (Es-Sabery et al., 2021). The softmax activation function is calculated by (Goodfellow et al., 2016):

$$f(x_i) = \frac{e^{(x_i)}}{\sum_j^K e^{(x_j)}} \quad (1)$$

Besides, the softmax activation function is also commonly used in multi-class models, consisting of the computation of the probabilities for each class with the highest probability selected as the target class. It is another essential layer for the output implementation of deep learning designs which are utilized in different models.

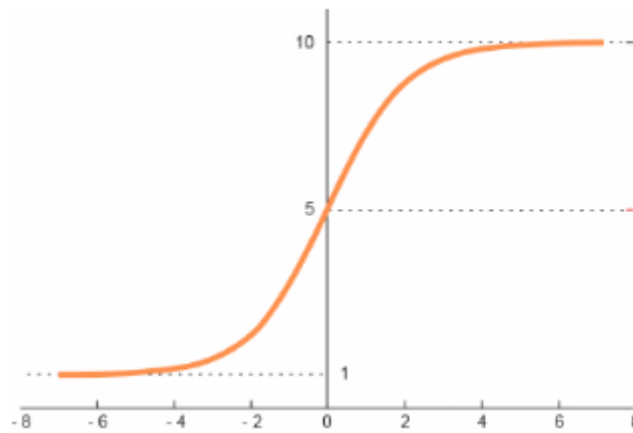


Figure 2.2: Softmax activation function graph.

Source: (Es-Sabery et al., 2021)

### 2.3.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) has become a powerful tool for different computer vision problems due to its ability to recognize and classify picture features (Bhatt et al., 2021). Like traditional neural networks, the architecture of CNN was modelled by neurons found in the brains of humans and other animals. For instance, the CNN simulates the complex cell sequence that makes up the visual cortex in a cat's brain (Hubel and Wiesel, 1962). This biological inspiration provides insights into why CNN excels over other models in autonomously identifying significant characteristics without human oversight.

Besides, in the year of 1990, LeCun et al. introduced a seminal paper that laid the foundation for the contemporary Convolutional Neural Networks (CNN) framework (Cun et al., 1989). Their work led to the development of LeNet-5 which is a multi-layer artificial neural network renowned for its precision in classifying handwritten digits. Which is also similar to other neural networks, LeNet-5 consists of multiple layers and can be trained using the backpropagation algorithm (Gu et al., 2015). Hence, LeNet-5 is noteworthy for its capacity to produce effective representations of original pictures, which makes it possible to directly identify patterns in raw pixel data with little preprocessing requirements. Thus, taking LeNet-5 as an example to study the CNN architecture has more advantages due to its simple architecture compared to modern CNN architecture. As shown in Figure 2.1, CNN architecture consists of three types of layers convolution, pooling, and fully connected layers.

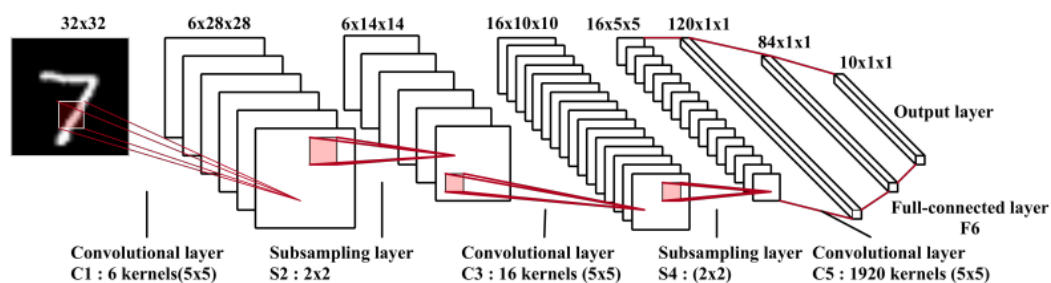


Figure 2.3: LeNet-5 network.

Source: (Gu et al., 2015)

The first layer in the CNN architecture, the convolution layer, convolves a convolution of two digital signals by using a set of matrix operations. Such an operation involves shuffling a tightly packed convolutional array referred to as kernel filter atop an input data tensor. The banking filter likewise bears the same four-dimensional shape as the input tensor but bears a proportionately smaller constant parameter value (Purwono et al., 2023). Let's take an indicator example the input image tensor has a dimension of  $32 \times 32 \times 3$  (height, width, channel). A filter size of  $5 \times 5 \times 3$  is often used, and the stride is set to 1. Hence, this boils down to the fact that the corresponding kernel filter must be smaller than the input tensor dimensions.

The kernel calculates an element-wise product of its coefficients and the input tensor values by selecting pairs of coefficients and matrix elements that are in each overlapping region. The next step would involve passing those products through a layer called a convolutional layer. This results in the development of a data set represented by the output tensor. Such a spatial location is related to a particular activation map (Purwono et al., 2023). The main rotation procedure is exemplified in a particular pattern. It is presented in such works just as it is done in different research works. Through this mathematical representation, the computational process drawn by the CNN is illuminated in the ending with receiving the activation map result. A convolutional layer is an indispensable part of representation learning, as it significantly assists in feature extraction from the input data helping subsequent layers to capture higher-level data structures where simpler features in the input data turn into more complex features (Alzubaidi et al., 2021).

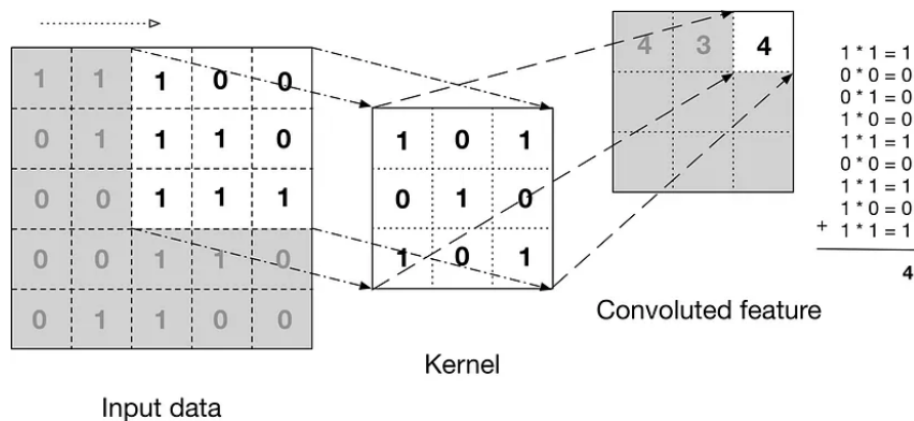


Figure 2.4: Convolutional Layer.

Source: (Yin, 2018)

Following the convolution layer which extracts low-level features from the input data, the pooling layer is the second layer of CNN architecture, and it works to concisely and comprehensively summarize the information from the convolution layer. Not only do the successive convolutional layers proceed directly; instead, the pooling operation serves as a bridge, which lowers the spatial dimensionality of the feature maps generated by the convolutional layer (Purwono et al., 2023). In this

scenario, subsampling is a mechanism of dimensionality reduction, where the sliding pooling layer repeatedly covers the feature maps with small windows. For every window, it calculates a simple summarization statistic such as the maximum or the average value. Such border value is then aggregated to give the output, which is a single scalar with no values window. Through the process of the pooling, the pooling layer is capable of condensing the feature maps, thus greatly reducing the computational burden for subsequent layers as well as introducing a degree of translation invariance, which allows the network to detect local features, irrespective of slight positional displacements.

However, different types of pooling strategies exist, including tree pooling, gated pooling, and min pooling, yet the most common are max pooling and mean pooling. Max pooling indicates and propagates the maximum activation in each window, filtering out the rest of the activations, whereas average pooling focuses on the mean value, combining all activations into a single summarized representation (Alzubaidi et al., 2021). Another essential layer is a pooling layer that allows deeper CNN models to do feature information processing and, at the same time, summarization efficiently.

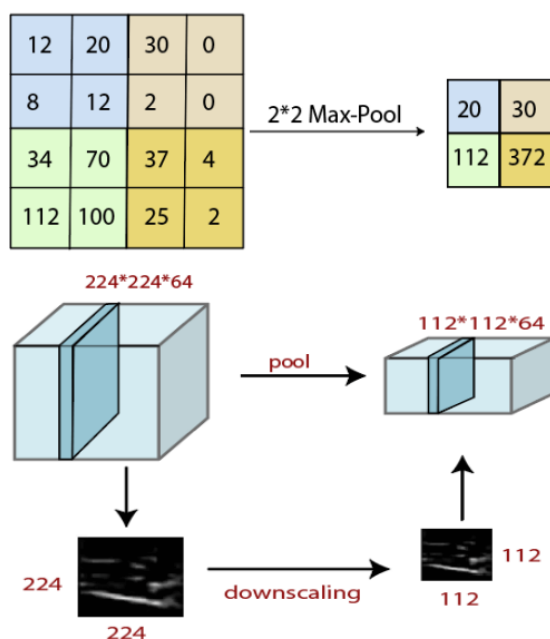


Figure 2.5: Max pooling.

Source: (Prasad and Senthilrajan, 2021)

Then, the fully connected layer receives the result of the addition of the convolutional and pooling groups of operations, which deliver the multi-dimensional feature maps replete with specific representational knowledge. But to display these deep layer architectural specifications, this feature maps tensor undergoes a flattening transformation that stretches it out and keeps all the feature information by that until the next layer processing. (Basha et al., 2020)

In the fully connected layer, every single neuron computes a weighted sum over the flattened feature vector by adding a bias term in the summation process, which helps to control the activation term. Here, these non-linear products are carried out by target functions which play the same role as the operations performed in the neural networks'(Alzubaidi et al., 2021). Hence, these activations, when added together, serve as the final product of the CNN model. Such gets represented in the form of class scores, probabilities, or as an outcome (yes/no).

Significantly, the fully connected layer's dense interconnectivity outstandingly gives it a sensational ability to model complex, multi-dimensional correlations and influence within feature space due to the convolutional and pooling stages (Taye, 2023). This continues to introduce joint learning of both features via dense connections of the neuron level that simulate the combination and refinement of the layers to result in advanced concepts for the network to perform robust classes. The dense connections also help to position the network to predict complex behaviours as a result of the ability to extract and join together abstract features which are needed for sophisticated tasks.

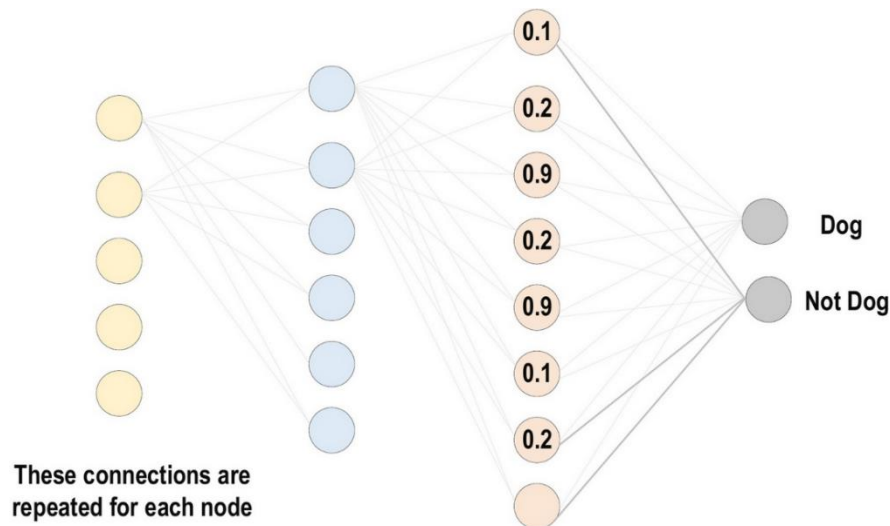


Figure 2.6: Fully connected layer.

Source: (Alzubaidi et al., 2021b)

By understanding the fundamentals of CNN architecture, there are different modern CNN architecture which is the enhancement of the fundamental CNN architecture. These modern CNN architectures are employed in similar violence detection projects, showcasing the evolution and versatility of CNN technology in addressing such challenges and providing different results.

### 2.3.2.1 3D Convolutional Neural Network

In 2017, a study proposed factorizing the 3D convolution into a sequential combination of 2D spatial convolutions followed by 1D temporal convolutions (Tran et al., 2017). This R(2+1)D convolution model outperformed I3D pre-trained on ImageNet by 2.2% on RGP input and 3.2% on optical flow input. However, when fusing the two streams, R(2+1)D was slightly worse than I3D by 0.3%. Thus, it is suitable for pre-trained similarly to I3D, but R(2+1)D may perform well on specific datasets from scratch whereas the I3D performed well on large-scale datasets which are widely adopted due to many cases needed to perform training with large-scale datasets. Additionally, because the main goal of this research is to examine the recognition of action and not that of violence, it conducts a critical analysis of a variety of legitimate and illegitimate approaches for the implementation of 3D CNNs in definite applications.

Besides, 3D skeleton points interaction learning (SPIL) was also introduced in the year 2020, it can focus on the human skeleton point and perform video violence detection (Su et al., 2020). 3D SPIL uses the SPIL model to differentiate the weights among the human skeleton points to capture the unique motion action of individuals. Then, a multi-head mechanism will be added to process various information simultaneously and aggregate them, which will reduce the time to train (Vaswani et al., 2017). The multi-head mechanism will act as a connection between the joint points of the human skeleton point which the joint points will constantly change its position. For instance, each head focuses on a particular aspect of one's behaviour or interaction between individuals and makes the AI model easier to analyze and learn. Although 3D SPIL outperforms the other existing methods with different violence or nonviolence datasets. However, 3D SPIL also has a chance to misclassify as violent behaviour from the datasets, such as actions that are not genuine violent acts but are similar to violence.

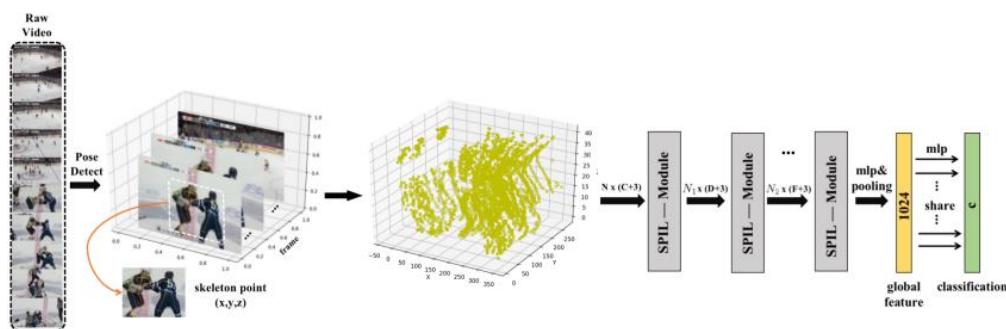


Figure 2.7: SPIL overview model pipeline.

Source: (Su et al., 2020)

### 2.3.2.2 Mobile Convolutional Neural Network

Mobile Convolutional Neural Network architectures are particularly made to be lightweight which is suitable for deployment on resource-constrained devices like mobile phones, embedded systems, and IoT devices. Thus, Mobile CNN aims to achieve a trade-off between model size, computational efficiency, and performance, allowing them to perform such tasks as image recognition, object detection, and other computer vision tasks on devices with limited resources (Gunawardena et al., 2022). Hence, unlike the traditional sequential stacking



order, these networks use layers that are located both serially and in parallel, often grouped into modules that are fused to give rise to the entire architecture. For instance, in the form of the Inception module, which was first introduced to the world in the context of the Inception model,  $1\times 1$ ,  $3\times 3$  and  $5\times 5$  together with the max-pooling layers achieve the multi-level features extraction and reduce the count of the parameters at the same time (Vieira et al., 2022).

One of the lightweight Mobile Convolutional Neural Network architectures, known as SqueezeNet, attains comparably similar accuracy to the groundbreaking AlexNet model but has a much smaller parameter count (Ucar and Korkmaz, 2020). The strategic combination of squeeze and expand layer types makes the design sustainable and convenient for the user. The network uses only  $1\times 1$  convolutional filters in the squeeze layers whereas the expand layers employ a mixture of  $1\times 1$  and  $3\times 3$  filters (Vieira et al., 2022). The selective application of  $1\times 1$  filters with a much lower number of parameters compared to the  $3\times 3$  filters leads to a notable decrease in the model parameters in SqueezeNet.

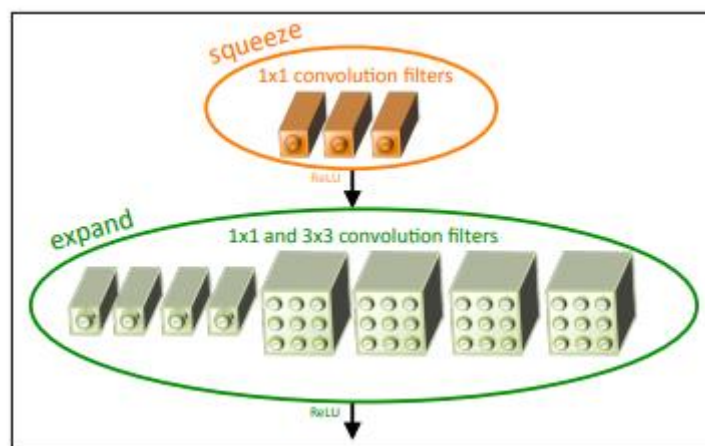


Figure 2.8: The squeezed module.

Source: (Iandola et al., 2016)

Besides, the MobileNet-v1 architecture was the first to ever utilize a depth-wise separate convolution method as a parameter reduction technique. This technique deconstructs traditional convolutions into two distinct stages: a depth-wise convolution that makes use of filters that are applied independently across each input channel, and a pointwise convolution that performs  $1\times 1$

convolutions over all the channels at the same time (Howard et al., 2017). Comparatively, this decomposition approach has a profound effect in reducing the number of parameters required to be trained. As a consequence, architectural designs will be more streamlined and efficient.

To look for ways of making the already successful MobileNet-v1 even better, the MobileNet-v2 architecture conceptualized the inverted residual sparse architecture that facilitates the development of more compact models. This implementation of structure uses the operation sequence consisting of  $1 \times 1$  convolutions depth-wise separable convolutions and linear functions (Sandler et al., 2018). Via this well-designed structure, MobileNet-v2 reaches even lower parameters and much higher performance in comparison with others, this move is bringing efficiency to the next level.

The NASNet architecture achieves parameter optimization by dynamically searching for optimal convolution layer combinations (Zoph et al., 2018). Starting with an architecture that has been tested on a limited dataset, the algorithm searches and tweaks the configuration, ultimately expanding to a larger dataset for validation. The NASNet portable version specialized to run on resource-limited environments comprises sparsely connected convolution layers with variable filter sizes ( $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$ ), which makes the model power and efficient (Vieira et al., 2022).

### **2.3.3 Recurrent Neural Network Architecture (RNN)**

The Recurrent Neural Networks (RNNs) are being utilized in current research projects, recognized as studying powerful learning tools that can recursively compute new states by applying a series of transfer functions to prior states and inputs. Such transfer functions, in most cases, will include the initial affine transformation along with the nonlinear operations, which will be adapted as per certain specific problem scenarios. Mass et al. breakthrough study offers RNNs using the universal approximation property to model accurate nonlinear system complexities with the highest possible precision (Maass et al., 2007).

RNNs' architecture is essential since it determines the information flow between neurons and subsequently influences the network's learning efficiency. RNNs can forecast future outcomes when fed sequential data in predictive tasks due to optimization techniques such as gradient descent, which are primarily

based on the minimization of the difference between predicted and actual outputs. A unique and surprising thing about RNNs with good training examples may also have generative capabilities of producing sequences similar to training data.

The structural anatomy of a basic RNN is made up of interconnected input, hidden, and output nodes, which are tightly woven into a dense dynamic topological graph (Koutník et al., 2014). Time-shift operators come with the ability of nodes to tap into past and future temporal data, leading to the possible capturing of complicated temporal dependencies. Though conceptually representable, the practical application of RNNs may face difficulties, such as collecting established techniques along with the quicker development of architectural designs.

The resolution of such issues might be provided with the use of different methods of training that include conventional gradient-based learning, real-time recurrent learning, and evolutionary algorithms (Bianchi et al., 2017). There may be new ways being discovered, but the effective training of RNNs remains a daunting task that requires a good deal of expert knowledge, which is a sign of their complexity. Besides, there are RNN variants utilizing different training strategies that are no longer based on gradient calculations to reach acceptable results.

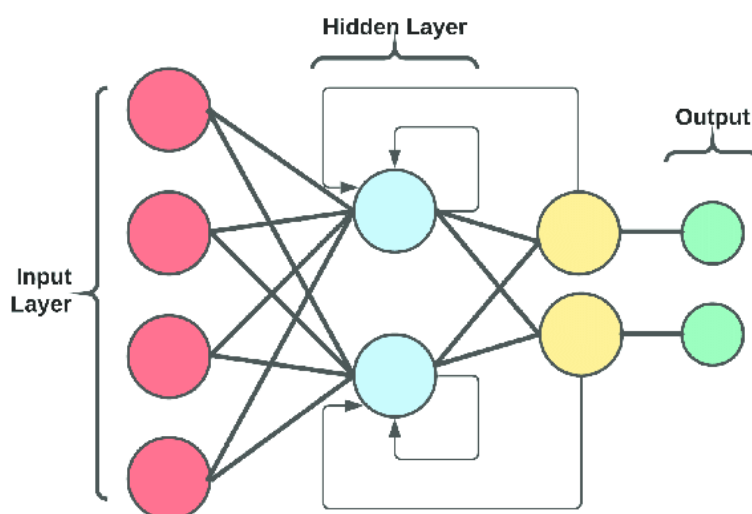


Figure 2.9: RNN architecture.

Source: (Arias et al., 2022)

### 2.3.4 Long Short-Term Memory (LSTM)

The LSTM architecture which was proposed by Hochreiter and Schmidhuber in the year of 1997 is an accurate recurrent neural network that can deal with the exploding or vanishing gradient problems that are faced during the training process of the neural networks as well as across the long dependencies range (Hochreiter and Schmidhuber, 1997). This model utilizes a unique method referred to as the constant error carousel (CEC) in which these units are integrated with an internal error signal. The fact that the patterns are repeated and perfectly executed may indicate an intentional design. They include both input and output gates for their operation (Van Houdt et al., 2020). On top of it, the recurrent loops can operate in a way where a one-step delay of feedback becomes possible.

A comprehensive investigation into the performance of various LSTM variants was conducted by Greff et al. in their study titled "LSTM: A Space Odyssey" (Greff et al., 2015). They compared eight LSTM configurations that were used in three different tasks: speech recognition, handwriting recognition, and polyphonic music modelling. These changes were represented by various modifications including input, forget, or output gates switch-off, aggressive activation functions, coupled gates, peephole connections, and recurrent gates. The study strictly used activities of the normal activation functions which include sigmoid and hyperbolic tangent functions. In addition to that, the role of hyperparameters including hidden layer size, learning rate, momentum, and input noise in performance was investigated, only through an experimental procedure in which random values were selected within specific ranges to avoid exhaustive testing. Among the bottom lines, it can be seen that the learning rate plays the central role, and after comes the hidden size layer. Intriguingly, the inclusion of noise at the inputs was found to decrease the accuracy and to lengthen the training length as well (Bolboacă and Haller, 2023). Surprisingly, system inertia did not display any evident impact on performance and training variables. However, after all tests were done, LSTM showed itself to be a strong competitor among the other alternatives when compared.

Moreover, in the year of 1997, Schuster and Paliwal proposed the 2 separate networks as Bi-directional Recurrent Neural Networks (BRNNs), the data could be processed from both directions (Schuster and Paliwal, 1997). This

way was favourable within this context that the beginning and end of data were known in advance, for example within phoneme boundary estimation. A BRNN is a network in which neurons are split into feed-forward and feedback states thus granting the network processing capabilities in both directions of time flow. Moreover, in the year of 2018, Aziz Sharfuddin et al. offered the use of BiLSTM which employs two LSTM layers, one for the memory of events in the past and the other for predictions about future states (Aziz Sharfuddin et al., 2018). Similar to Zhao et al. in the year of 2018 as well, the BLSTM architecture was explained, and the language of BiLSTM emphasized the back-and-forth structure of bidirectional flow and unidirectional flow between input, hidden, and output layers (Zhao et al., 2018).

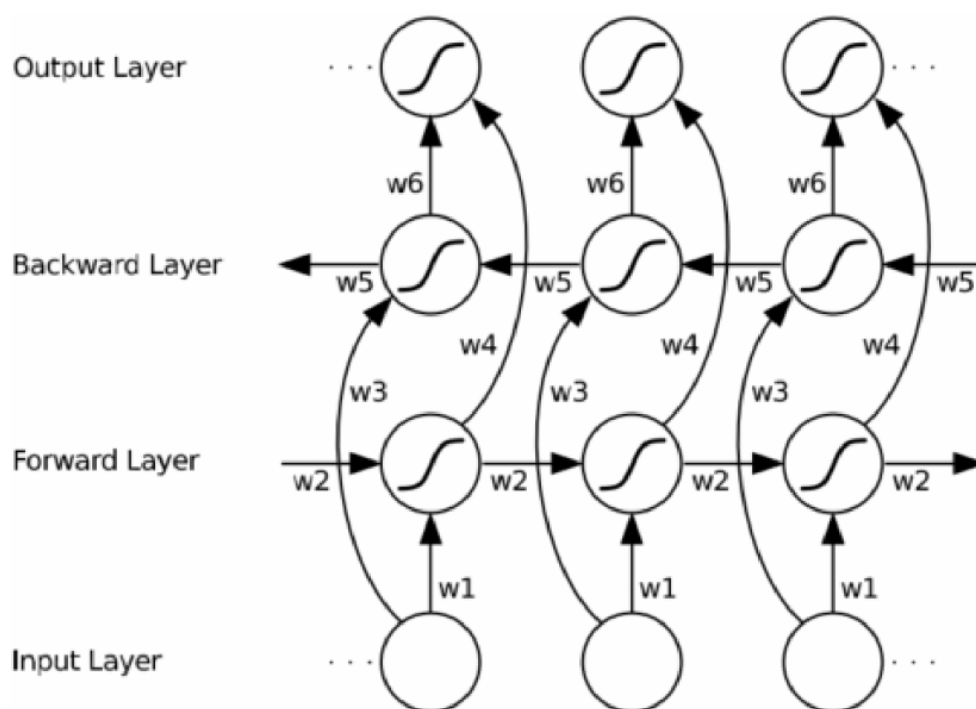


Figure 2.10: BiLSTM architecture.

Source: (Aziz Sharfuddin et al., 2018)

### **2.3.5 Hybrid CNN + LSTM Architecture**

In the year 2023, the authors, Tiwari et al. provided an entirely new concept to detect violence by taking advantage of the key factors of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks (Tiwari et al., 2023). The proposed method by the authors uses CNN and LSTM hybrid to detect the violent activities in the surveillance footage. By doing this they capitalize on the advantages given by each type of neural network. As a result of thorough investigations and side-by-side studies, it was established that by using a hybrid CNN-LSTM model performance of such methods was improved in terms of accuracy. Recognizing the abnormality, the model had an accuracy of 98.63% in identifying violent behaviours present in video sequences.

Furthermore, another approach uses 2D spatio-temporal convolutional networks (STCNN) to extract the local features and then incorporate recurrent networks (RNN) to improve the refinement. The utilization of this holistic method offers not only computational effectiveness in execution but also brings about the promise of good results that can be seen (Traore and Akhloufi, 2020). First and foremost, Abdali and Al-Tuma in the year 2019 used VGG19, which was initially used on the ImageNet dataset as a method of feature extraction. The extracted features are fed into a long-short term memory (LSTM) for further processing and then are taken through a time distributed fully connected layer through which violence is detected. The success of this technique was confirmed across the datasets of the Violent Flux, Hockey, and Movies, and it returned competitive performance.

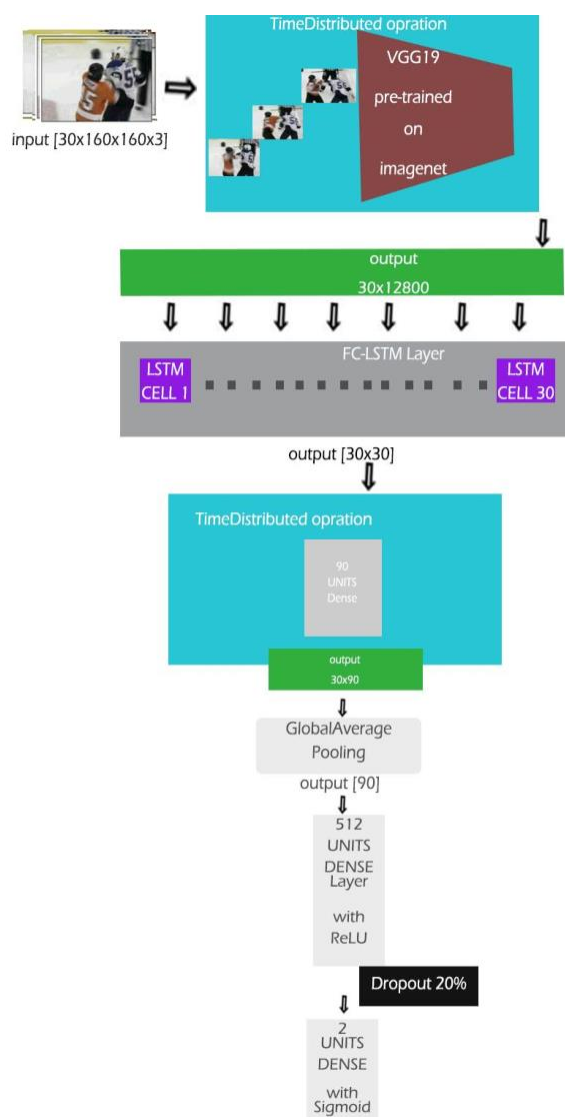


Figure 2.11: VGG19 with LSTM model.

Source: (Abdali and Al-Tuma, 2019)

Moreover, in the year 2020, Halder and Chatterjee proposed a study consisting of utilizing a CNN-BiLSTM model for verifying violence in sequential video frames (Halder and Chatterjee, 2020). First, video frames are being extracted and are going to pass through the CNN for a feature extraction process. Afterward, a Bidirectional LSTM block is formed that analyses past and future frames along with current frame information to identify temporal patterns of violence. Then, a classifier identifies if there are any violent types of human behaviours and patterns balancing in both spatial and temporal features, which results in highly upgraded accuracy prediction analysis. The proposed

CNN-BiLSTM model demonstrates a great performance, with accuracies being 99.27%, 100%, and 98.64% correspondingly to the Hockey Fight, Movie, and Violent Flow sets respectively, and superior to the majority of other methods. Significantly, it is lightweight, which means it has minimal training time and fewer epochs than the previous models.

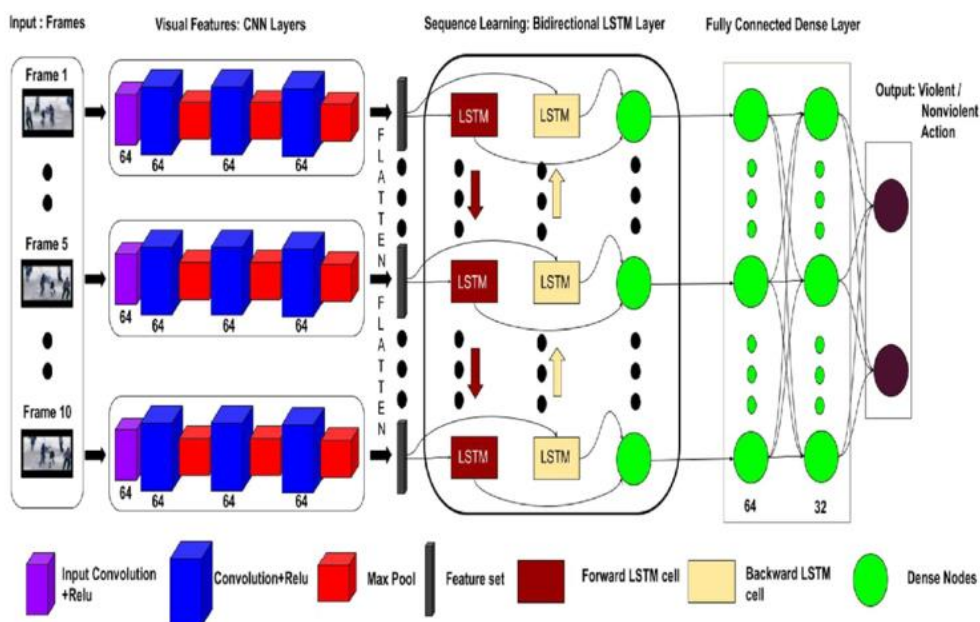


Figure 2.12: CNN-BiLSTM model.

Source: (Halder and Chatterjee, 2020)



## 2.4 Summary

The comparison of the deep learning models is based on the literature review of different deep learning models that have successfully detected violent and nonviolent behaviour in different datasets with high performance. Thus, these deep learning models are compared based on their accuracy, which measures their capability to distinguish between violent and nonviolent behaviour.

Table 2.1: Comparison of different deep learning models.

Model	Datasets	Accuracy (%)
3D SPIL (Su et al., 2020)	RWF-2000: 2000	89.30
	Hockey-Fight	96.80
	Crowd	94.50
	Movies-Fight	98.50
SqueezeNet (Vieira et al., 2022)	Violence: 1193 Nonviolence: 1477	87.00
MobileNet-v1 (Vieira et al., 2022)	Violence: 1193 Nonviolence: 1477	90.00
MobileNet-v2 (Vieira et al., 2022)	Violence: 1193 Nonviolence: 1477	92.00
NASNetMobile (Vieira et al., 2022)	Violence: 1193 Nonviolence: 1477	90.00
CNN-LSTM (Tiwari et al., 2023)	Violence: 5842 Nonviolence: 5231	98.63
VGG19-LSTM (Abdali and Al-Tuma, 2019)	Hockey-Fight: 1000	98.00
	Hockey-Fight + Movie-Fight: 1259	94.77
CNN-BiLSTM (Halder and Chatterjee, 2020)	Hockey-Fight: 1000	98.78
	Movie-Fight: 200	99.66
	Violent Flows: 246	98.18

Based on Table 2.1, the highest accuracy performance is the CNN-BiLSTM model which achieves 99.66% for the Movie-Fight datasets, whereas the second highest accuracy performance is still the BiLSTM model due to it

achieving 98.78% accuracy. Although BiLSTM achieves high accuracy among others, the datasets they use are smaller than other models. Thus, the CNN-LSTM model comes in better in the case of relatively larger datasets where the model achieves a satisfactory 98.63% accuracy rate value. It is also worth mentioning that CNN-LSTM accuracy is lower than that of CNN-BiLSTM, with a difference of about 0.15% in the accuracy level if compared with the Hockey Fight dataset. Therefore, the CNN-BiLSTM and CNN-LSTM have the highest accuracy performance among others.

Using the CNN model which requires large resources might cause a delay in the prediction time. The consequence of the delay in time to predict violent behaviour might be a concern. Therefore, the MobileNetV2 model may help to prevent such issues due to it is a lightweight CNN model which suitable for mobile devices or low power devices. The MobileNetV2 itself scores a 92% accuracy as shown in Table 2.1, which is higher than the other 3 independent CNN models.

In conclusion, this research project focuses on using MobileNetV2 with LSTM to test the overall performance and improve the accuracy of video-based violence detection.

## CHAPTER 3

### METHODOLOGY AND WORK PLAN

#### 3.1 Introduction

The methodology for violence detection using deep learning involves building a model that accurately predicts the behavior of violence in public videos through feature extraction, classification using a deep learning classifier, and sequential pattern recognition.

#### 3.2 Experimental Setup

This section's main aim is to demonstrate the project's employed hardware and software infrastructure, which are shown in Table 3.1.

Table 3.1: Hardware and Software Specifications.

Hardware	CPU 9 <sup>th</sup> Gen Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
	8 GB RAM
	GPU NVIDIA GeForce GTX1050
Software	Kaggle / Google Collab
	Python Libraries (Tensorflow, Keras, Open CV)

### 3.3 Workflow plan

Figure 3.1 presents the workflow flowchart, with each step explained in detail in the subsequent sections.

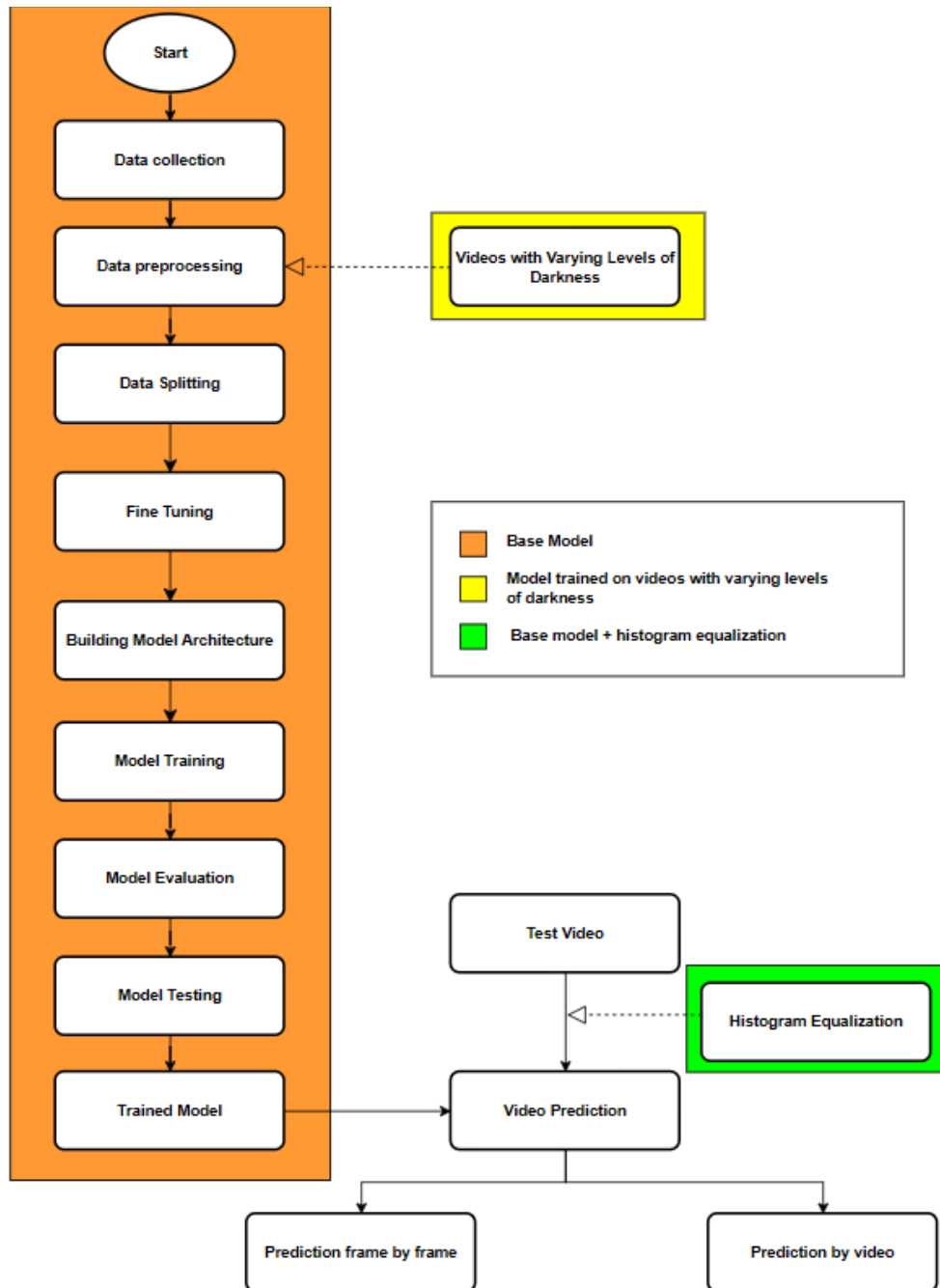


Figure 3.1: Flowchart for the Entire Workflow.

### 3.3.1 Data Collection

The dataset for this project is mainly video which is similar to the CCTV footage. Thus, this video dataset contains quite a few types of actions and situations that can be found in the conditions of surveillance systems. Each video clip in this dataset is a session of footage in which events and behaviour in the monitored areas are captured and this visual data provides a good substratum of information for analysis and modeling. Moreover, the dataset also includes indoor and outdoor environments which consist of violent or nonviolent behaviour. Besides, the video datasets also consist of videos along with violence and nonviolence labels. The labels depict what the video shows, including violent or nonviolent behavior in the footage. These representations act as the ground truth annotation for training and for the evaluation of the deep learning models and henceforth supervised learning approaches for violence detection and classification tasks can be performed. The dataset is an online source from Kaggle and contains 1000 videos for violence and 1000 videos for nonviolence. The datasets consist of real-life situations that stimulate the CCTV footage due to the quality of the video being similar to CCTV footage. Moreover, there will be another dataset used for this research as it contains 2,486 videos in total, which have 1199 videos as violence and 1287 videos as nonviolence. This dataset will be further preprocessed by lowering the contrast to stimulate a dark environment video and allow the model to train.

Table 3.2 Summary of Dataset Used

Dataset	Total Videos	Violence Video	Nonviolence Video
Real Life Violence Situations Dataset (Soliman et al., 2019)	2000	1000	1000
Smart-City CCTV Violence Detection Dataset (SCVD) (Aremu et al., 2024)	2486	1199	1287

### **3.3.2 Data Preprocessing**

In the preprocessing stage, several important steps are required to prepare the video data, before putting it into the deep learning algorithms to train it.

#### **3.3.2.1 Video Frame Extraction**

Initially, frames are extracted from the video files to produce a series of frames in which each sequence forms a single video sample. These frames record the temporal changes of scenery and become the fundamental decisive factor for the subsequent analysis. Consequently, to make sure that images of the same size are used across all frames, each frame undergoes resizing to a fixed height and width, which could be set as 64x64 pixels. The size standardization is achieved to ensure uniformity in spatial dimensions. This avoids any interference with the neural network model processing such as distortion and blurriness.

#### **3.3.2.2 Normalization**

After the pixels are resized, each frame intensity value is normalized in the range 0 and 1. This normalization process includes dividing every pixel value by 255, whereby the intensity values of all the frames are standardized. Therefore, it is necessary for problem-handling with feature scaling and model learning processes.

#### **3.3.2.3 Video Dataset Construction**

After that, the dataset is constructed by iterating through one video file after another in the dataset directory. Then, the video is split into individual frames next, which form a series. The set of these frame sequences that are grouped with either violent or non-violent class labels is then added up to form the dataset.

### **3.3.3 Data Splitting**

In the data splitting process, the dataset is divided into three distinct subsets: a training data sample, a validation data sample, and a test data sample. This delimitation is of great importance for acquiring and refining the model that is free from any errors and then this model is used to assess the model's performance. Therefore, the data are divided as a training set of 85% while a

15% split is allocated to the testing sets. The validation sets will further split 15% out of the 85% of training sets in the training model. Thus, this manner of distribution gives samples that will be used for model training and validation which are the necessities, and also for verifying the performance of the model.

### **3.3.4 Fine Tuning**

Following the data division, the CNN model is applied as a feature extractor, an approach that uses its pre-trained weights to capture key features from the input images. The upper layers of CNN, which are meant for classification tasks, are skipped to enable the model to be trained on violence detection. Fine-tune is then done by making the last 40 layers trainable and freezing the rest of them. In this way, selective tuning helps the model to remain faithful to the feature information learned from the pre-trained weights and simultaneously match it to the particulars of the violence detection task.

### **3.3.5 Model Architecture**

The model architecture is similar as shown in Figure 2.12, the starting of the architecture is the input layer taking a 5D tensor that represents sequence length, image height, image width, and color channels (RGB). The convolutional backbone is a set of four residual blocks, each block contains three convolutional layers with 3x3 kernels, after which a batch normalization and ReLU activation takes place. The number of filters in each block is 64, 128, 256, and 512 in this order. Through residual convolutional blocks, these spatial features are extracted from the individual images and then fed into the TimeDistributed layer, which are feature maps.

The TimeDistributed layer denotes the convolutional backbone applied separately to all time steps of the input sequence, enabling the extraction of spatial features on individual images within the stream. The output of the TimeDistributed layer is transformed through a TimeDistributed Flatten layer to a single dimension with all spatial features combined. Then 256 units of BiLSTM layer process the earlier flattened feature maps with temporal dependencies and context in both past and future sequences.

The output of the bidirectional LSTM is passed to two fully connected layers with 256 and 128 units, respectively using ReLU activation. Dropout

operation with a rate of 0.3 is done after dense layers to prevent overfitting. The following layer is a dense layer with a SoftMax activation function, which can achieve class probability scores for the predefined classes.

### **3.3.6 Model Training**

The training methodology commences by setting up two essential callbacks: Early stopping and learning rate reduction. The callback for early stopping is looking into the validation accuracy after every epoch, thus terminating the process immediately without any improvement over an assigned number of epochs and, hence avoiding overfitting. On the same lines, the callback ReduceLRonPlateau also decreases the learning rate if the validation error is approaching the plateau, leading to better training and preventing the overfitting issue.

Upon setting the callbacks, the model is compiled using categorical cross-entropy for its loss function, stochastic gradient descent as its optimizer, and accuracy as its evaluation metric. An epoch is set in which 8 batches are used with data shuffled before every epoch to achieve generalization. In addition to that, 15% out of 85% portion of the training data is reserved for validation which makes sure the model is evaluated on unseen data during training. These callbacks EarlyStopping and ReduceLRonPlateau are implemented to keep track of the training dynamics and adjust the learning rate as needed.

### **3.3.7 Model Evaluation**

Next is to evaluate the trained model using two graphs, one is the total loss versus the total validation loss and the other is to demonstrate the total accuracy versus the total accuracy in the validation. Through the analysis of these graphs, it will evaluate how well the model performs in terms of loss decrease and accuracy improvement throughout the training process.

### **3.3.8 Model Testing**

The testing process of the model starts by making predictions based on the model parameters learned from the training phase. Then, the calculated labels are compared with true labels to determine prediction accuracy. Besides, the confusion matrix and heatmap are used to provide a look into the model's



accuracy in distinguishing between various classes through true positive (TP) and true negative (TN), which are determined as successfully classifying violence and nonviolence respectively. False positive (FP) and false negative (FN) are determined as misclassifying violence and nonviolence respectively. A classification report is also generated to provide exact precision, recall, and F1-score for each class of data to measure the performance of the classification model. Hence, some of the calculations of the evaluation metrics also can be used for the testing phase, which is listed below (James et al., 2013):

**Evaluation Metrics Equation:**

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (3)$$

$$Specificity = \frac{TN}{FP+TN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (7)$$

Besides, to measure the efficiency of the particular model to classify the positive and the negative classes of data, there is a standard metric called the

Receiver Operating Characteristic (ROC) curve. The ROC curve is used to represent the specificity and sensitivity different values of a decision threshold takes when being used to classify patterns in a data set. Based on Figure 3.2, a fertile quantity that is obtained from the ROC curve is called the Area Under the Curve (AUC), which measures the performance of the model across various probabilities. If the AUC is close to 1, this means that the classifier has a high capacity for separating one class from another; if its value is 0.5 means the performance level similar to one that is based on chance.

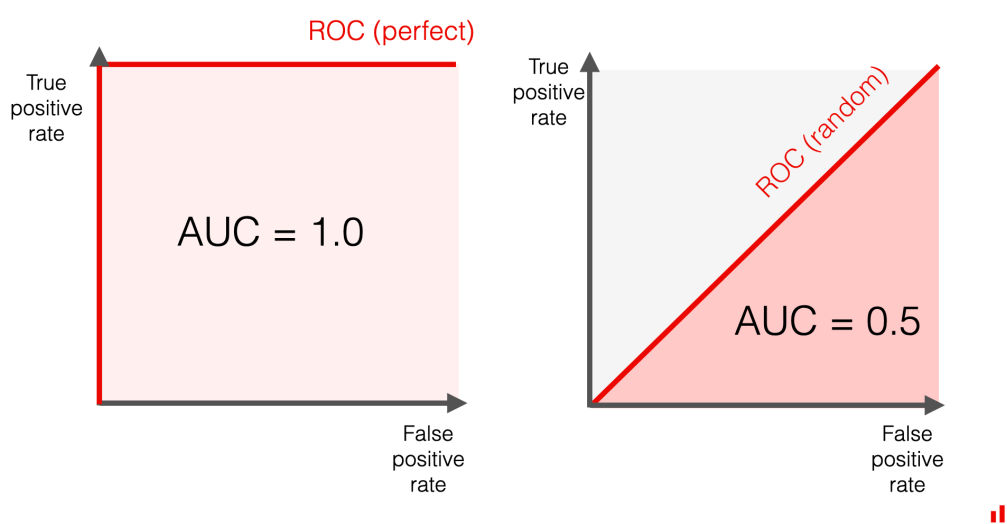


Figure 3.2: ROC and AUC graph

Source: (Evidently AI Team, n.d.)

In summary, precision and recall will be more focused on this research. Due to the main focus is to minimize the false positive (FP) and false negative (FN) as much as possible to prevent causing issues such as false prediction or miss predict before proposing to integrate into real-world systems. Moreover, the implementation of the ROC curve along with its AUC promotes such kind of evaluation by ensuring a more overall assessment of the model's performance within any possible decision thresholds, thus providing robustness in varying situations.

### **3.3.9 Histogram Equalization**

The test video frames are divided into: Normal, Night vision, Dark, and Very dark video. Therefore, to improve the video quality, a histogram equalization was performed on the test video frames, and this technique redistributes the brightness values (luminance) of an image to achieve a more uniform histogram thereby leading to an increase in contrast.

The function then applies the color space conversion on the test video reads it through a frame by a frame manner and then passes through the YCrCb color space, which is a color space that relays the luminance details of an image through the Y channel and the color details through the Cr and Cb channels. A Contrast Limited Adaptive Histogram Equalization (CLAHE) is then applied to the Y channel, in contrast to the normal way of applying CLAHE directly to all three color planes. Unlike the standard histogram equalization, it operates by constraining the contrast in some parts of the image to prevent over enhancement of contrast and noise amplification. It could be of great help in videos where parts of the frame may have different light conditions from the other parts.

The obtained adjusted brightness channel is then combined with the CLAHE processed Y channel to get the final CLAHE processed frame in the standard BGR color space. The modified frame is then stored in the output video with the additional frame.

### **3.3.10 Video Prediction**

#### **3.3.10.1 Prediction Frame by Frame**

Each frame in a video goes through pre-processing again such as resizing and normalization. After that, the frame is appended to a deque (double-ended queue) with a maximum length set to 16, this is to make sure that a sequence of frames is fixed length. After decking a sequence of frames, equal to the number of frames specified, the sequence is passed through the pre-trained model. This sequence is fed in the model that gives this particular sequence a set of output probabilities for different class labels. The final classification label of the sequence is then determined to be the class label with the highest label probability. It designates the action or event in the video as conceived by the model at the above given time.

Moreover, once the sequence length is reached, the frames are visualized for prediction by a pre-trained model. The predicted class label is then written on the current frame to label it or simply to identify it. To improve the readability of the results, labels get distinguished based on the color – the “Violence” label is in red if the content is violent, and in green otherwise. Said frame is then written to an output video file where it is stored with the visual representation of the model’s predictions for further analysis.

At the end of the processing of all frames in the video, the output processed video becomes accessible in frames for review on individual frames. The class label of the image is predicted at each frame, which gives a full view of how the model’s prediction is made gradually. Also used is a function, to review the frames chosen by the algorithm randomly from the processed video, so there is no need to review the whole sequence to evaluate the model outcomes at different moments of the video.

### **3.3.10.2 Prediction by Video**

When making predictions about the video whether it contains violent or nonviolent behaviour, the video will be extracted from the main information such as width, height, and length of the video. After that, the frames are sampled from the video sequence in an ordered manner, where it is adjusted so that all frames are equally represented. For each given frame, it will go through pre-processing again before putting into the pre-trained model. Once the input frames are placed the pre-trained model will predict and calculate the class probabilities for each frame of the video. Once the probabilities are then computed the model assigns the label with the highest probability as the predicted label. The name of the corresponding class is also read from a predefined list of classes. Furthermore, the result visualizes the confidence scores of the prediction and these scores are useful as a pointer to instructively give insight into the model’s prediction accuracy.

### **3.4 Violence Detection Models**

#### **3.4.1 Base Model**

This flow contains the training of the base model (MobileNet-v2 with the BiLSTM) and testing the base model. The flow includes all the orange colour steps which are highlighted in Figure 3.1 and also the video prediction with the test video. Then it will categorize into video prediction frame by frame and the entire video sequence.

#### **3.4.2 Model Trained on Videos with Varying Lighting Conditions**

The base model area is trained with an additional video dataset which contains varying levels of darkness, and also with the original dataset trained with it. This flow is almost the same as the base model, the only difference will be adding the videos with varying levels of darkness dataset. The purpose of this flow is to enhance the model's effectiveness and record the time taken to train the model.

#### **3.4.3 Base Model with Histogram Equalization Enhancement**

Histogram equalization is applied to the test video rather than during the training dataset. After the base model has been trained, the histogram equalization is used to preprocess the test video. This eventually improves the contrast of the video frames, and the trained model is then used to predict the result of the preprocessed test video.

### **3.5 Gantt Chart**

Gantt chart is a comprehensive visualization of the project schedule which allows effective time management, assessing the order of precedence of the tasks, and the identification of interdependency.

The following section describes the details of every methodology that has been used in each research phase which will be illustrated by the Gantt chart of timelines. This integrated approach seeks to clarify and disclose the process of monitoring the execution of the research project according to the agreed-upon time frames and milestones in a bid to remain on track. Figures 3.3 and 3.4 show the gantt chart for fyp1 and fyp2.

Gantt Chart

No.	Project Activities	Planned Completion Date	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17
1.	Project Planning	2024-02-17	■	■	■														
2.	Data Collection	2024-02-24		■	■	■													
3.	Model Selection	2024-03-02			■	■	■												
4.	Literature Review	2024-03-30			■	■	■	■	■	■									
5.	Code Implementation	2024-03-30					■	■	■	■									
6.	Proposal Writing	2024-04-19					■	■	■	■	■	■	■	■					
7.	Prepare oral presentation slides and presentation	2024-04-26											■	■	■				

Figure 3.3: Gantt Chart for FYP1.

Gantt Chart

No.	Project Activities	Planned Completion Date	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17
1.	Project planning	2024-06-23	■																
2.	Collecting Datasets	2024-06-23	■																
3.	Testing base model	2024-06-30		■	■														
4.	Investigate base model limitation	2024-07-07		■	■														
5.	Investigate solution to solve the limitation of base model	2024-08-11			■	■	■	■	■	■									
6.	Performance evaluation and discussion	2024-08-19									■	■							
7.	Poster Preparation	2024-08-29										■	■						
8.	Report Writing	2024-09-02			■	■	■	■	■	■	■	■	■	■	■				
9.	Prepare for presentation and presentation slide	2024-09-20													■	■			

Figure 3.4: Gantt Chart for FYP2.

### **3.6 Summary**

In this research project, the overall workflows are data preprocessing, splitting the data, applying fine tuning, building the model architecture, model evaluation, model testing, and using the trained model to predict the entire video and frame by frame. The research used three types of models: the base model starts with training the MobileNet-v2 with BiLSTM and tests it on standard data; the model trained on videos with varying lighting conditions improves the base model by training more datasets with different levels of darkness; and the base model with histogram equalization enhances the base model in video prediction on test data and enhances the prediction output from the trained base model.

## CHAPTER 4

### RESULTS AND DISCUSSIONS

#### 4.1 Introduction

This research aimed to improve the performance of the violence detection model by adding image processing before apply to the trained model. This section provides the evaluation of the proposed enhancements and insights into the accuracy/computational efficiency trade-offs.

#### 4.2 Performance Evaluation of the Base Model: MobileNet-v2 with BiLSTM

The performance of the base model which is the MobileNet-v2 with BiLSTM was assessed using several key evaluation metrics such as precision, recall, F1 score, and accuracy. The results of these metrics are shown in the table below:

Table 4.1 Evaluation Metrics for Base Model (MobileNet-v2 with BiLSTM)

Evaluation Metric	Value
Precision	0.93
Recall	0.93
F1 score	0.93
Accuracy	0.93

As the table above demonstrates across all the evaluation parameters with precision, recall, F1 score, and accuracy all four stand at 0.93. This supports the base model's capabilities in correctly predicting positive samples and holding lower levels of both false positive and false negative rates.



### 4.2.1 Confusion Matrix of Base Model

Besides, the base model was also evaluated by using the confusion matrix as well.

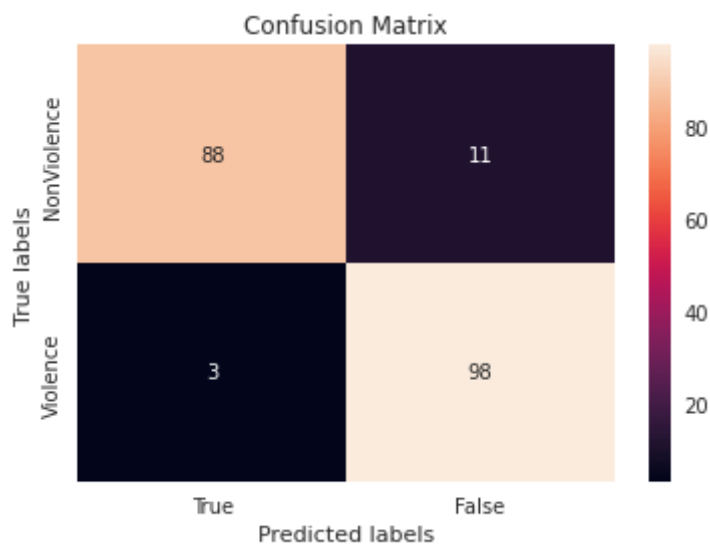


Figure 4.1 Confusion Matrix of Base Model

In the confusion matrix shown in Figure 4.1, the first row corresponds to the non-violence class, and the second row corresponds to the violence class. The values reveal that the model classified 98 actual instances of violence (true positive) while 88 actual instances of non-violence (true negative). However, it classified 11 non-violence cases as violence (False Positives) whereas it failed to recognize 3 actual violence cases (False Negatives).

#### 4.2.2 Evaluation of the Receiver Operating Characteristic (ROC) Curve for the Base Model

Lastly, the ROC curve graph is used to evaluate the performance. The ROC curve plots the True Positive Rate (sensitivity) against the False Positive Rate at increasing threshold values and thus gives an overall idea of how the model discriminates between the classes.

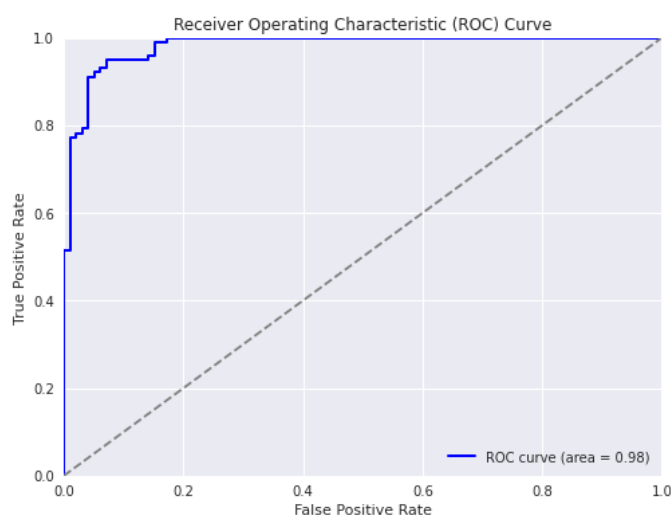


Figure 4.2 ROC Curve Graph

Based on the ROC curve graph shown in Figure 4.2, a strong upward trend is shown which curved towards the top left corner of the graph. This means that the model performs well, where it has a high true positive rate and a low false positive rate across different thresholds. Another measure complementary to this is the area under the ROC curve (AUC) of the classifier where a value closer to 1 indicates better classifier performance in terms of classifying. The AUC indicates that the base model can nearly classify between violence and non-violence cases and help the classification module for total robustness.

#### 4.2.3 Example Results of Base Model Prediction

The base model undergoes two different predictions: predicted as frame by frame (examples shown in Figures 4.3, 4.5, and 4.7) and predicted as an entire video sequence (examples shown in Figures 4.4, 4.6, and 4.8).



Figure 4.3 Prediction as Frame by Frame in Normal Condition



Figure 4.4 Prediction as Entire Video Sequence in Normal Condition



Figure 4.5 Prediction as Frame by Frame in Dark Condition

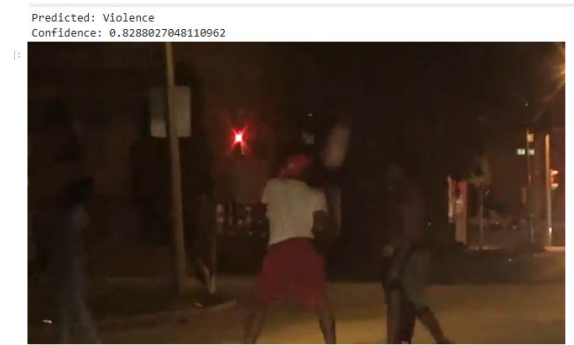


Figure 4.6 Prediction as Entire Video Sequence in Dark Condition

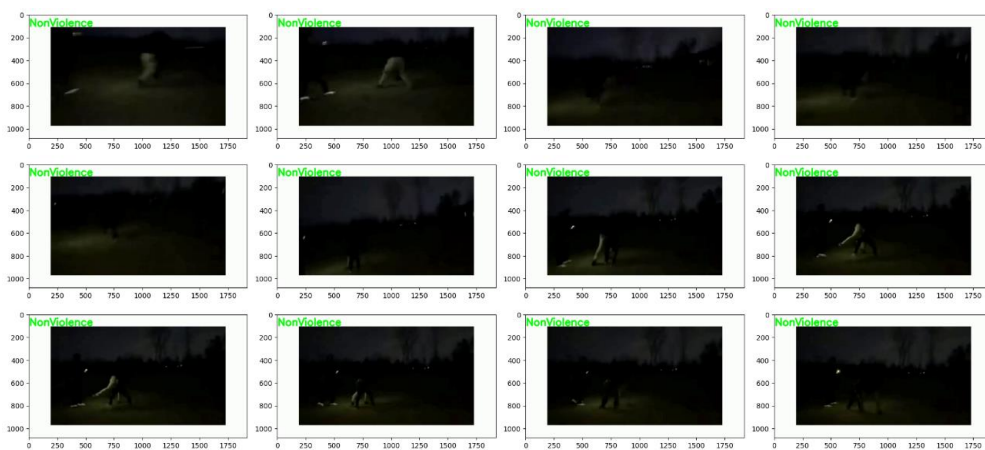


Figure 4.7 Prediction as Frame by Frame in Very Dark Condition

Predicted: NonViolence  
Confidence: 0.7511202096939087



Figure 4.8 Prediction as Entire Video Sequence in Very Dark Condition

### 4.3 Comparison of Violence Detection Models Across Different Lighting Conditions

To evaluate the performance of the deep learning models, three different approaches can be used which are shown in Table 4.1. At first, the models were trained on a normal dataset on top of the MobileNetV2 architecture in combination with the BiLSTM model. This approach served as the baseline for evaluating the model's accuracy. Next, the baseline model was added with more dataset containing videos with varying levels of darkness, along with the normal dataset. Finally, the baseline model was trained with the normal dataset, however during the testing phase the videos were subjected to histogram equalization to increase the performance of the baseline model. Additionally, the test videos in this section which are used for video prediction for all three models are fixed at 30 fps, 5 seconds in length and original resolution.

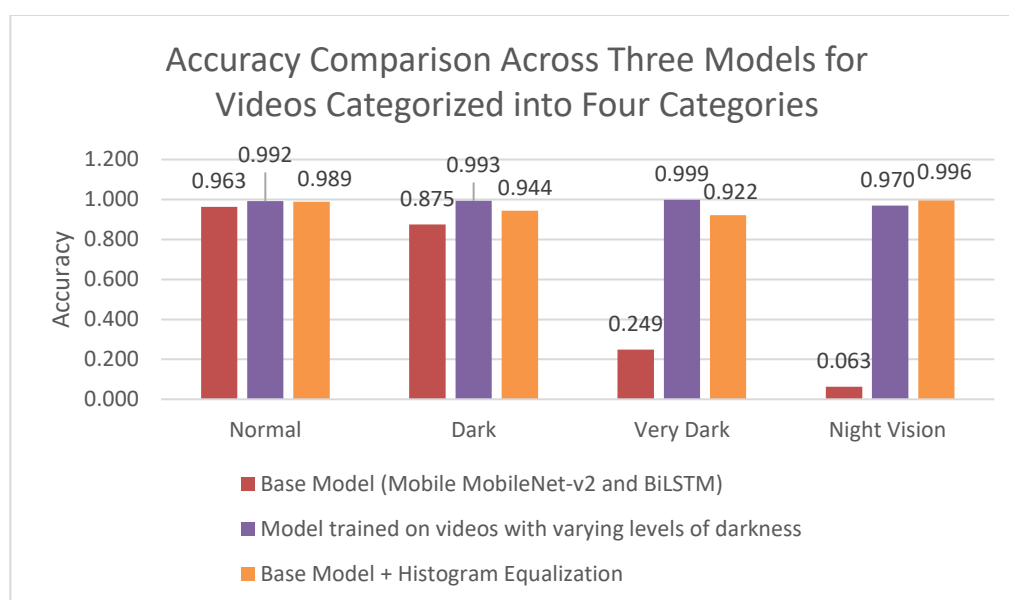


Figure 4.9 Accuracy Comparison Across Three Models Across Different Lighting Conditions

Based on Figure 4.9, the comparison of three models under varying lighting conditions provides insights into the effectiveness of different approaches to handling varying levels of lighting conditions. The base model, MobileNet-v2 coupled with BiLSTM gives good accuracy under normal light conditions with an accuracy of 0.963. However, accuracy reduces significantly as the lighting levels decrease giving an accuracy of 0.875 in dark conditions,

but is similar to the base model with histogram equalization and model trained on the videos with different levels of darkness. However, it shows an accuracy of 0.249 and 0.063 under very dark conditions and night vision conditions, the lowest performance compared to the other two models.

The performance of the model trained on the videos with different levels of darkness is quite impressive across the entire range of lighting conditions. This model gives the highest accuracy in normal lighting at 0.992 and maintains nearly perfect performance even in very dark conditions with an accuracy of 0.999.

The base model with histogram equalization during the test phase demonstrates a significant increase in accuracy under all low-light conditions. The accuracy is slightly lower than the model trained with varying levels of darkness videos. It shows an accuracy of 0.989 under normal lighting conditions, 0.944 under dark conditions, and 0.922 under very dark conditions. However, under night vision conditions, the histogram equalization approach outperforms the model trained with varying levels of darkness videos, 0.996 versus 0.970.

Thus, this indicates that the base model with histogram equalization is able to give similar accuracy to the trained model with varying levels of darkness videos. It is worth mentioning that the proposed histogram equalization does not require extensive datasets and training, yet achieves comparable accuracy.

#### **4.4 Analysis of the Proposed Flow of Base Model with Histogram Equalization**

##### **4.4.1 Performance of Different Video Frame Rates**

The frame rate of the video data not only impacts the performance of analysis models that operate on videos but also has implications for the time necessary to execute operations such as histogram equalization. This section explores the trade-off between precision and performance based on frames per second (FPS).

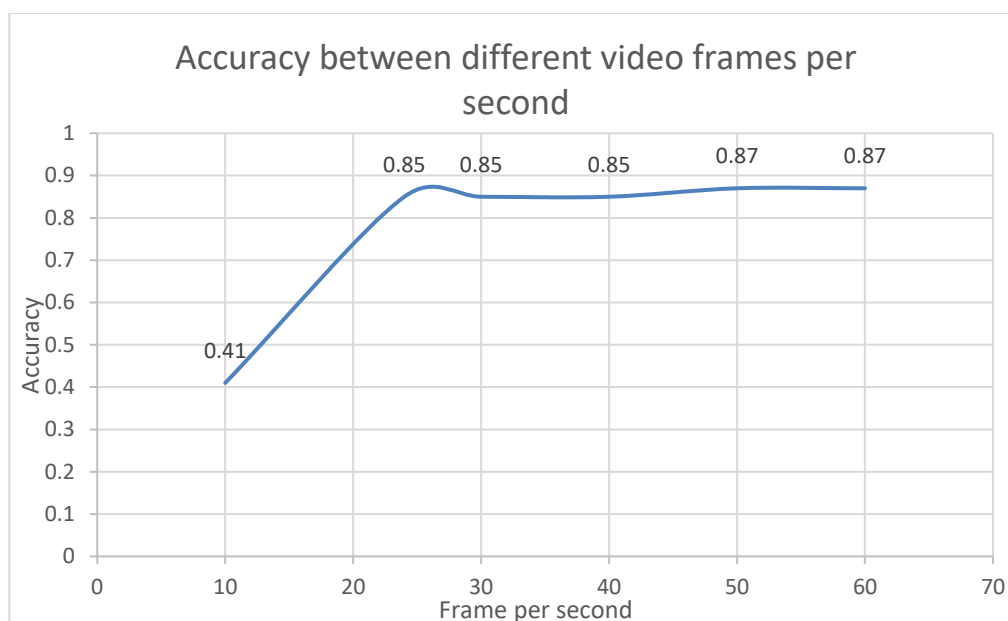


Figure 4.10 Accuracy Across Different Video Frame Rates

The findings of the experiment reveal that raising the video frame rate from 10 FPS to 24 FPS results in a significant boost in accuracy, following an increase from 0.41 at 10 FPS to 0.85 at 24 FPS. However, as the frame rate rises higher than 24 FPS, the improvement in the accuracy is small and remains constant at about 0.87 for frame rates of 50 FPS and 60 FPS. This implies that even though the higher frame rates increase the degree of accuracy it is not proportional and further increases only increase the rate slightly after 24 FPS have been reached.

However, it is still necessary to take into account the dependency of the frame rate on the time it takes to apply some filters such as the histogram equalization. Reduced FPS means faster frame rates to process because it involves fewer frames. This decrease in processing time is especially beneficial in contexts, where efficiency and speed are of paramount importance and hence lower number of FPS is more reasonable and feasible.

Therefore, a frame rate of 24 FPS and 30 FPS is recommended as it offers a well-balanced trade-off between high accuracy and efficient processing time. There is always a gain in accuracy with high frame rates but the difference is not very significant and implementation takes time. There is always a gain in accuracy with high frame rates but the difference is not very significant and implementation takes time. Hence, based on the specifics of the application, it

may be more advantageous to accept a frame rate of 30 FPS since the video default is set to 30 FPS. Thus, 30 FPS can be focused more on computational speed even if marginal improvements in accuracy can be achieved.

#### 4.4.2 Processing Time of Histogram Equalization Across Different Video Lengths

This section examines the findings of the evaluation of the amount of time taken to perform histogram equalization on videos across different lengths of videos and under different lighting conditions. The analysis focuses on two specific scenarios: a very dark environment (Video 1) and a dark environment (Video 2). The amount of time taken for each processing was recorded and the result was graphed as shown in Figures 4.11 and 4.12.

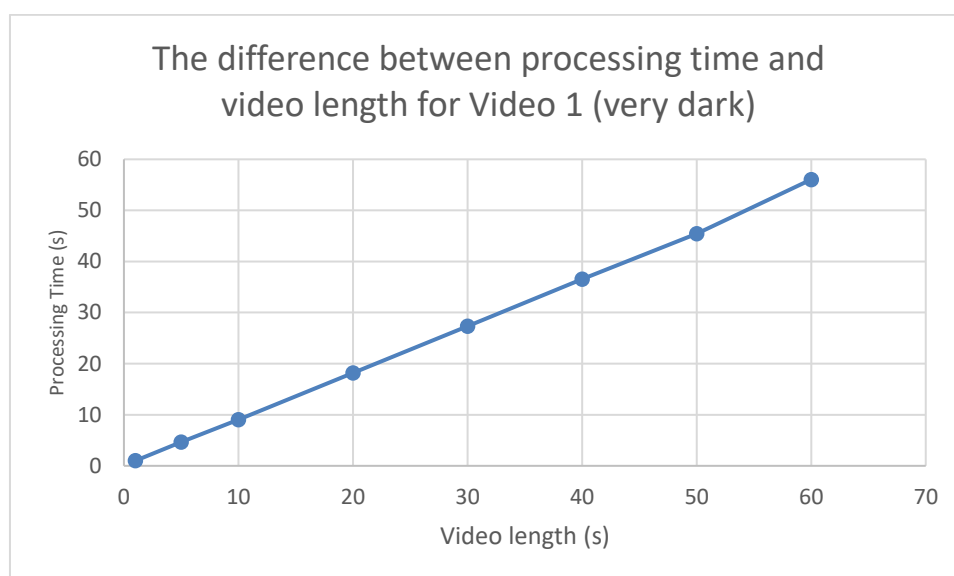


Figure 4.11 Processing Time vs. Video Length for Video 1 (Very Dark)



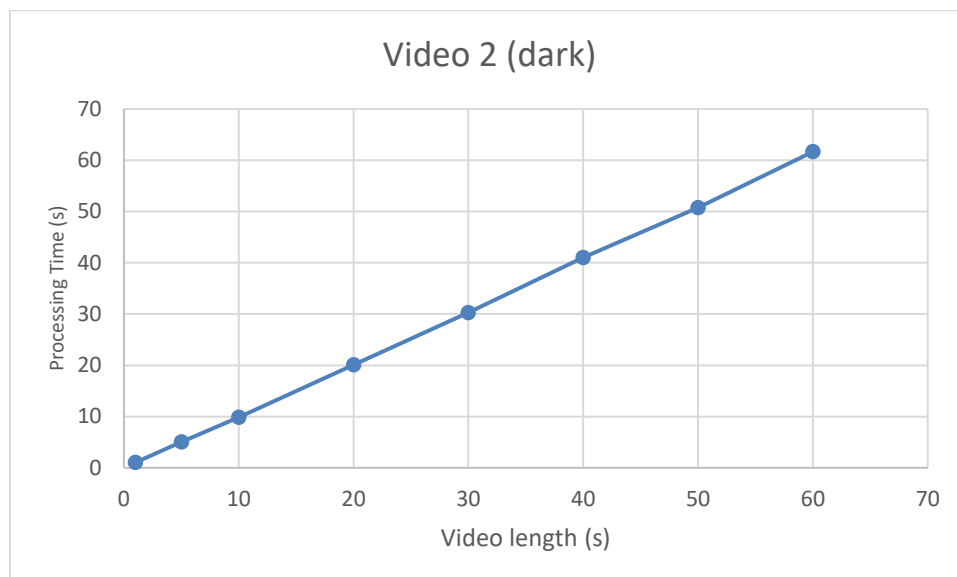


Figure 4.12 Processing Time vs. Video Length for Video 2 (Dark)

The results shown by Figures 4.11 and 4.12 prove that there is a direct positive correlation between the video length and the amount of time it takes to process videos in very dark (Video 1) and dark (Video 2). Therefore, the processing time training was consistent throughout Video 1, ranging from 1 second for the 1-second video to 56.05 seconds for a 60-second video. Video 2 also had a linear pattern but the processing took slightly longer than Video 1 ranging from 1.04 seconds for a 1-second video to 61.66 seconds for a 60 seconds video. This indicates that the relationship between histogram equalization's processing time and video length is directly proportional. Hence, is it preferred to choose which video length is suitable for particular needs.

#### 4.4.3 Performance Improvement Through Reduction of Video Resolution

To enhance the effectiveness in processing histogram equalization for the frames of the video, a resolution reduction technique was implemented. This technique was used to be able to always maintain the accuracy of the histogram equalization while at the same time reducing the computation time required.

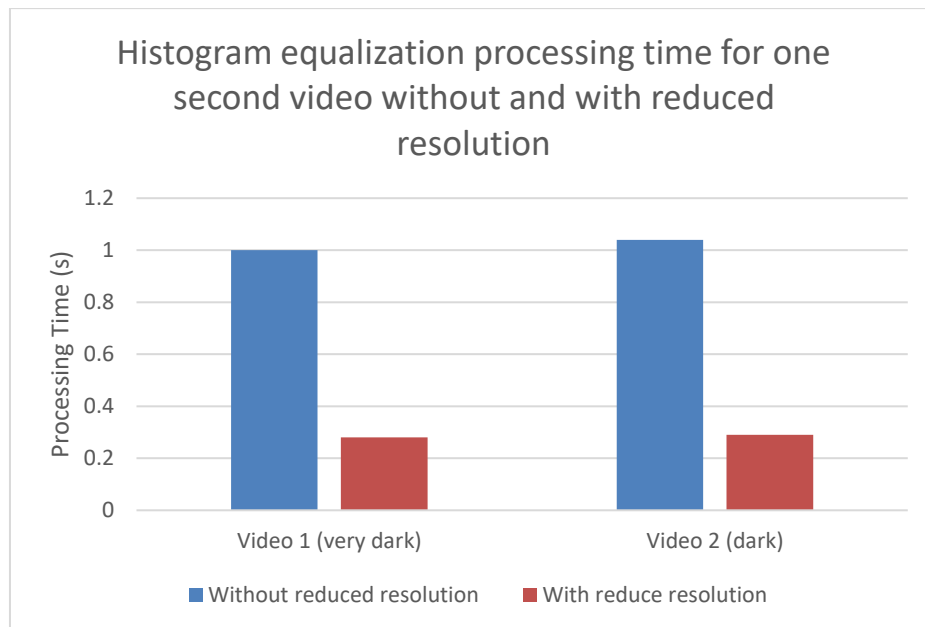


Figure 4.13 Histogram Equalization Processing Time for One-Second Videos under Different Lighting Conditions

As can be seen in Figure 4.13, the overall processing time for the histogram equalization was greatly decreased once the resolution had been lowered. Indeed, regarding a one-second video, the time for processing dropped by roughly 70% if the resolution was decreased. It seems that this effect did not vary with different video conditions such as very dark and dark videos.

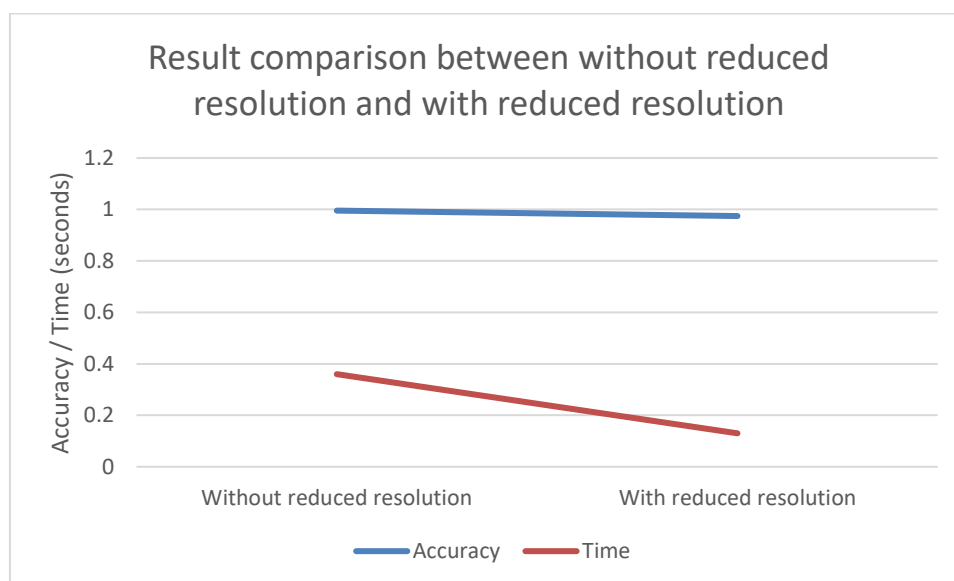


Figure 4.14 Comparison of Accuracy and Histogram Equalization Processing Time with and without Reduced Video Resolution

From the above Figure 4.14, the accuracy of the base model has slightly changed when the resolution was reduced. The slight decrease in accuracy did not adversely affect the positive results of the reduced processing time. This proved that accuracy was a stable parameter near 1 at the end of the prediction in both cases, confirming that the loss of detail associated with lowering the resolution did not affect the model performance when properly identifying and categorizing the content of the video input.

## CHAPTER 5

### CONCLUSIONS AND RECOMMENDATIONS

#### 5.1 Conclusions

In this research, an approach to violence detection in the surveillance video has been created and tested using deep learning methodology, particularly addressing the problem of lighting condition variability. The baseline model, which involved a MobileNet-v2 network model for feature extraction, followed by Bidirectional Long Short-Term Memory (BiLSTM) for classification achieved a low accuracy when used to predict violent behavior in very dark conditions, as it only achieved 24.89% accuracy. To counter this, histogram equalization increases the visibility of a very dark test video during the video prediction phase thus improving the prediction accuracy to 92.21% regardless of the need to retrain the model.

In addition to this, another set of video data that contained videos that were recorded under different lighting conditions was used to fine-tune the base model. This approach also enhanced performance to 99.97%, especially during the very dark environment but had the drawback of increased training time as compared to the previous methods and even required to find additional datasets with varying light conditions which is quite challenging. It was found that, while the base model greatly failed in detecting objects from dark videos, both the histogram equalization and the additional dataset methods significantly engineered enhanced detection rates under such circumstances.

In conclusion, the project successfully demonstrated that applying histogram equalization to test videos can enhance the performance of a pre-trained model in low-light scenarios. Additionally, retraining the model with diverse lighting conditions also proved effective, although it required greater investment in training time and challenges in dataset searching. These findings provide valuable insights into optimizing violence detection models for real-world surveillance applications, particularly in environments with varying lighting conditions.

## 5.2 Recommendations for future work

Based on the research conducted towards the development of this concept, some suggestions for future courses of action are suggested below. First, although the proposed application of the histogram equalization provided a highly appreciated degree of the model's performance boost in situations with low lighting, it is crucial to consider other possible image enhancement approaches used under various conditions. Other methods of image processing as changes of contrast, removing noise, or edge sharpening could be applied to different types of difficult conditions to improve the overall performance of the model in various conditions.

Secondly, future work may investigate how the model performs under other effective lighting conditions, the influence of environmental noise, motion blur, or partial occlusions. Solving these problems would make the adaption of the model more suitable for real world surveillance applications.

Lastly, there is an opportunity for future development of this research through the implementation of the proposed model into an end-to-end surveillance system, as well as, assessing its feasibility in real-world applications. This could confirm the efficiency of the model in practice and identify the improvements required to enhance the process of violence detection under various conditions.

## REFERENCES

- Abdali, A.-M.R. and Al-Tuma, R.F., 2019. Robust Real-Time Violence Detection in Video Using CNN And LSTM. *2019 2nd Scientific Conference of Computer Sciences (SCCS)*. March 2019 IEEE, pp. 104–108.
- Alzubaidi, L. et al., 2021a. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), p.53.
- Aremu, T. et al., 2024. SSIVD-Net: A Novel Salient Super Image Classification and Detection Technique for Weaponized Violence. In: pp. 16–35.
- Arias, F. et al., 2022. Sentiment Analysis of Public Social Media as a Tool for Health-Related Topics. *IEEE Access*, 10, pp.74850–74872.
- Aziz Sharfuddin, A., Nafis Tihami, Md. and Saiful Islam, Md., 2018. A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification. *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. September 2018 IEEE, pp. 1–4.
- Bhatt, D. et al., 2021. CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics*, 10(20), p.2470.
- Bianchi, F.M. et al., 2017. An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting.
- Bolboacă, R. and Haller, P., 2023. Performance Analysis of Long Short-Term Memory Predictive Neural Networks on Time Series Data. *Mathematics*, 11(6), p.1432.
- Cun, L. et al., 1989. *Handwritten Digit Recognition with a Back-Propagation Network*, Neural Information Processing Systems.
- Du, X., Cai, Y., Wang, S. and Zhang, L., 2016. Overview of deep learning. *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. November 2016 IEEE, pp. 159–164.
- Emanuel, R.H.K., Docherty, P.D., Lunt, H. and Möller, K., 2024. The effect of activation functions on accuracy, convergence speed, and misclassification confidence in CNN text classification: a comprehensive exploration. *The Journal of Supercomputing*, 80(1), pp.292–312.
- Es-Sabery, F. et al., 2021. Sentence-Level Classification Using Parallel Fuzzy Deep Learning Classifier. *IEEE Access*, 9, pp.17943–17985.
- Evidently AI Team, *How to explain the ROC AUC score and ROC curve?* [Online]. Available at: <https://www.evidentlyai.com/classification-metrics/explain-roc-curve> [Accessed: 21 August 2024].

Feng, J., Liang, Y. and Li, L., 2021. Anomaly Detection in Videos Using Two-Stream Autoencoder with Post Hoc Interpretability. *Computational Intelligence and Neuroscience*, 2021, pp.1–15.

Gichoya, J.W. et al., 2022. AI recognition of patient race in medical imaging: a modelling study. *The Lancet Digital Health*, 4(6), pp.e406–e414.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep Learning*, MIT Press.

Greff, K. et al., 2015. LSTM: A Search Space Odyssey.

Gu, J. et al., 2015. Recent Advances in Convolutional Neural Networks. Available at: <http://arxiv.org/abs/1512.07108>.

Gunawardena, N., Ginige, J.A., Javadi, B. and Lui, G., 2022. Performance Analysis of CNN Models for Mobile Device Eye Tracking with Edge Computing. *Procedia Computer Science*, 207, pp.2291–2300.

Halder, R. and Chatterjee, R., 2020. CNN-BiLSTM Model for Violence Detection in Smart Surveillance. *SN Computer Science*, 1(4), p.201.

Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786), pp.504–507.

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735–1780.

Van Houdt, G., Mosquera, C. and Nápoles, G., 2020. A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8), pp.5929–5955.

Howard, A.G. et al., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Hubel, D.H. and Wiesel, A.T.N., 1962. *54 With 2 plate and 20 text-figures* Printed in Great Britain *RECEPTIVE FIELDS, BINOCULAR INTERACTION AND FUNCTIONAL ARCHITECTURE IN THE CAT'S VISUAL CORTEX*.

Iandola, F.N. et al., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size.

James, G. (Gareth M., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An introduction to statistical learning : with applications in R*, Springer.

Khandelwal, S., 2024, *Selangor's Soaring Crime Rate: A Glimpse into the 2023 Surge in Violent, Narcotic, and Commercial Crimes* [Online]. Available at: <https://bnnbreaking.com/world/malaysia/selangors-soaring-crime-rate-a-glimpse-into-the-2023-surge-in-violent-narcotic-and-commercial-crimes> [Accessed: 3 March 2024].

Koutník, J., Greff, K., Gomez, F. and Schmidhuber, J., 2014. A Clockwork RNN.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), pp.84–90.

Li, J. et al., 2024. Medical image identification methods: A review. *Computers in Biology and Medicine*, 169, p.107777.

Maass, W., Joshi, P. and Sontag, E.D., 2007. Computational Aspects of Feedback in Neural Circuits. *PLoS Computational Biology*, 3(1), p.e165.

Montesinos López, O.A., Montesinos López, A. and Crossa, J., 2022. Fundamentals of Artificial Neural Networks and Deep Learning. In: *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Springer International Publishing, Cham, pp. 379–425.

Prasad, Mr.P.S. and Senthilrajan, Dr.A., 2021. Leaf Features Extraction for Plant Classification using CNN. *International Journal of Advanced Research in Science, Communication and Technology*, pp.148–154.

Prudhomme, C. et al., 2023. *Deep Learning Datasets Challenges For Semantic Segmentation-A Survey*.

Radiuk, P.M., 2017. Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets. *Information Technology and Management Science*, 20(1).

Sandler, M. et al., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks.

Schuster, M. and Paliwal, K.K., 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), pp.2673–2681.

Soliman, M.M. et al., 2019. Violence Recognition from Videos using Deep Learning Techniques. *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*. December 2019 IEEE, pp. 80–85.

Su, Y., Lin, G., Zhu, J. and Wu, Q., 2020. Human Interaction Learning on 3D Skeleton Point Clouds for Video Violence Recognition. In: pp. 74–90.

Tiwari, R.G., Maheshwari, H., Agarwal, A.K. and Jain, V., 2023. Hybrid CNN-LSTM Model for Automated Violence Detection and Classification in Surveillance Systems. *2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART)*. 22 December 2023 IEEE, pp. 169–175.



Tran, D. et al., 2017. A Closer Look at Spatiotemporal Convolutions for Action Recognition. Available at: <http://arxiv.org/abs/1711.11248>.

Traore, A. and Akhloufi, M.A., 2020. Violence Detection in Videos using Deep Recurrent and Convolutional Neural Networks. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 11 October 2020 IEEE, pp. 154–159.

Vaswani, A. et al., 2017. Attention Is All You Need.

Vieira, J.C. et al., 2022. Low-Cost CNN for Automatic Violence Recognition on Embedded System. *IEEE Access*, 10, pp.25190–25202.

Yin, L., 2018, *A Summary of Neural Network Layers* [Online]. Available at: <https://medium.com/machine-learning-for-li/different-convolutional-layers-43dc146f4d0e> [Accessed: 28 March 2024].

Zhao, Y. et al., 2018. Applying deep bidirectional LSTM and mixture density network for basketball trajectory prediction. *Optik*, 158, pp.266–272.

Zoph, B., Vasudevan, V., Shlens, J. and Le, Q. V., 2018. Learning Transferable Architectures for Scalable Image Recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018 IEEE, pp. 8697–8710.

## APPENDICES

### Appendix A: Accuracy Comparison Across Three Models for Videos Categorized into Four Categories Tables

Model	Accuracy			
	Normal	Dark	Very Dark	Night Vision
Base Model (Mobile MobileNet-v2 and BiLSTM)	0.9633	0.8750	0.2489	0.0625
Model trained on videos with varying levels of darkness	0.9915	0.9934	0.9987	0.9699
Base Model + Histogram Equalization	0.9891	0.9442	0.9221	0.9957