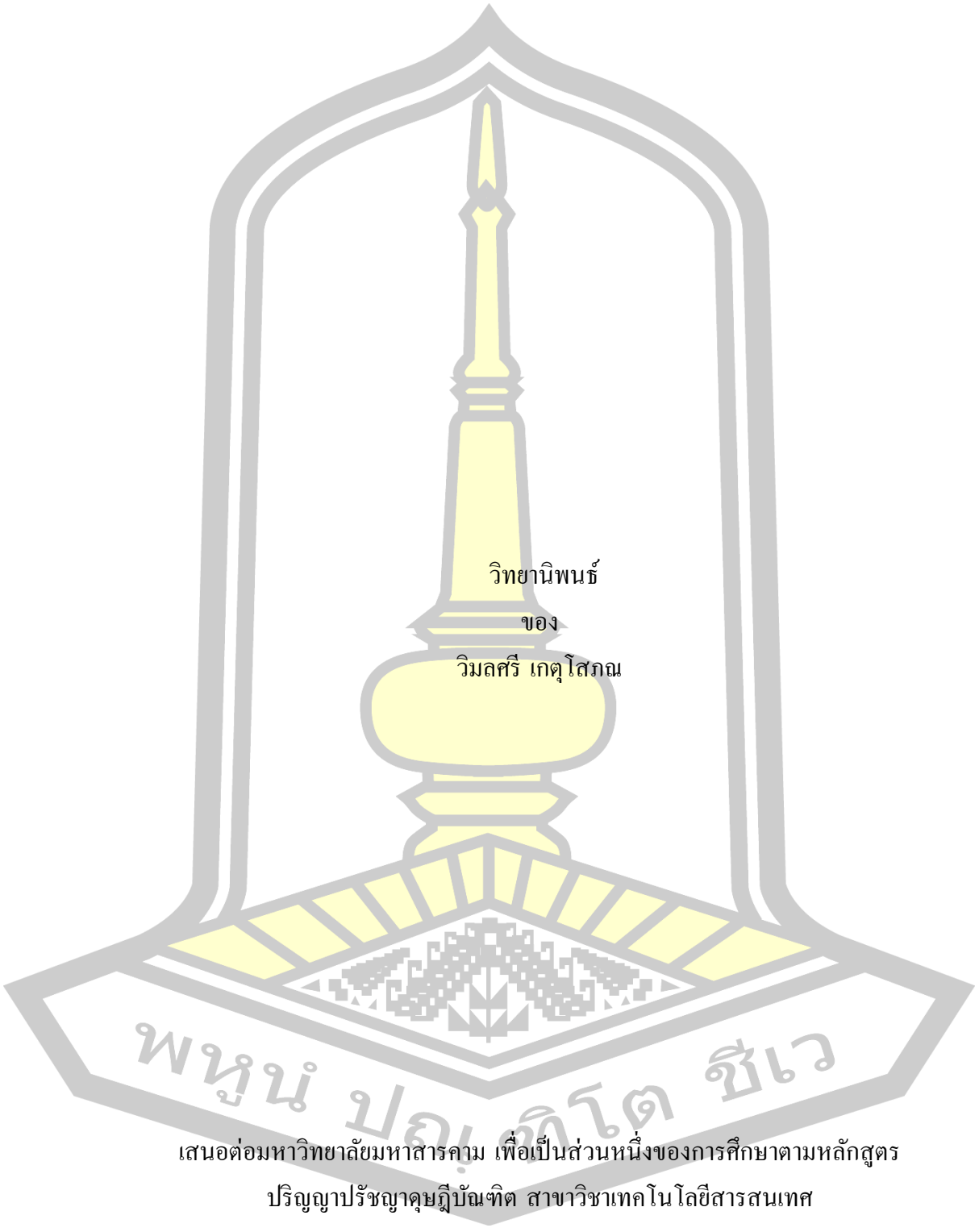
Deep Learning for Understanding Violence in Videos

Wimolsree Getsopon

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Information Technology
January 2023

การเรียนรู้เชิงลึกสำหรับการเข้าใจความรุนแรงในวิดีโอ



วิทยานิพนธ์

ของ

วิมลศรี เกตุโสภณ

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

มกราคม 2566

Deep Learning for Understanding Violence in Videos

Wimolsree Getsopon

A Thesis Submitted in Partial Fulfillment of Requirements

for Doctor of Philosophy (Information Technology)

January 2023

The examining committee has unanimously approved this Thesis, submitted by Miss Wimolsree Getsopon , as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology at Mahasarakham University

Examining Committee

.....Chairman
(Prof. Rapeepan Pitakaso , Ph.D.)

.....Advisor
(Asst. Prof. Olarik Surinta , Ph.D.)

.....Committee
(Asst. Prof. Rapeeporn Chamchong , Ph.D.)

.....Committee
(Asst. Prof. Phatthanaphong Chompoowises , Ph.D.)

.....Committee
(Asst. Prof. Chatklaw Jareanpon , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology

.....
(Assoc. Prof. Jantima Polpinij , Ph.D.)
Dean of The Faculty of Informatics

.....
(Assoc. Prof. Krit Chaimoon , Ph.D.)
Dean of Graduate School

TITLE Deep Learning for Understanding Violence in Videos

AUTHOR Wimolsree Getsopon

ADVISORS Assistant Professor Olarik Surinta , Ph.D.

DEGREE Doctor of Philosophy **MAJOR** Information Technology

UNIVERSITY Mahasarakham University **YEAR** 2023

ABSTRACT

Chapter 1 briefly introduces violent video understanding and research questions. Additionally, the objectives of the dissertation and contributions are described.

Chapter 2 describes a background of violent video understanding using deep learning techniques and related work. The background includes deep learning techniques, convolution neural networks, convolution neural network architecture, 3D Convolutional Neural Networks (3D-CNN), Recurrent Neural Networks (RNN), Deep feature extraction, deep feature fusion methods, and violent video datasets. Next, a related work section, which has reviewed research from the past until now, consists of six main parts as follows: deep learning for video classification, handcrafted features for violent recognition, violent recognition with 2D-CNN, violent recognition with 3D-CNN, violent recognition with combination of CNN and RNN, and violent recognition with fusion features.

Chapter 3 proposed a fusion MobileNets-BiLSTM architecture. In the first part, I proposed using the lightweight MobileNetV1 and MobileNetV2 to extract the robust deep spatial features from the video so that only 16 non-adjacent frames were selected. The spatial features were transferred to the global average pooling, batch normalization, and time distribution layer. In the second part, the spatial features from the first part were concatenated and then transferred to a Bidirectional Long Short-Term Memory (BiLSTM). The proposed fusion MobileNets-BiLSTM architecture was evaluated on the hockey fight dataset. The experimental results showed that the proposed method achieved 95.20% accuracy on the test set of the hockey fight dataset.

Chapter 4 proposed a method to understand violence within video using deep feature integration with 3D-CNN. I proposed CNN to extract the spatial feature from the last convolution layer at the frame level. The concatenate operation was proposed to combine the spatial features of both CNNs at the frame level before being transferred to the 3D-CNN architecture to learn the spatiotemporal features, consisting of batch normalization, 3D convolution, dropout layers, global average pooling layer followed by a fully connected layer. Finally, the softmax was used to classify as a violent and non-violent video.

Chapter 5 comprises two main sections: the answers to the research questions and suggestions for future work. This chapter briefly explains the proposed approaches and answers two main research questions in video understanding.

Keyword : Violent Video Understanding, Violent Video Recognition, Video Recognition, Convolutional Neural Network, Recurrent Neural Network, Feature Extraction, Features Fusion Technique

ACKNOWLEDGEMENTS

I would like to thank my esteemed supervisor - Assistant Professor Olarik Surinta, Ph.D. for his invaluable supervisor, support, and tutelage during the course of my Ph.D. degree. My gratitude extends to the Faculty of Business Administration and Information Technology, Rajamangkala University of Technology ISAN Khon Kaen Campus for the funding opportunity to undertake my studies at the Faculty of Informatics, Mahasarakham University. Additionally, I would like to thank my friend, lab mates, colleagues, and researcher team for a cherished time spent together in the lab, and in social setting. My appreciation also goes out to my family for their encouragement and support throughout my studies.

Wimolsree Getsopon

TABLE OF CONTENTS

Page

ABSTRACT ..... D

ACKNOWLEDGEMENTS ..... F

TABLE OF CONTENTS ..... G

List of Table I

List of Figure ..... K

Chapter 1 1

Introduction 1

1.1 Research questions 3

1.2 The objective of this dissertation 4

1.3 Contribution 5

Chapter 2 7

Background 7

2.1 Deep Learning 7

2.2 Convolutional Neural Network 8

2.3 Convolutional neural network architecture 11

2.4 3D Convolution (Ji et al., 2013) 14

2.5 Recurrent Neural Network architecture 15

2.6 Deep features extraction 18

2.7 Deep features fusion method 19

2.8 Violent dataset 20

2.9 Related work 24

Chapter 3 32

Fusion Lightweight CNNs and Sequence Learning Technique 32

3.1 Introduction 32

3.2 Related work 33

3.3 Fusion Lightweight CNNs and Sequence Learning Architecture.....35
3.4 Violent Video Datasets.....38
3.5 Experiment Setup and Results.....39
3.6 Comparison of the Fusion MobileNets and BiLSTM architecture and the Existing Methods.....44
3.7 Conclusions.....44
Chapter 4.....46
Violence recognition with 3D-CNN.....46
4.1 Introduction.....46
4.2 Related work.....48
4.3 Proposed framework.....54
4.4 Violence Datasets.....60
4.5 Experimental results and discussion.....62
4.6 Comparison.....85
4.7 Conclusions.....87
Chapter 5.....89
Discussion.....89
5.1 Answers to The Research Questions.....91
5.2 Future work.....93
REFERENCES.....95
BIOGRAPHY.....103

List of Table

Page

Table 1 A number of videos in each category of UCF-crime dataset. 23

Table 2 A number of parameter of CNN architectures. 37

Table 3 Experimental results with different frames using MobileNetV2-LSTM. 40

Table 4 The average accuracy (%) and the standard deviation of CNN architectures combined with the LSTM network obtained on cross-validation and test sets. 42

Table 5 The accuracy (%) and computational times of violence recognition experiments on the hockey fight dataset. 43

Table 6 The comparison of the proposed method with existing methods. 44

Table 7 The number of the parameters in all layers of C3D architecture. 58

Table 8 Network architecture of the proposed 3D convolutional neural network. 59

Table 9 Evaluation of the violent recognition results using MobileNetV1 64

Table 10 Evaluation of the violent recognition results using MobileNetV2 65

Table 11 Testing accuracy of feature-extraction with the MobileNetV1, trained with merging all datasets and testing with separate datasets. 66

Table 12 Testing accuracy of feature-extraction with the MobileNetV2, trained with merging all datasets and testing with separate datasets 67

Table 13 The accuracy results with C3D on three datasets. 69

Table 14 The five-difference 3D convolution structures. 70

Table 15 Performance of the 3D convolution with integrated deep features on hockey fight dataset. 71

Table 16 Performance of the 3D convolution with integrated deep features 71

Table 17 Performance of the 3D convolution with integrated deep features on violent flow dataset. 72

Table 18 Performance of the 3D convolution with integrated deep features(merged all datasets) on the hockey fight dataset. 73

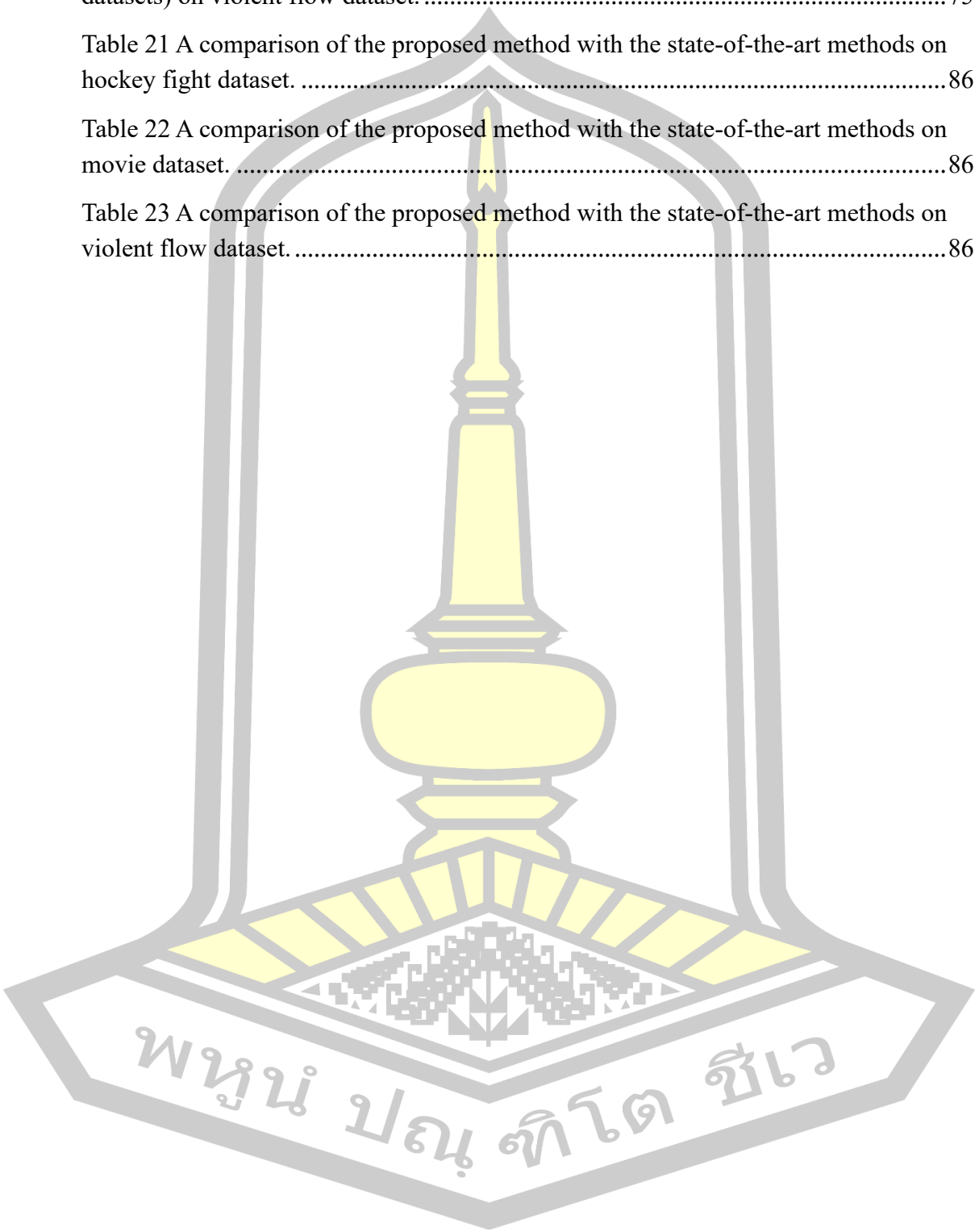
Table 19 Performance of the 3D convolution with integrated deep features(merged all datasets) on movie dataset. 74

Table 20 Performance of the 3D convolution with integrated deep features (merged all datasets) on violent flow dataset. 75

Table 21 A comparison of the proposed method with the state-of-the-art methods on hockey fight dataset. 86

Table 22 A comparison of the proposed method with the state-of-the-art methods on movie dataset. 86

Table 23 A comparison of the proposed method with the state-of-the-art methods on violent flow dataset. 86



List of Figure

Page

Figure 1 An example of a deep learning network. 7

Figure 2 Illustrate of the Convolutional Neural Network structure. 8

Figure 3 The example of a convolution operator. 9

Figure 4 Example of a convolution operator with padding. 9

Figure 5 Illustration of max pooling. 10

Figure 6 Illustration of average pooling layer. 11

Figure 7 Illustrate of the depthwise separable convolution (Howard et al., 2017). 12

Figure 8 The structural of MobileNetV2 (Sandler et al., 2018). 12

Figure 9 The structural of MobileNetV2 (Sandler et al., 2018). 13

Figure 10 illustrates the ResNet50V2 architecture (He et al., 2016). 14

Figure 11 The structure of 3D convolution (Ji et al., 2013). 14

Figure 12 The architecture of C3D (Tran et al., 2014). 15

Figure 13 Recurrent Neural Network. 16

Figure 14 Long Short - Term Memory architecture (Hochreiter & Schmidhuber, 1997). 16

Figure 15 Long Short - Term Memory cell (Hochreiter & Schmidhuber, 1997). 17

Figure 16. Bidirectional Long Short-Term Memory architecture (Graves & Schmidhuber, 2005). 17

Figure 17 Gate Recurrent Unit architecture. (Cho et al., 2014) 18

Figure 18 Deep feature extraction using CNN. 19

Figure 19. Feature fusion using addition operation. 20

Figure 20. Feature fusion using concatenation operation. 20

Figure 21 Samples of hockey fight dataset. 21

Figure 22 Samples of movie dataset. 22

Figure 23 Samples of violent flow dataset. 22

Figure 24 Sample of RWF2000 dataset. 23

Figure 25 Samples of UCF crime dataset. 24

Figure 26 Illustration of the fusion lightweight MobileNets and BiLSTM architecture for violence video recognition. 35

Figure 27 Some examples of (a) violent video and (b) non-violent video of.....39

Figure 28 Illustration of the (a) adjacent and (b) non-adjacent frames of 41

Figure 29 The proposed framework deep features integration with 3D convolutional to recognize the violent video. 54

Figure 30 show (a) the standard convolution kernel (b) the depthwise convolution kernel and (c) pointwise convolution kernel. (Howard et al., 2017) 56

Figure 31 The structural of CNN (a) MobileNetV1 and (b) MobileNetV2 (Howard et al., 2017; Sandler et al., 2018) 57

Figure 32 The architecture of C3D (Tran et al., 2014). 58

Figure 33 Samples of hockey fight dataset, (a) violence video and 61

Figure 34 Samples of movie dataset, (a) violence video and (b) non-violence video. 61

Figure 35 A samples of violent flow dataset, (a)violence and (b) nonviolence video. 62

Figure 36 Receiver operating characteristic (ROC) curve and area under the curve (AUC) for each 3D-CNN model (a) on the hockey fight dataset, (b) on the movie dataset, and (c) on violence flow dataset. 77

Figure 37 Precision recall curve and area under the precision recall curve (AUC-PR) for each 3D-CNN model (a) on hockey fight dataset, (b) on movie dataset, and (c) on violent flow dataset. 78

Figure 38 Training and validation loss of our proposed model on the (a) hockey fight dataset, (b) movie dataset, and (c) violent flow dataset. 79

Figure 39 The confusion matric of test datasets (a) on hockey fight dataset, (b) movie dataset, and (c) violent flow dataset. 80

Figure 40 Example of missing video prediction on hockey fight dataset, (a) false negative prediction and (b) false positive prediction. 81

Figure 41 Example of missing video prediction on violent flow dataset, ...82

Figure 42 Example of violent video with different camera angles: (a) very long shot, (b) medium-close-up shot, and (c) close-up shot. 85

Chapter 1
Introduction

The advancement of artificial intelligence technology has been developing rapidly and has been widely applied in various tasks recently. These include health services, industrial security, surveillance, and assistance systems for individuals with disabilities. Computer vision is a subsection of artificial intelligence using techniques that enable computer systems to understand and respond to images or videos in a way similar to the human visual system. Illustrations of computer vision technology utilization within health service systems.-for example, monitoring the daily activities that pose a risk of accidents for elderly individuals. With the system sending an alert notifying the involved authorities when an older adult is injured from falls or exhibits symptoms similar to a stroke (Attal et al., 2015; Liu et al., 2016). Furthermore, computer vision helps to collect information about daily behavior that affects individual health (Suryadevara & Mukhopadhyay, 2014). For industrial applications, robots can replace humans to perform tasks that humans cannot perform or to work in hazardous areas (Dallel et al., 2020). In surveillance systems, applying computer vision involves identifying unusual incidents like criminal activities, acts of violence, theft, and accidents. The incidents can happen anywhere, from residential areas and educational institutions to roads, parking lots, bus terminals, and commercial establishments such as shopping centers. Computer vision technology ensures that abnormal events can be detected and classified efficiently. Subsequently, the surveillance system can notify the relevant individual to stop the incident in real time. Furthermore, computer vision technology can also be applied to assist individuals with disabilities (Wu et al., 2017), serving as an aiding tool or creating equal opportunities for accessing information and data.

Video understanding is a significant component within the field of computer vision that has garnered considerable attention. It enables computer systems to understand and analyze meaningful video information or patterns. Video understanding is applied to various tasks, for instance, surveillance systems, video annotation, video recommendation, video search, and related video retrieval (Lee et al., 2018). Many researchers have proposed a practical approach for effectively

analyzing and processing video for applying various tasks (Xie et al., 2017; B. Zhou et al., 2017; Zolfaghari et al., 2018). This thesis focuses on violent video classification using deep learning techniques. Violent behavior refers to aggressive behaviors such as fighting, smashing, riots, and collisions (Yao & Hu, 2023).

Violent video recognition is a subfield of action recognition since recognizing violent behavior in video data is to understand some human actions. Surveillance systems are installed in public and private areas to monitor, collect evidence, and prevent criminal activities. However, manually monitoring and analyzing video data from many CCTV cameras in real time can be costly and time-consuming. Therefore, utilizing machine learning technology for automated crime scene recognition from the video is very important and assists security systems in detecting and categorizing various anomalies or violent occurrences, alerting, and notifying the security monitoring system to respond immediately.

Video violence classification typically consists of two main parts, feature extraction and classification sections. In the past, many researchers tried to develop efficient techniques for video feature extraction. Handcraft feature extraction is a valuable method for video recognition that considers local features. Souza et al. (2010) proposed feature extraction, namely Local spatiotemporal features, using a bag of words. Das et al. (2019) used Histogram Orientation Gradient (HOG). Hassner et al. (2012) and Gao et al. (2016) improved the feature extraction method by considering the optical flow direction in global feature extraction. Classification is a technique that involves classifying data into predefined types. Machine learning for classification, can use Support Vector Machine (SVM), Random Forest, Logistic Regression, and Neural Networks. Although research on surveillance video recognition has presented various handcraft feature extractions, these methods have some limitations. Examples of constraints include the effectiveness of handcrafted features developed from specific datasets, resulting in the incapability to extract appropriate video representation and failure to achieve a generalized model.

Many researchers have recently demonstrated deep learning efficiency as a feature extraction method. Specifically, a convolution neural network (CNN) was developed for image or video recognition. This is a high-performance network that can be applied to various tasks. For video recognition work, some researchers have

used 2D-CNN for spatial feature extraction of each frame and then classified the features with different methods. For example, Carneiro et al. (2019)and Soliman et al. (2019) used VGG-16 for feature extraction. For learning about time information, recurrent neural networks (RNN) are also used combined with CNNs to improve recognition performance, such as LSTM or GRU. Sudhakaran and Lanz (2017) used 2D-CNN to extract hierarchical features from the video frames, which were then aggregated using the LSTM. Then, they were classified as violent or non-violent with a fully connected layer. Mumtaz et al. (2022) proposed a multi-scale of VGG-19 architecture for violence video classification. The VGG-19 was used to initialize the spatial features extractor, followed by the widely followed Bi-LSTM structure for optimal violent recognition. 3D-CNN is proposed for end-to-end networks, learning spatial and temporal information. Ji et al. (2013) proposed a 3D-CNN to extract spatial and temporal features from video data for action recognition. The experimental results show that the proposed models significantly outperformed 2D-CNN architecture.

In addition, some research has uses deep learning to learn from different feature types. Lou et al. (2021) employed frame and audio information to recognize violent behavior. Carneiro et al. used VGG-16 for a multi-stream that included spatial, temporal, rhythm, and depth information. The results showed that the feature fusion method increased the efficiency of violent video recognition.

1.1 Research questions

An essential component of recognizing violence in a video task is understanding the information to learn and effectively analyze the content within videos. This process involves extracting significant information, recognizing patterns, and understanding the visual information in a video. An effective video violence recognition system has the potential to automate and accurately classify representatives of violence, particularly within security surveillance systems, to stop the violence in time and prevent further violence. Therefore, I aim to improve the efficiency of violence recognition in videos to enhance the capabilities of security surveillance systems.

RQ1. Generally, violent video understanding applies Recurrent Neural Networks (RNN) such as LSTM, BiLSTM, or GRU to learn the feature from sequential frames within the video data. RNN can distinguish patterns and movements, accurately classifying actions, or activities in a video. However, some research has used Convolutional Neural Networks (CNN) to extract deep features from the individual frame, which received high accuracy for violence recognition Karisma et al. (2021) and Irfanullah et al. (2022). Therefore, if I utilize CNN to extract the deep features from video frames and then transfer the received deep features to RNN to learn information within the video, will this improve the performance of understanding violent videos?

RQ2. The 2D-CNN outperforms in extracting spatial features within individual frames, making it well-suited for tasks where static visual patterns hold pivotal significance, such as image classification and object detection. Conversely, 3D-CNN surpasses 2D-CNN in tasks requiring the incorporation of necessary temporal dimensions, as it can directly understand spatiotemporal features from video sequences. This renders 3D-CNN notably advantageous for applications like action recognition, wherein comprehending temporal alterations and motion is imperative. Although 2D-CNN demonstrates computational efficiency and is commonly employed for image-based tasks, 3D-CNN extends its functionalities to video analysis by seamlessly incorporating temporal information into the learning process. Therefore, if 2D-CNN are used to extract spatial features from frames and integrate the obtained features, the features are then transferred to 3D-CNN for spatiotemporal learning and classified into violent or nonviolent videos. Can the proposed approach improve the performance of violent video recognition?

To answer all these questions (RQ1 and RQ2), Chapter 3 and Chapter 4 of this thesis describe the result of this research. Finally, Chapter 5 provides concrete answers to research questions.

1.2 The objective of this dissertation

This study will focus on two detailed objectives:

1.2.1 Improve violent video recognition by combining CNN and RNN with deep feature fusion techniques.

1.2.2 Improve violent video recognition by deep feature integration with three-dimensional convolution neural network (3D-CNN).

1.3 Contribution

The contribution of the dissertation is a novel deep learning technique to extract robust features and provide the best performance for violent video recognition system. The work involved experiments on three benchmark violent video datasets of hockey fight, movie, and violent flow datasets. The contributions of the dissertation are as follows.

In Chapter 3, I introduced the utilization of MobileNets to extract robust spatial features; MobileNet has few parameters and a small model size but is still highly accurate. Additionally, I employed a bidirectional long short-term memory (BiLSTM) to understand the temporal context and acquire information from past and future video frames. This approach incorporated a concatenation operation to combine spatial features obtained from MobileNetV1 and MobileNetV2 before being transferred to the BiLSTM network. Furthermore, the classifier for the proposed architecture was implemented using the softmax function. Hence, we opted for a selection of 16 non-adjacent frames, although alternative methods were assessed using 20 and 40 frames. The resulting output was categorized as violent and non-violent. This chapter is based on the following publication.-

Wimolsree Getsopon and Surinta (2022). Fusion Lightweight Convolutional Neural Networks and Sequence Learning Architectures for Violence Classification. ICIC Express Letter Part B: Applications, 13(10), pages 1027-1035.

In Chapter 4, I propose an approach for recognizing violent video content, employing deep feature integration with three-dimensional convolution. I focus on conducting feature extraction at the frame level utilizing two distinct Convolutional Neural Network (CNN) models, specifically MobileNetV1 and MobileNetV2. These models were applied at the last convolutional layer to extract robust spatial features. Subsequently, I executed feature vector integration through concatenation operations. The integrated video feature vector was subjected to a three-dimensional convolutional process, allowing the model to capture temporal dependencies within

the video data. Finally, the classification of videos as either violent or non-violent was carried out through a softmax function, enabling the model to make informed categorizations based on the learned features. To assess the efficacy of this proposed method, I evaluated three challenging video datasets encompassing violent scenarios, namely hockey fight, movie, and violent flow. These evaluations validated the robustness and effectiveness of this approach in the domain of violent video recognition.

Chapter 2
Background

2.1 Deep Learning

Deep learning is a subset of machine learning that utilizes neural networks to solve complex problems (Sharifani & Amini, 2023). The basis of deep learning is inspired by human brain function. Deep learning can learn from sample data and train model knowledge by automatically recognizing patterns or classifying data. Then, it provides an answer by predicting the probability value that simple artificial intelligence techniques cannot extract to correctly infer conclusions from the data.

The structure of the deep learning network comprises input, hidden, and output layers. The network has multiple hidden layers, and the hidden layers are composed of several neurons. The primary function of a neuron is to multiply the input values with the assigned weight generated randomly at the beginning of model training, sum up the result, and add bias. Then, the results were adjusted with an activation function such as sigmoid, Tanh, or RELU to get a value between 0 and 1 (Mercioni & Holban, 2023), as shown in Figure 1.

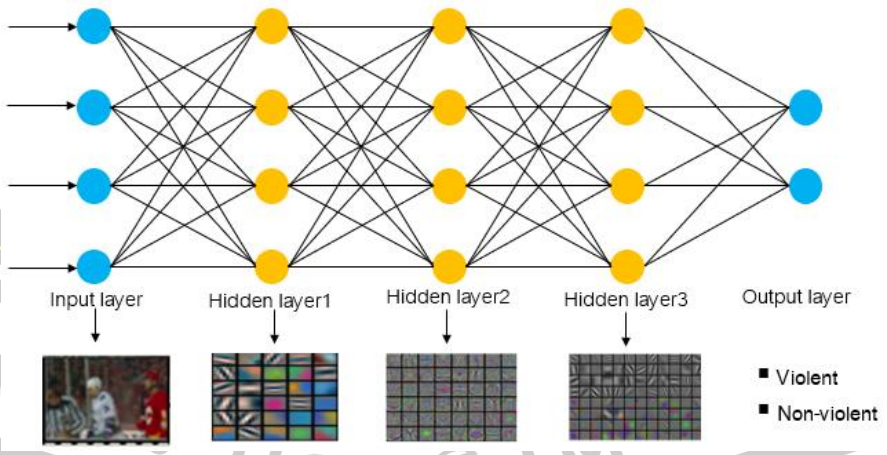


Figure 1 *An example of a deep learning network.*

Many different kinds of deep learning networks have been proposed. They are efficient for various applications, such as Convolution Neural Networks (CNN)

for image processing, Recurrent Neural Networks (RNN) for sequence data processing, Natural Language Processing (NLP), and audio processing.

2.2 Convolutional Neural Network

Convolutional neural networks (CNN) are a type of deep learning proposed by LeCun et al. (2015). CNN are the most significant and effective forms of deep neural networks (Zafar et al., 2022) and are extensively employed in image processing applications that simulate human vision processing images by considering parts of the image with filters. The filter will extract various features of the image, called convolution operation. Then, the convolution result in the previous layer will be the input in the next layer. The strength of convolutional neural networks is that they can automatically extract features without human intervention (Alzubaidi et al., 2021). The convolutional neural network structure consists of a convolution layer, a pooling layer, and a fully connected layer, as shown in Figure 2.

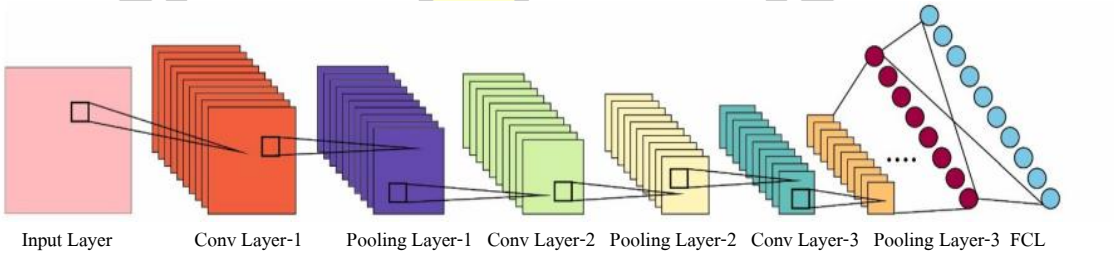


Figure 2 *Illustrate of the Convolutional Neural Network structure.*

2.2.1 Convolution layer

The convolution layer is a layer that transforms the input data to processing with filters to extract outstanding features related to the image. It is divided by the width, height, and color characteristics of the image as $W \times H \times D$, where W and H are image width and image height, respectively, and D is the color dimension of the image. For example, an RGB image can be divided into three dimensions: red, green, and blue. Characterization of an image is performed by calculating the dot product between the matrix and the filter. Convolution takes the weights of the filters together and shifts the filter until it reaches every region in the image with a stride that determines the step to move the filter. The result obtained by the convolution layer equals the number of filters applied. The result is called a feature map, as shown in Figure 3. In addition, padding can be used to increase the

margins where the borders of the image are meaningful, as shown in Figure 4. The equation for convolution operation is:

$$I_1^l = \left(\sum_j I_j^{l-1} \otimes w_{ij}^l + b_i^l\right) \quad (1)$$

Where $I_j^{l=1}$ is the output with $m \times n$ size, $\otimes$ indicates a convolution operator, w_{ij} represents convolution kernels and b_i is bias value.

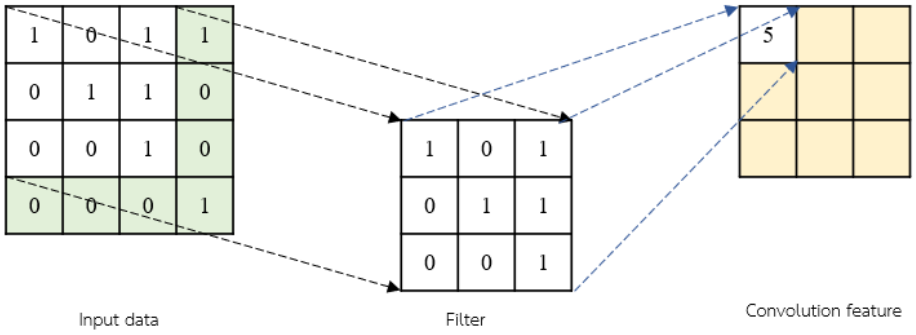


Figure 3 *The example of a convolution operator.*

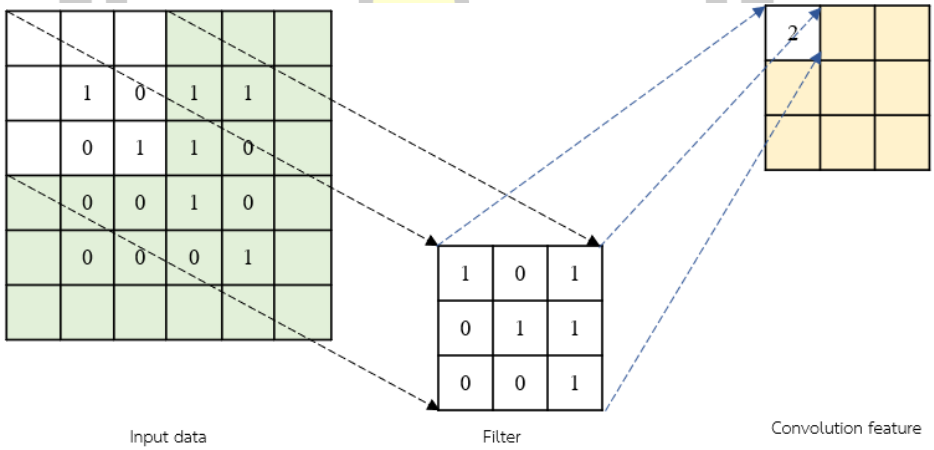


Figure 4 *Example of a convolution operator with padding.*

2.2.2 Pooling layer

The pooling layer is used to reduce the size of the features to bring only the essential information, most often the next layer after the convolution. The pooling layer can help to learn invariant features, reduces overfitting, and reduces computational complexity by down sampling the feature maps (Nirthika et al., 2022). Typically, the CNNs classify the pooling method into two types. (1) Local pooling is the first method to display feature maps by pooling data from small local regions. (2) Global pooling, which creates a scalar value representing the image from the feature

vector for each feature across the feature map (Zafar et al., 2022). The widely used pooling techniques are max pooling and average pooling (Boureau et al., 2010) , both used in local and global pooling layers. The max pooling technique identifies the biggest element in each pooling region (Singh et al., 2021) and discards other irrelevant information. The equation for max pooling is:

$$f_{max}(x) = max_i\{x_i\}_{i=1}^{N} \quad (2)$$

where N represents pooling region. An example of the max pooling technique gives the input data size of 4×4 and a filter size of 2×2 with a stride of 2. The maximum value is selected as the output, as shown in Figure 5.

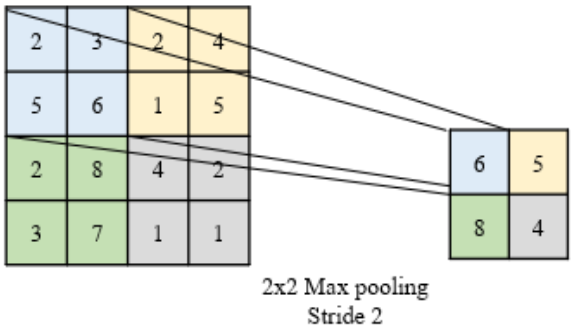


Figure 5 *Illustration of max pooling.*

The average pooling layer reduces the dimension of the features map by calculating the average value of a pool region, which does not consider the importance of a specific element in the pooling region . Mostly, average pooling is used as the global pooling operator to capture the contribution of all the features (Nirthika et al., 2022). The average pooling layer is usually used after a convolutional layer. The equation for average pooling is:

$$f_{avg}(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (3)$$

An example of the average pooling, the average pooling applied in patches of feature map with a stride of 2 is shown in Figure 6. (average pooling involves calculating the average for each patch).

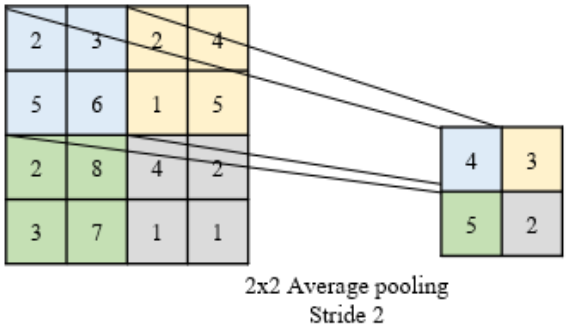


Figure 6 *Illustration of average pooling layer.*

2.2.3 Fully connected layer

The fully connected layer is the last layer of the convolution neural network and consists of many neurons that are interconnected. The fully connected layer connects to the complete output by flattening into a vector of dimension $1 \times m$ before entering the classification process. The equation for fully connected layer is:

$$X_{output}^{l} = f(X^{l-1} \times D^{l} + B^{l}) \quad (4)$$

2.3 Convolutional neural network architecture

A Convolution neural network produces an effective model when trained with a sufficient training data and appropriate functions. CNN architectures are used quite often because they can learn the features of the problems automatically. CNN architectures include, MobileNetV1 (Howard et al., 2017), MobileNetV2 (Sandler et al., 2018), NASNetMobile (Zoph et al., 2018), and ResNet50V2 (He et al., 2016).

2.3.1 MobileNetV1 (Howard et al., 2017)

MobileNetV1 is a lightweight convolutional neural network architecture for highly efficient image classification on mobile and embedded devices with limited resources. Howard et al. (2017) introduce MobileNetV1 in 2017 based on the concept of depthwise separable convolution, which consists of two separate layers: depthwise convolution and pointwise convolution. Depthwise convolution applies a single filter to each input channel, while the pointwise convolution with a 1×1 convolution was performed to change the dimension and create a linear output, as shown in Figure 7. This concept reduces the computational cost significantly compared to traditional convolutional layers.

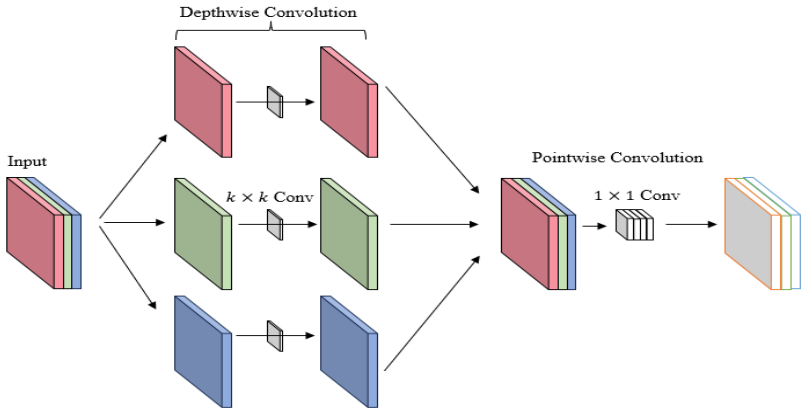


Figure 7 *Illustrate of the depthwise separable convolution (Howard et al., 2017).*

2.3.2 MobileNetV2

MobileNetV2 is the improved version of MobileNetV1. Two layers were added in the MobileNetV2 architectures: an inverted residual and a linear bottleneck, to enhance memory efficiency (Sandler et al., 2018). The inverted residual block contained a convolution layer, depthwise convolution, and convolution layer, with one stride. First, a pointwise (1x1) convolution is used to expand the dimensional input feature map to a higher dimensional with ReLU6 applied. Next, a depth-wise convolution is performed using 3x3 kernels, followed by ReLU6 activation. Finally, the spatially filtered feature map is reduced dimensionally using another pointwise convolution, and the linear is used instead of ReLU to avoid information loss. The shortcut connection was connected between each residual block the same way as in the residual network, as shown in Figure 8.

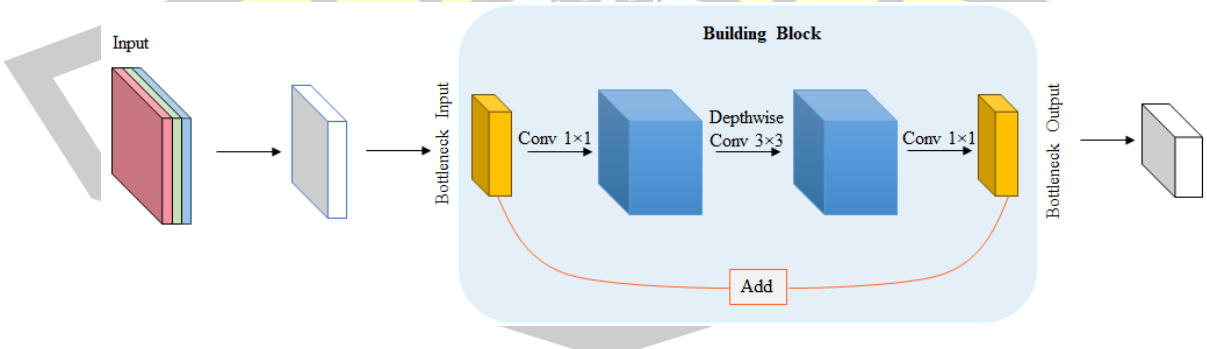


Figure 8 *The structural of MobileNetV2 (Sandler et al., 2018).*

2.3.3 NASNetMobile (Zoph et al., 2018)

NASNetMobile is the lightweight version of NASNet. It was designed to explore the best convolutional layer on a small dataset, such as the CIFAR-10 dataset, and then transfer the best layer by stacking the layers together to a large dataset, such as ImageNet (Zoph et al. 2018). To search for the best convolutional layer, it searches from many sets of convolutional operations, for example, identity, 3x3 convolution, 3x3 depthwise convolution, 3x3 average pooling, and 3x3 dilated convolution, using a recurrent neural network (RNN). NASNet consist of two main cells stacked together: normal and reduction cells. Although the normal and reduction cells were stacked together, the NASNet architecture could be adjusted by repeating many normal cells with N times.

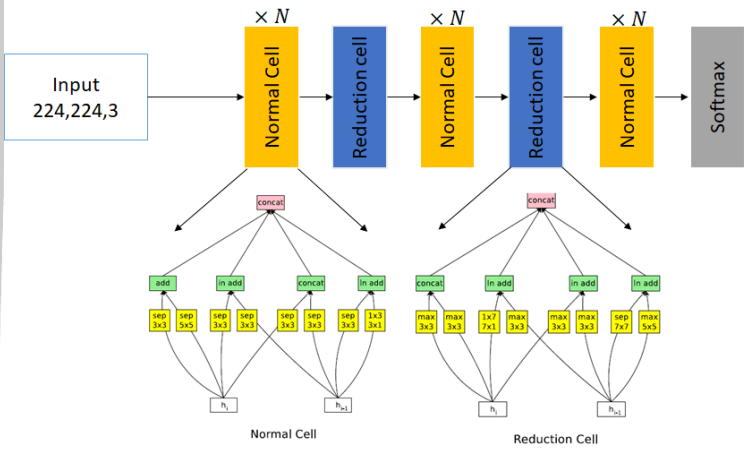


Figure 9 *The structural of MobileNetV2 (Sandler et al., 2018).*

2.3.4 ResNet50V2 (He et al., 2016)

ResNet50V2 is a modified version of ResNet50 that performs better than the original ResNet50 and ResNet101 on the ImageNet dataset. The difference between the residual block in the original ResNet and the modification ResNetV2 is the number of the convolution operation. The original residual block contained the weight layer, BN, ReLU, weight layer, and BN, respectively. Before combining to the following layer, the ReLU function was performed. While the modified residual block in ResNetV2 contains BN, ReLU, weight layer, BN, ReLU, and followed by weight layer. Hence, it adds to the following layer without applying the ReLU function. The

structure of the ResNet50V2 architecture consists of 50 convolutional layers, as shown in Figure 10.

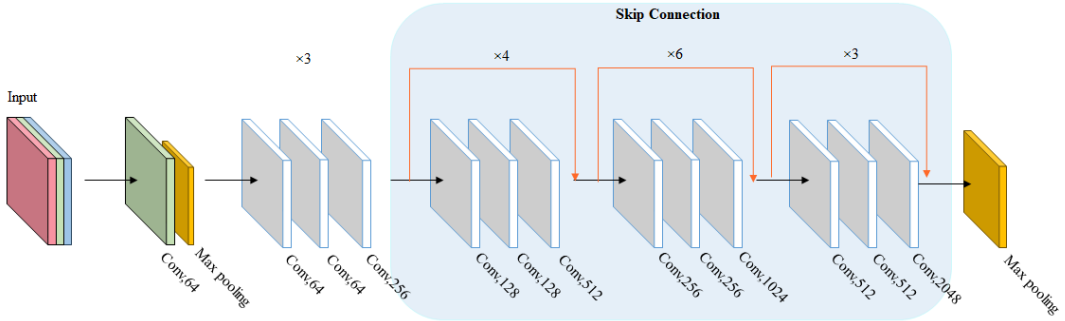


Figure 10 *illustrates the ResNet50V2 architecture (He et al., 2016).*

2.4 3D Convolution (Ji et al., 2013)

Three-dimensional convolutional neural networks (3D-CNN) recognize 3D images or video data, which differs from 2D convolutions for image classification. Ji et al. (2013) proposed the first 3D-CNN for action recognition, which extracts features from spatial and temporal dimensions by performing 3D convolutions with multiple adjacent frames. The dimension of input data for 3D convolution is F x W x H x D, where F is the video frame, W is the width of each frame, H is the height of each frame, and D is the color dimension of the image, such as the RGB system. The 3D kernel is used for convolution operations to the multiple contiguous frames together. The convolution shifts the kernel using a stride as the shift step. In addition, the padding layer can be used to increase the area of the video frame when it has necessary elements at the edges of the frame. The output layers of convolution becomes a cube consisting of output values, as shown in Figure 11.

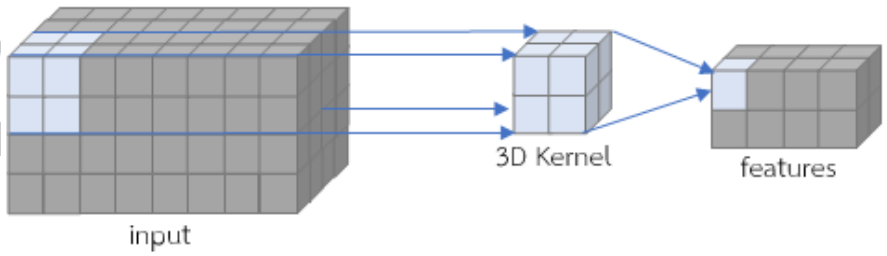


Figure 11 *The structure of 3D convolution (Ji et al., 2013).*

2.4.1 C3D (Tran et al., 2014)

C3D is a deep three-dimensional convolutional neural network for spatiotemporal feature learning of video data. C3D proposed by Tran et al. (2014), can learn both spatial features and temporal features from continuous frames by using 3D convolution and 3D pooling operation. The architecture of C3D consists of 8 convolutions, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are size 3×3×3 with stride 1 in both spatial and temporal dimensions. The convolution has a number of filters such as 64, 128, 256, 256, 512, 512, 512 and 512. All pooling kernels are 2×2×2, except for pool1 is 1×2×2. The fully connected layer has 4096 output units. The architecture of C3D is shown in Figure 12.

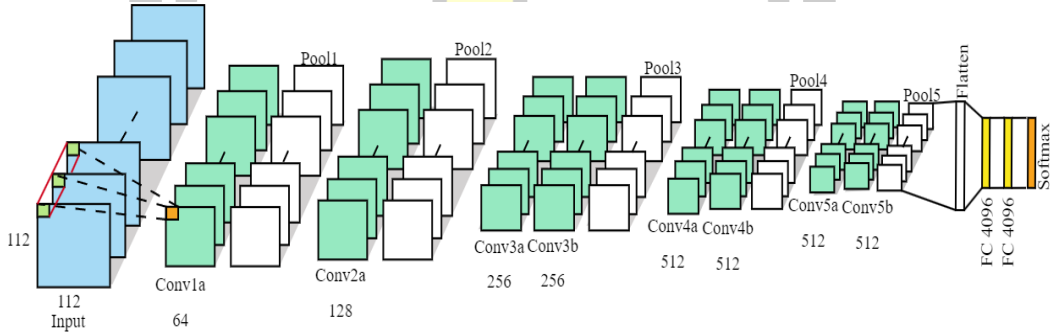


Figure 12 *The architecture of C3D (Tran et al., 2014).*

2.5 Recurrent Neural Network architecture

Recurrent Neural Network (RNN) is a network that forwards the output data from the hidden layer of the previous time step as the input data for the next time step as shown in Figure 13. RNN is applied to time series data or sequence data, for example, including language translation, speech recognition, handwriting recognition, video understanding, and generating image descriptions. An example of a recurrent neural network is the Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), and the Gate Recurrent Unit (GRU).

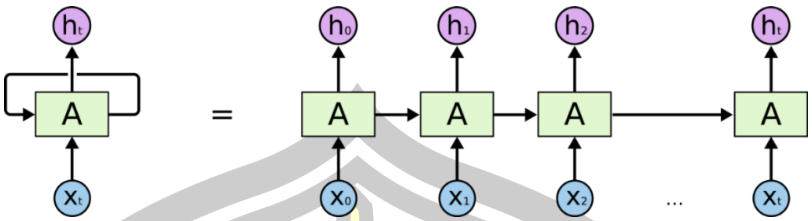


Figure 13 *Recurrent Neural Network.*

2.5.1 Long Short - Term Memory (LSTM)

LSTM is a type of artificial neural network developed to address the issue of vanishing information when dealing with long data sequences over extended periods. LSTM was proposed by Hochreiter and Schmidhuber (1997) in 1997, LSTM is designed based on gating mechanisms to control the flow of information and the state within the LSTM units during operation. The approach minimizes losing crucial information or keeping unnecessary data when dealing with extended sequences. The architecture of LSTM consists of four fundamental components including cell state, input gate, forget gate, and output gate, as shown in Figure 14.

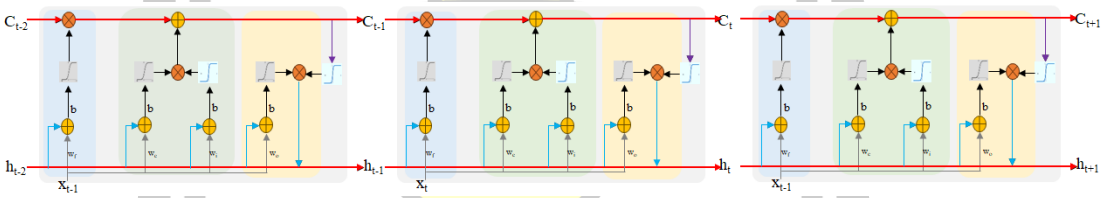


Figure 14 *Long Short - Term Memory architecture (Hochreiter & Schmidhuber, 1997).*

2.5.1.1 Cell state is core memory of the LSTM, the cell state, enables the network to retain information over long sequences. The cell state can store and modify data, essential for keeping context across different time steps.

2.5.1.2 Input gate decides which information from the current input and the previous time step should be stored in the cell state. The gate selectively updates the cell state with new data, enabling the network to adapt to changing patterns.

2.5.1.3 Forget gate considers if information should be discarded from the cell state. By screening out unessential data, LSTM avoids information overload and ensures that the cell state remains concise.

2.5.1.4 Output gate handles the amount of information extracted from the cell state to generate the output. This controlled flow of information helps in producing accurate predictions or classifications.

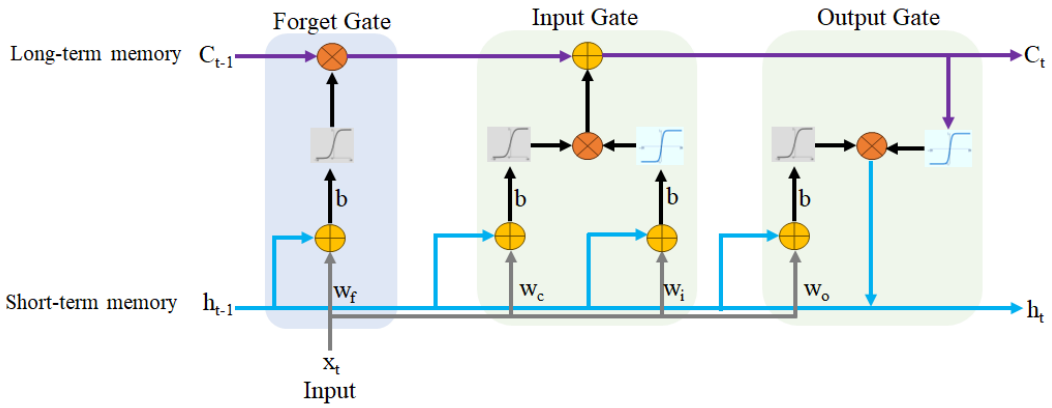


Figure 15 *Long Short - Term Memory cell (Hochreiter & Schmidhuber, 1997).*

2.5.2 Bidirectional Long Short-Term Memory (Bi-LSTM) (Graves & Schmidhuber, 2005)

Bi-LSTM was proposed by Graves and Schmidhuber (2005) and developed as an extension of the LSTM. Because LSTM processes data in one direction only. The Bi-LSTM difference from LSTM enables the capture of context from both directions in sequential data, including forward and backward directions. The Bi-LSTM architecture consists of two LSTM layers, including one processing the sequence in the forward direction and the other in the backward direction. The outputs from both layers are then combined to provide a comprehensive representation of the sequence data.

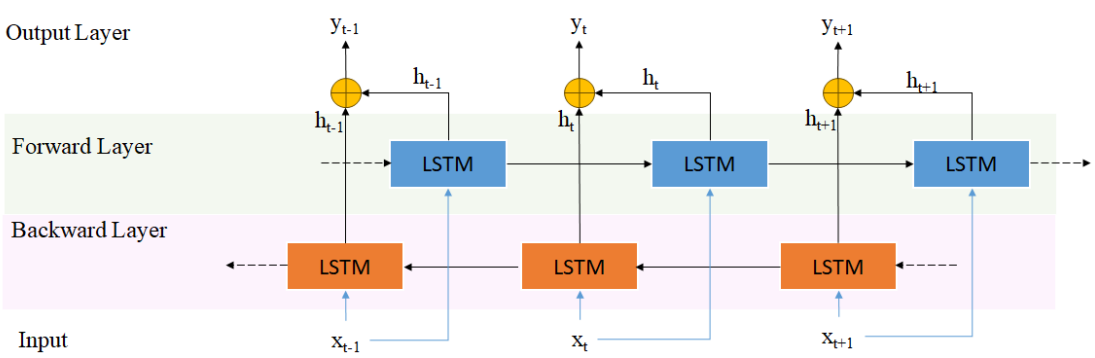


Figure 16. *Bidirectional Long Short-Term Memory architecture (Graves & Schmidhuber, 2005).*

2.5.3 Gate Recurrent Unit (GRU) (Cho et al., 2014)

GRU is a type of recurrent neural network architecture introduced by Cho et al. (2014) to address some of the limitations of traditional recurrent neural networks, such as the vanishing gradient problem and the difficulty of capturing long-range dependencies in sequences. GRU has the same function as the LSTM network but has a simplified architecture with fewer parameters, making it computationally less intensive and often easier to train. The previous sequence information is controlled by reset and update gates, as shown in Figure 17. Further, the update gate combines the input and forget gates into a single gate. The GRU network has fewer hyperparameters to adjust. Thus, it trains the model faster than the LSTM network (Toharudin et al., 2020).

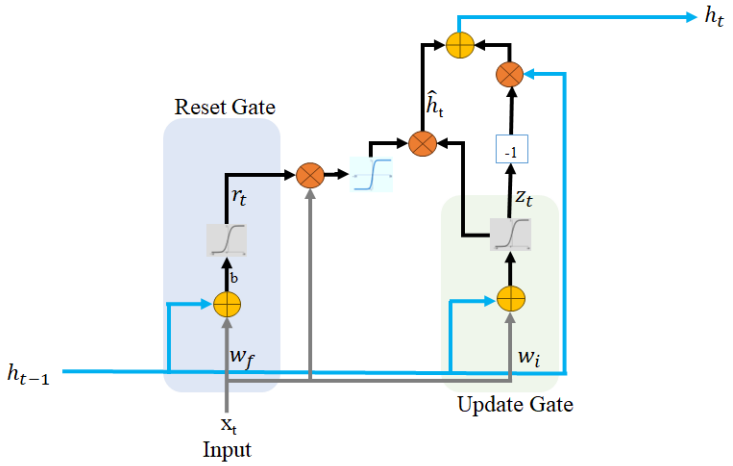


Figure 17 *Gate Recurrent Unit architecture. (Cho et al., 2014)*

2.6 Deep features extraction

Feature extraction is a critical step in machine learning, as it involves transforming raw data into a more suitable structure for model training and analysis. The CNN models are widely used for feature extraction, pre-trained to extract features, such as the MobileNet model, and the ResNet model trained in big data, such as ImageNet. The advantage of this method is that it can use the existing classical model, which has been pre-trained by many data (Lu et al., 2023). CNN model can learn meaningful representations from the input data autonomously, and it can extract different levels (low, medium, and high) of features from raw data at the difference convolution layer , as shown in Figure 18. In the CNN model, the first layers discover

low-level features, such as edges, lines, and corners. The other layers discover mid-level and high-level features, for instance, structures, objects, and shapes (LeCun et al., 2015).

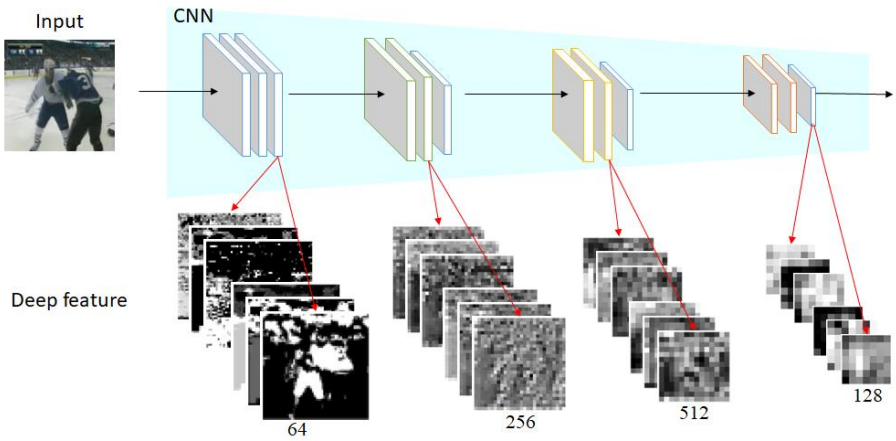


Figure 18 *Deep feature extraction using CNN.*

For video recognition, deep feature extraction is mostly used to extract the essential features at the frame level. Then, the resulting deep features are sent to sequenced data models learning such as LSTM or Bi-LSTM to understand essential features between frames and can be used to classify videos effectively.

2.7 Deep features fusion method

Integrating deep features involves combining distinct features extracted from different sources, such as features extracted from diverse convolutional layers or features derived from different models. These features are merged to create new representations that perform as description of the data. Subsequently, these integrated features are utilized to train classification models for subsequent tasks. Fusing features from multiple sources offers several advantages over learning from a single feature set (He et al., 2016). For instance, features extracted from various convolutional layers capturing different feature levels can be integrated to create a more comprehensive representation of the input data. Integrating deep features facilitates the creation of enriched and more informative representations that enhance classification performance. The methods for deep feature fusion include addition and concatenation operations (J. Liu et al., 2022). The addition operation corresponds to

an increased information for the features describing the image. However, the dimensions describing the image do not increase, as shown in Figure 19.

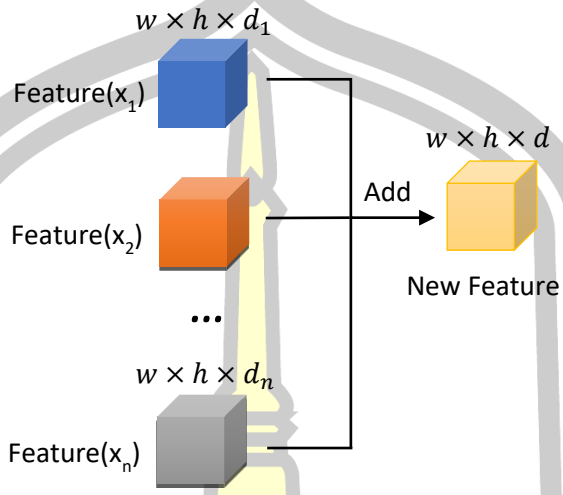


Figure 19. *Feature fusion using addition operation.*

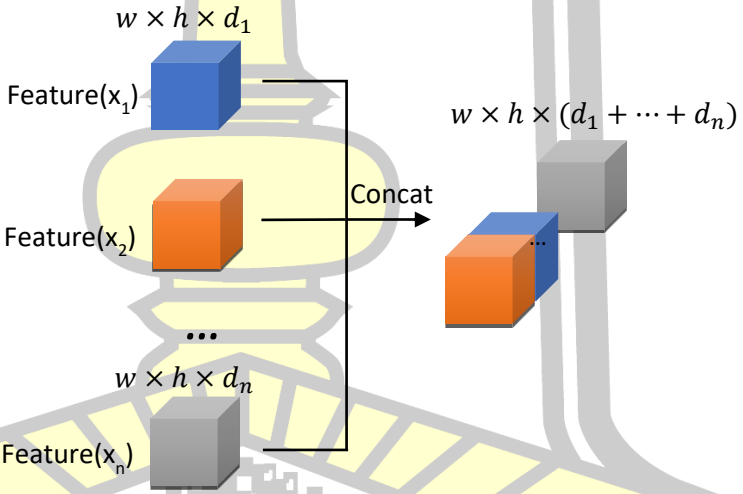


Figure 20. *Feature fusion using concatenation operation.*

In contrast, concatenation operation refers to a merger of the number of channels, and the number of channels refers to the sum of Feature (X_1) to Feature (X_n) channels, as shown in Figure 20.

2.8 Violent dataset

This section describes violent video datasets that are widely used in violence recognition. The dataset used for violent videos was collected from various sources

such as sports games, YouTube, movie, and CCTV. Each dataset has a different resolution of videos, people, and scenes. I can explain each data set as follows.

2.8.1 Hockey fight dataset (Bermejo et al., 2011)

Bermejo et al. (2011) proposed a hockey fight dataset in 2011 collected from National Hockey League hockey games. The dataset consists of 1,000 videos, which are divided into 500 violent videos and 500 nonviolent videos. The video consists of 41 frames and a resolution of 720×576 pixels. A sample frame from the hockey fight dataset is shown in Figure 21.

violence video non-violence video

Figure 21 *Samples of hockey fight dataset.*

2.8.2 Movie dataset (Bermejo et al., 2011)

The movie dataset proposed by Bermejo et al. (2011) consists of 200 videos collected from action movie. The violence class consists of 100 videos collected from action movie scenes, while the nonviolence class was collected from other publicly available action recognition datasets that do not contain violent action. The duration of each video clip is around 2 seconds. The sample frame from the movie dataset is shown in Figure 22.

violence video non-violence video

Figure 22 *Samples of movie dataset.*

2.8.3 Violent flow dataset (Hassner et al., 2012)

In 2012, Hassner et al. (2012) proposed the violent flow dataset consisting of 246 videos that contain crowds with scenes of a fighting between people. The videos were collected from violent situations that occured in football matches. The dataset is divided into 123 violent videos and 123 nonviolence videos. The videos in this dataset range from 1.04 seconds to 6.53 seconds. A sample frame from the violent flow dataset is shown in Figure 23.

violence video non-violence video

Figure 23 *Samples of violent flow dataset.*

2.8.4 RWF2000 dataset (Cheng et al., 2021)

Cheng et al. (2021) proposed the RWF2000 dataset which collected real-world fighting videos from YouTube, consisting of 2,000 real-world video clips, surveillance cameras, and social media. Half of the videos include violent behaviors, while others depict nonviolent activities. For violence videos include, any form of

subjectively identified violent actions such as fighting, robbery, explosion, hooting, blood, and assault. The duration of each video clip is around 5 seconds with 30 FPS. A sample frame from the RWF2000 dataset is shown in Figure 24.

violence video non-violence video

Figure 24 *Sample of RWF2000 dataset.*

2.8.5 UCF crime dataset (Sultani et al., 2018)

Sultani et al. (2018) proposed the UCF crime dataset as a long untrimmed video collection of 1,900 real-world surveillance videos, comprising 950 for violent and 950 nonviolent videos. Videos in this dataset usually have a duration from 1 to 10 minutes. The dataset consists of 13 types of regular activities and violent classes: abuse, arrest, arson, assault, traffic accident, burglary, explosion, fight, robbery, burglary, shooting, theft, shoplifting, and vandalism. The number of videos in each category and an example video of the UCF crime dataset are shown in Table 1 and Figure 25, respectively.

Table 1 *A number of videos in each category of UCF-crime dataset.*

Classes	**videos**	**Classes**	**video**
Abuse	50	Road accident	150
Arson	50	Robbery	150
Arrest	50	Shooting	50
Assault	50	Shoplifting	50
Burglary	100	Stealing	100
Explosion	50	Vandalism	50
Fighting	50	Normal	950
Total			1,900

Figure 25 *Samples of UCF crime dataset.*

2.9 Related work

2.9.1 Deep learning for video classification.

Deep learning techniques are essential and successful tools in video classification tasks. The advantage of the deep learning model is the ability to automatically recognize and classify the videos accurately. Video recognition differs from image recognition because it can realize various input data, while image classification is performed on images. ur Rehman et al. (2023) categorized the video classification task as a uni-modal or multi-modal video classification. The uni-modal recognizes video from single input data, such as text, audio, or visual information. In contrast, multi-modal classification is a combination of text, audio, or visual information. Some research has resulted in methods being proposed based on a single modal for video classification. Yadav and Vishwakarma (2020) propose a deep affect-based movie genre classification framework. This proposed method involves cropping video frames with faces and ignoring the rest in a preprocessing step. Then, the spatial

features were extracted via the InceptionV4 network to obtain robust features. The Bi-LSTM and LSTM were added to help in generating an effective feature. The final feature was passed to softmax classification to obtain probabilities. Finally, a stacked ensemble was used for classifying a movie's trailer. The result indicated that the proposed method outperforms all the state-of-the-art methods significantly.

Ramesh and Mahesh (2022) proposed a framework based on deep learning to classify sports videos using sports video as an input. First, the frame extraction process converts the input videos into frames and reduces noise with the fuzzy adaptive median filtering technique. Then, an enhanced threshold-based frame difference algorithm is applied to identify the keyframe. Finally, CNN is utilized for feature extraction and classification. The result shows that this framework offers improved performance with less computational expense, and feature extraction architectures using CNN can outperform hand-crafted features. Z. Liu et al. (2022) presented a pure transformer backbone architecture for video recognition implemented through a spatiotemporal adaptation of the Swin Transformer, which achieves state-of-the-art performance on benchmark datasets.

Recently, multi-modal video classification has gained attention for video classification. Some research utilizes characteristics of video, audio, and text attributes to improve more efficient results than incorporating only one feature. Gao et al. (2019) proposed a framework for efficient action recognition in video that considers jointly frame and audio. The image frame captures most of the appearance information within the video, while the audio provides important dynamic information. The pair of images and audio were selected to perform efficient video-level action recognition. Tahir et al. (2020)extracted features of frame, movement, and audio information of video scenes through VGG-19. They further extracted movement features with the BiLSTM model. All features are concatenated and forwarded to a fully connected neural network to detect the disturbed and fake embedded content in videos. The result shows that the combined features outperform the individual features.

Also, Lou et al. (2021) proposed a fusion of auditory and frame-level features through the CNN-LSTM for violence recognition. The result proved that the fusion feature method obtained better recognition results and improved the accuracy

of violent behavior recognition. Pratama et al. (2023) proposed violence recognition using a two-stream 3D convolution network, which used video frame and optical flow as input. Ma et al. (2023)proposed a two-stream inflated 3D convolution network for human behavior recognition that learns action features directly from RGB and optical flow inputs. The results showed that the proposed method achieved the highest performance on UCF-101 and HMDB-51 datasets by reducing misclassification by 57% and 33%, respectively. Wang et al. (2023) proposed a two-stream deep learning architecture for video violent activity detection. The RGB frames and optical flow data were used as inputs for each stream to extract the spatiotemporal features of videos. After that, the spatiotemporal features from the two streams were concatenated and fed to the classifier for the final decision.

2.9.2 Hand crafted features for violent recognition

Recognition of violence in surveillance video used a handcrafted approach for feature extraction based on images. Then aggregate the features were aggregated using encoding strategies and machine learning applied as a classifier (Li et al., 2019). Some research has considered spatiotemporal descriptors around an interesting point to recognize the violence in surveillance video. Souza et al. (2010) presented a violence detector based on local spatiotemporal features with a bag of visual words and a support vector machine. The results confirm that motion patterns are crucial to distinguish violence from regular activities compared to visual descriptors in the space domain. Bermejo et al. (2011)introduced a fight dataset and used space-time interest points and motion scale-invariant feature transform method to extract spatial-temporal features. Then, the feature vector was sent to the support vector machine classifier.

Similarly, Xu et al. (2014)used the motion scale-invariant feature transform method to extract the low-level description of a query video. The kernel density estimation is exploited for feature selection to obtain the highly discriminative video feature. A sparse coding method with a max pooling procedure generates a discriminative high-level video representation from local features. The result showed that the proposed method outperformed violence detection in crowded and non-crowded scenes. Das et al. (2019) proposed a system to detect violence from video, applied a histogram of oriented gradient as a feature descriptor to extract features

from the images, and employed various classifier models and a majority voting technique to decide whether a video clip contains violence or not. The result showed that the system is robust enough to detect violence in different surveillance situations. Several researchers have proposed methods for global feature extraction, such as Hassner et al. (2012)presenting a violent flow feature descriptor based on optical flow magnitude changes between adjacent violent video frames. Gao et al. (2016) improved the violent flow feature descriptor to use the orientation information of optical flow, a namely oriented violent flow which considers both magnitude and orientation information. The features are encoded into the bag of word representation and a support vector machine for violence in the video classifier. However, the hand-crafted features are usually dataset dependent and do not generalize well (Wang et al., 2023).

2.9.3 Deep learning for violent recognition

Many approaches have been proposed to recognize violent video, which is categorized into 2D-CNN, 3D-CNN, combination of CNN and RNN, and fusion of features approaches.

2.9.3.1 Violent recognition with 2D-CNN

The image-based approach utilizes a two-dimensional convolution neural network (2D-CNN) for frame-level feature extraction. 2D-CNN can capture spatial features from individual frames of video. The resulting discriminative features are then classified using a state-of-the-art classification model such as SVM. Some researchers have developed convolutional neural networks (CNNs) for performing violent video recognition. Irfanullah et al. (2022) proposed real-time violence detection in surveillance videos using convolutional neural networks. This research compares the performance of different CNN models such as AlexNet, VGG-16, GoogleNet, and MobileNet for violence recognition. The result indicated that the MobileNet model outperformed the other models regarding accuracy, loss, and computation time. Khan et al. (2019) presented a violence detection approach using deep learning. The video was segmented into shots and selected representative frames with a maximum saliency score. Then, the selected frames were learned by a lightweight deep learning model and classified as depicting violence or non-violence. Keçeli and Kaya (2017) used a pre-trained CNN for deep

high-level features extraction that applied an optical flow as the input of the network and classified violent activities by SVM and subspace k-nearest neighbor (SkNN). Karisma et al. (2021)used a pre-trained VGG16 model for the feature extraction method and classified it using the support vector machine (SVM) algorithm with the linear kernel. VGG16 extracted 4,096 features and was used as the input to the SVM. The experimental results showed that the VGG16 combined with SVM achieved an accuracy of 96.4%.

2.9.3.2 Violent recognition with 3D-CNN

From the above, 2D-CNN is performed on individual frames without considering the temporal information between adjacent frames (Lin et al., 2019), especially for video recognition tasks that consider the time information involved. Therefore, further developments will extend the capabilities from 2D-CNN to 3D-CNN to extract appropriate video features. 3D-CNN can analyze both spatial and temporal information. Several 3D-CNN architectures have been proposed to sustain more factual video recognition performance. Ji et al. (2013) proposed a 3D-CNN to extract spatial and temporal features from video data for action recognition. The experimental results showed that the proposed models significantly outperformed 2D-CNN architecture. Su et al. (2022) employed the X3D network to detect violence captured by surveillance cameras. The X3D network is a 3D-CNN that is designed for activity recognition and fine-tuned to detect violence in real time. The 3D kernels are designed to deal with information from both the spatial and temporal domains in the same manner. The experimental result demonstrates that our modified model outperforms most other violence detection methods with simple hyperparameter adjustments.

Alharthi et al. (2023) examined various deep-learning models to enhance abnormal behavior detection on the massive Hajj crowd dataset. To extract spatiotemporal features from a video containing anomalous behavior, pre-trained C3D models are employed. The obtained spatiotemporal features are fed to fully connected layers that are trained to classify the video into one of the seven abnormal classes or the normal class. The result shows that the C3D model outperformก VGG-16 and the other research approaches. Maqsood et al. (2021) proposed a framework for

recognizing anomaly videos by learning spatiotemporal features using a deep 3-dimensional convolutional network. The experiment was trained on the University of Central Florida (UCF) Crime dataset. The proposed approach consists of 3D feature extraction and spatial augmentation by the proposed 3D ConvNet. The result shows that the 3D ConvNet outperforms significantly from the state-of-the-art method on anomalous activity recognition having 82% AUC. Jahlan and Elrefaei (2022) proposed a novel approach using the fusion technique to detect violence. First, both Alexnet and SqueezeNet networks are followed by Convolution Long Short-Term Memory (ConvLSTM) to extract robust features from the video. Then, the obtained features were fused and fed into the max-pooling layer, fully connected layer, and softmax classifier.

2.9.3.3 Violent recognition with combination of CNN and RNN

Another method for violent video recognition is using RNN to encourage performances with spatial and temporal features, which jointly consider information about the previous and current frames. The survey of Morshed et al. (2023) found that current approaches based on RNN often use LSTM to handle lengthy action sequences because this architecture may avoid the overall disappearance of gradient issues. For violent video recognition, many studies apply CNN for spatial feature extraction and then employ RNN for temporal feature extraction in video recognition. Sudhakaran and Lanz (2017) use frame difference as input of a 2D-CNN to extract hierarchical features from the video frames and then aggregated them using the convLSTM layer. Then they were classified as violence or non-violence with a fully connected layer. The experimental result showed that a deep neural network trained on the frame difference performed better than a model trained on raw frames.

Soliman et al. (2019) proposed the pre-trained VGG-16 model on ImageNet to extract spatial and LSTM to extract temporal features before being classified by a fully connected layer. Experiments on standard violent data sets showed that the model outperformed the state-of-the-art approach. Besides, they created a real-life violence situations (RLVS) dataset for fine-tuning the model, achieving the best accuracy of 88.2% on the hockey fight dataset. Sumon et al. (2019) demonstrated the efficiency of the deep learning method by using CNN, LSTM and

combining CNN with LSTM. The experiment on violent video datasets found that the CNN model with transfer learning performed better than LSTM and CNN-LSTM models. Naik and Gopalakrishna (2021) proffered the deep neural network model Mask Region-based Convolutional Neural Network (Mask RCNN) to detect a single person in the video and extract interest points. Then the extracted features were fed to LSTM for feature learning across a time series frame. The results showed that the model had excellent performance.

Some researchers have applied bidirectional LSTM to improve model performance by combining CNN with BiLSTM. Mumtaz et al. (2022) proposed a multi-scale of VGG-19 architecture for violence video classification. The VGG-19 was used to initialize the spatial features extractor, followed by the widely followed Bi-LSTM structure for optimal recognition of violent. Hanson et al. (2019) proposed a spatiotemporal encoder to detect video violence. First, each video frame was extracted as feature maps with the VGG13 network. Then the feature maps were passed to BiConvLSTM to extract the temporal information by passing forward and backwards in time. Finally, elementwise maximization was applied to represent the video and classified as violent or non-violent in the video. The testing accuracy achieved 96.96% on the hockey fight dataset, 100% on the movie dataset, and 90.6% on the violent flow dataset.

2.9.3.4 Violent recognition with fusion features.

Some research uses deep learning to learn from various features type. Jahlan and Elrefaei (2022) apply the feature fusion technique to recognize violence, in which the features were fusion obtained from AlexNet, SqueezNet, and LSTM. Correspondingly, Tahir et al. (2020) extracted the features from the VGG-19 and BiLSTM model and then combined all features with concatenation for violence recognition in YouTube videos. P. Zhou et al. (2017) constructed ConvNets, namely FightNet, to model long-term temporal structures for recognizing violence. The input consists of an RGB image, optical flow, and acceleration field to extract the motion information better. Their approach demonstrated that deep ConvNets could capture more essential features and detect violence accurately.

Carneiro et al. (2019) used VGG-16 for a multi-stream that includes spatial, temporal, rhythm, and depth information. The model achieved an accuracy of 89.10% on the hockey fight dataset and 100% on the movie dataset. The results showed that multi-stream methodology increased the efficiency of violent video recognition. Lou et al. (2021) proposed a method for an autoencoder mapping method for auditory-visual information fusion. The model comprised four parts: visual feature extraction, auditory feature extraction, autoencoder model, and full connection recognition model. CNN was used to extract frame level features and auditory features. Next, the frame level feature and auditory features were sent to LSTM network to process temporal relationship of the features. Then, the visual features and the auditory features were fused with the concatenate method. Finally, the full connection model was used to identify violent behavior. The result showed that the proposed method improved the performance of violent behavior recognition.

Chapter 3
Fusion Lightweight CNNs and Sequence Learning Technique

Stopping violent incidents in real-life is more dangerous for ordinary people. It may harm people's lives. Calling the police is the best choice to stop the violence. We should have an automatic system to recognize violence and warn the police on time. This paper proposes a method to classify violent incidents from video. However, classification of violent videos faces many challenging problems, such as video length, quality, and angles and orientations of the recording devices. The proposed method is called fusion MobileNets-BiLSTM architecture. In the first part, we propose to use the lightweight MobileNetV1 and MobileNetV2 to extract the robust deep spatial features from the video so that only non-adjacent 16 frames were selected. The spatial features were transferred to the global average pooling, batch normalization, and time distribution. In the second part, the spatial features from the first part were concatenated and then sent to create the deep temporal features using the bidirectional long short-term memory (BiLSTM). The proposed fusion MobileNets-BiLSTM architecture was evaluated on the hockey fight dataset. The experimental results showed that the proposed method provides better results than the existing methods. It achieved 95.20% accuracy on the test set of the hockey fight dataset.

3.1 Introduction

Video surveillance systems are essential to save human life and reduce the risks of becoming a victim of crime (Lejmi et al., 2020) (Lejmi, Ben Khalifa, and Mahjoub 2020). A crime can happen anywhere and anytime, causing damage to life and property. Most public or private places have established video surveillance systems to monitor human activity and prevent crime. However, using human monitoring through video surveillance may not stop the incident. Therefore, applying computer vision technology to video surveillance systems is crucial to identify in real-time and warn related agencies when an abnormal event occurs. The need is to recognize violent activities such as fighting, punching, and kicking from a person or crowd. It is imperative to understand video and efficiently apply it to the real world.

The main contributions of the proposed architecture are presented in the following. We proposed the lightweight MobileNets to extract the deep spatial features and bidirectional long short-term memory (BiLSTM), which is a recurrent neural network, to learn from the sequence video frames and extract the temporal features. We proposed the concatenating operation to combine the spatial features that were extracted using the MobileNetV1 and MobileNetV2 before sending the spatial features to the BiLSTM network. The softmax function was used as the classifier of the proposed architecture. Hence, we selected keyframes which were the only 16 non-adjacent frames. However, other methods were examined with 20 and 40 frames. In this paper, all 16 keyframes were input to the proposed fusion lightweight CNNs and sequence learning architecture. The output was classified as violence and non-violence.

The remainder of this chapter is organized as follows. Section 3.2 summarizes the overview of related work. Section 3.3 describes the proposed fusion lightweight CNNs and sequence learning architecture. The violence video dataset is explained in Section 3.4. The experimental setup, and experimental results are presented in Section 3.5. The conclusion and future work are given in Section 3.6.

3.2 Related work

Nowadays, deep learning is developing rapid detection and recognition of violence in surveillance video. When comparing deep learning methods with traditional methods, deep learning methods have strong feature expression ability and minor limitations (Jiaxin et al., 2021). Some researchers have developed convolutional neural networks (CNNs) for performing violent video recognition (Kreuter et al., 2020; Lejmi et al., 2020; Siregar & Mauritsius, 2021). Khan et al. (2019) presented a violence detection approach using deep learning. The video was segmented into shots and selected representative frames with a maximum saliency score. Then, the selected frames were learned by a lightweight deep learning model and classify them as violence or non-violence. Keçeli and Kaya (2017) used a pre-trained CNN for deep high-level features extraction that applied an optical flow as the input of the network and classified violent activities by SVM and subspace k-nearest neighbor (SkNN). Karisma et al. (2021) used a pre-trained VGG16 model for the

feature extraction method and classified it using the support vector machine (SVM) algorithm with the linear kernel. VGG16 extracted 4,096 features and was used as the input to the SVM. The experimental results showed that the VGG16 combined with SVM achieved an accuracy of 96.40%.

Some studies have proposed combining CNN and LSTM networks with learning sequence data from video. Soliman et al. (2019) proposed an end-to-end deep neural network model for recognizing violence in video. The VGG16 was used for spatial feature extraction, followed by LSTM for extracting the temporal features. Then, the fully connected and softmax layers were used as classification. Their method achieved the best accuracy of 95.10% on the hockey fight dataset. Ditsanthia et al. (2018)proposed a new visual feature descriptor, called multi-scale convolutional features, to partition the video frame into different regions and extract deep features. Then, the features were pooled together to obtain a meaningful feature vector. Finally, the frame-level features were fed into the BiLSTM to classify violence from the video.

Carneiro et al. (2019)focused on the using a multi-stream of VGG-16 networks and investigating conceivable feature descriptors of a video, including spatial, temporal, rhythmic, and depth information. Then, the outputs were classified using the ensemble method. Peixoto et al. (2020)proposed a fusion model based on visual and audio feature representation to tackle violence detection in video. First, the video frame features were extracted using C3D, CNN-LSTM, and InceptionV4, whereas the audio features were calculated using four standard audio feature extractor methods. Then, the different visual and audio features vectors were fused with a concatenation operation. Finally, A random forest and a softmax function were used as classifiers. The result showed that the classification accuracy increased 6% when combining visual and audio features. Lou et al. (2021) proposed an autoencoder mapping method for auditory-visual information fusion, using a CNN-LSTM architecture for feature extraction. Then, the visual and auditory features were integrated into the same shared subspace using an autoencoder model. Next, the output from autoencoder mapping was combined with the concatenation method. Finally, the softmax function was used to identify violent behavior. The result showed that their proposed method improved the performance of violent behavior recognition.

In the above studies, CNN extracted only spatial features. However, information sent to create the deep learning model for video classification is insufficient (Chen et al., 2021), although many studies use the RNN architecture to learn from the sequence data and increase the performance of the violence recognition. Therefore, for the surveillance system to recognize more accurately, the feature-fusion method receives more attention because the combination of features can significantly improve the efficiency of violence recognition.

3.3 Fusion Lightweight CNNs and Sequence Learning Architecture

In this section, we present the fusion lightweight CNNs and sequence learning architecture to classify violent incidents from videos.

Overview of the architecture, we divided the proposed architecture into two main parts. For the first part, the deep spatial features are extracted from the violence videos using lightweight MobileNetV1 and MobileNetV2. In addition, we removed the two last layers of MobileNetV1 and V2 and replaced them with global average pooling (GAP), batch normalization (BN), and time distribution layers. Hence, the deep spatial features from MobileNetV1 and V2 were connected with the concatenating operation. For the second part, we proposed the bidirectional long short-term memory (BiLSTM), which is a sequence learning architecture, to learn from the sequence features and extract the robust temporal features. The framework of the proposed architecture is shown in Figure 26. The details of each part are described in the following sections.

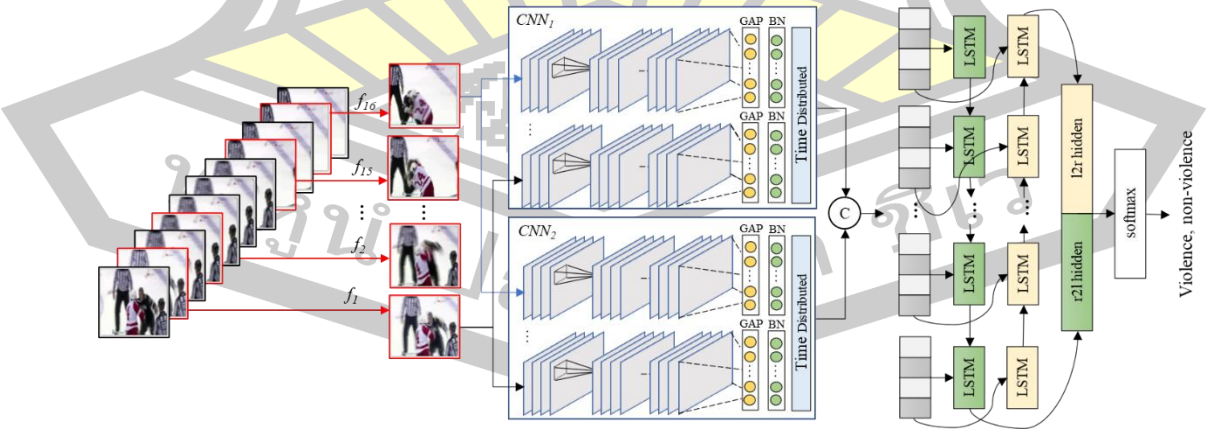


Figure 26 *Illustration of the fusion lightweight MobileNets and BiLSTM architecture for violence video recognition.*

3.3.1 Convolutional Neural Network Architectures. CNN is generally used for video recognition tasks because it effectively captures spatial information within video frames. CNN uses convolutional filters capable of capturing features such as edges, textures, and object shapes, which are crucial for understanding the content of individual frames. In this thesis, we are interested in lightweight neural network architectures with few parameters but still have high performance, including MobileNetV1, MobileNetV2, NASNetMobile, and ReNet50V2. The details of the CNN architectures are as follows.

3.3.1.1 MobileNetV1 is the lightweight CNN architecture, has a small number of parameters because the depthwise separable convolution operation was invented (Howard et al., 2017). Depthwise convolution was applied to each channel. Then, the pointwise convolution with a 1x1 convolution was performed to change the dimension and create a linear output. In the MobileNetV1 architecture, the depthwise separable convolution was attached to the convolution operation in every layer. Further, the BN and rectified linear unit (ReLU) activation function was combined after each convolution. The model of the MobileNetV1 is much smaller than VGG16 and GoogLeNet.

3.3.1.2 MobileNetV2 is the improved version of MobileNetV1. Two layers were added in the MobileNetV2 architectures: an inverted residual and a linear bottleneck, to enhance memory efficiency (Sandler et al., 2018). The inverted residual block contained a convolution layer, depthwise convolution, and convolution layer, respectively, with one stride. The shortcut connection was connected between each residual block the same way as in the residual network. The linear bottleneck block also contained the same layer as the inverted residual layer, but the stride was set as two.

3.3.1.3 NASNetMobile is the lightweight version of the NASNet. It was designed to explore the best convolutional layer on a small dataset, such as the CIFAR-10 dataset, and then transfer the best layer by stacking the layers together to a large dataset, such as ImageNet (Zoph et al., 2018). To search for the best convolutional layer, it searches from many sets of convolutional operations, for example, identity, 3x3 convolution, 3x3 depthwise convolution, 3x3 average pooling,

and 3x3 dilated convolution, using a recurrent neural network (RNN). NASNet consisted of two main cells were stacked together: normal and reduction cells. Although the normal and reduction cells were stacked together, the NASNet architecture could be adjusted by repeating many normal cells with N times.

3.3.1.4 ResNet50V2 is a modified version of ResNet50 that performs better than the original ResNet50 and ResNet101 on the ImageNet dataset (He et al., 2016). The difference between the residual block in the original ResNet and the modification ResNetV2 is the number of the convolution operation. The original residual block contained the weight layer, BN, ReLU, weight layer, and BN, respectively. Before combining to the following layer, the ReLU function was performed. While the modified residual block in ResNetV2 contains BN, ReLU, weight layer, BN, ReLU, and followed by weight layer. Hence, it adds to the following layer without applying the ReLU function. For my experiments, we removed the last two layers of each CNN architecture before extracting the deep spatial features. A summary of the CNN architectures is presented in Table 2.

Table 2 *A number of parameter of CNN architectures.*

CNN Architectures	No. of Parameters
MobileNetV1	4.2 M
MobileNetV2	3.2 M
NASNetMobile	5.3 M
ResNet50V2	25.6 M

3.3.2 Sequence Learning Architectures. For violent video understanding, using only CNN cannot capture long-term dependencies within sequence data due to the involvement of spatial information with convolutional operations. At the same time, RNN is designed to handle and effectively capture temporal dependencies in sequential data, emphasizing the importance of data sequence. RNN uses shared weights across different time steps, enabling it to capture dependencies across sequences effectively. In this study, the sequence information of 16 keyframes that were extracted from the violent video was first extracted using the CNNs and then

transferred to the sequence learning architectures. The brief details of the sequence learning architectures are as follows.

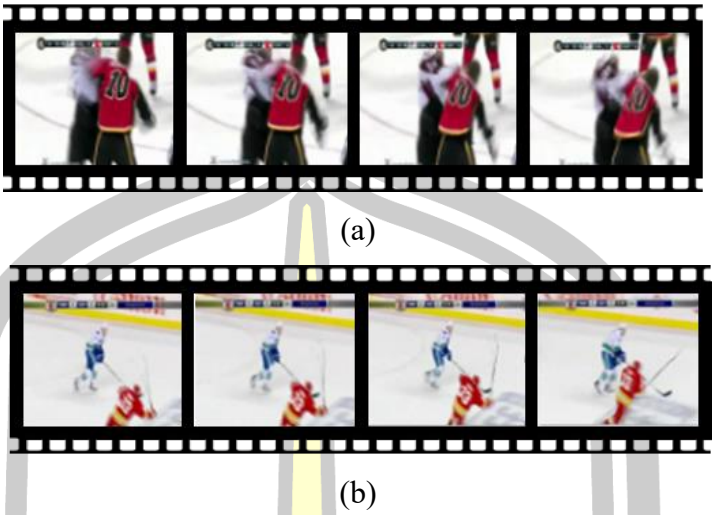
3.3.2.1 Long short-term memory (LSTM) was designed by Hochreiter and Schmidhuber (1997) to overcome the error of back-flow problems. LSTM has a memory block, which is a set of recurrently connected blocks, multiplicative units: input, output, and forget gates. The advantage of the LSTM network is that it was proposed to deal with long sequential data, including video, speech, and long text data. The gates were designed to keep or forget information while training the LSTM network. The LSTM learned from the sequence information and extracted the robust temporal features.

3.3.2.2 Bidirectional LSTM (BiLSTM) is a sequence learning architecture that processes sequence information in two directions (Graves & Schmidhuber, 2005). It consists of two independent LSTM networks: forward state and backward state. The forward state takes the input in a forward direction. At the same time, the backward state takes in a backward direction. The outputs of the two states are connected to the same output.

3.3.2.3 Gated Recurrent Unit (GRU) was introduced by Cho et al. (2014) and has the same function as the LSTM network. The previous sequence information is controlled by reset and update gates. The reset and update gates were designed to control the previous sequence information. Further, the update gate combined the input and forget gates into a single gate. The GRU network has fewer hyperparameters to adjust. Thus, it trains the model faster than the LSTM network (Toharudin et al., 2020).

3.4 Violent Video Datasets

We evaluated the proposed method on a benchmark violent video dataset that was collected from hockey games of the national hockey league (NHL) in North America, namely the hockey fight dataset (Bermejo et al., 2011). The hockey fight dataset includes two classes and contains 500 violent videos and 500 without violence. Each hockey video consists of 41 frames with 720x576 pixels resolution. Examples of violent and non-violent videos are shown in Figure 27.



(a)

(b)

Figure 27 *Some examples of (a) violent video and (b) non-violent video of the hockey fight dataset.*

3.5 Experiment Setup and Results

3.5.1 Experiment Setup

We implemented the proposed framework using Keras API based on the TensorFlow backend. All experiments were performed on Windows OS with Intel Core i9, 32GB of RAM, and NVIDIA RTX2070 GPU. I first used a pre-trained model of four state-of-the-art CNN architectures to train on the hockey fight dataset, including MoblileNetV1, MobileNetV2, ResNet50V2, and NASNetMobile. The hyperparameters of the CNNs were set as follows: SGD optimizer, the momentum of 0.9, batch size of 4, and train with 100 epochs. We also performed different learning rates (0.01, 0.01, 0.001, 0.0001, and 0.00001) to find the lowest loss value while training. To extract the deep features, I then deleted the last layer of each architecture, which was the fully connected (FC) and softmax layers and replaced it with three layers: global average pooling (GAP), batch normalization (BN), and time distribution layers. Second, the deep features were sent to the recurrent neural networks (RNNs), including LSTM, GRU, and BiLSTM. The softmax function was used as a classifier. The hockey fight dataset was divided into training and test sets that contained 750 and 250 videos, respectively.

3.5.2 Experiments with Frames Selection

To show the performance of the CNN and RNN architecture on the hockey fight dataset, we proposed to use the MobileNetV2 architecture to train and

extract deep features from all frames, which was 40 frames for each video. Subsequently, the deep features were combined with the LSTM network, called MobileNetV2-LSTM. We trained the MobileNetV2-LSTM model for 12 hours and 19 minutes. The result showed that it achieved 93.73% accuracy on the test set.

Existing violence recognition systems were designed to extract 16, 20, and 40 frames from the video (Carneiro et al., 2019; Ditsanthia et al., 2018; Keçeli & Kaya, 2017; Soliman et al., 2019). In this experiment, we trained MobileNetV2-LSTM by choosing only 16 frames from the video. Consequently, we experimented on choosing the key frame from different frame numbers (seeTable 3). As a result, the computational time was reduced and was three times faster than when training with 40 frames. It trained approximately four hours. The accuracy results of different frame numbers are shown in Table 3. I compared four keyframe numbers (see Table 3, Experiments 1-4). It can be seen from Table **3** that frame numbers 5, 7, 9, ..., 35, which are 16 frames, are the best keyframes in our experiments on the hockey fight dataset. It obtained 88.80% on the test set.

Table 3 *Experimental results with different frames using MobileNetV2-LSTM.*

Experiments	Frame Numbers	Accuracy (%)
1	1 - 16	83.20
2	13 - 28	87.60
3	25 - 40	88.00
4	5, 7, 9, ..., 35	**88.80**

Discussion of Experiments with Frames Selection. We found that the best performance was obtained when selecting non-adjacent frames. However, when the non-adjacent frames were selected, the CNN-LSTM model was trained from the redundant information. For the hockey fight dataset, we then selected every two frames. Also, training the CNN-LSTM model using 16 keyframes was much faster than training with the whole frames. An example of the adjacent and non-adjacent frames is illustrated in Figure 28.

(a)

(b)

Figure 28 *Illustration of the (a) adjacent and (b) non-adjacent frames of the hockey fight dataset.*

3.5.3 Experiments with different CNN architectures.

As with the experimental results described above, the best frames were selected from the frames selection experiment, including 16 frames of frame numbers 5, 7, 9, ..., 35. We evaluated the performance of the CNNs and LSTM using four state-of-the-art CNN architectures: MobileNetV1, MobileNetV2, ResNet50V2, and NASNetMobile. The different learning rates were examined and only the best learning rate was reported for each CNN in this experiment. For evaluation, the training set was used for 5-fold cross-validation (5-cv) to avoid overfitting and the test set was for final evaluation.

We present the experimental results with various CNN architectures combined with the LSTM network in Table 4. MobileNetV2-LSTM achieved an accuracy of 92.76% with cross-validation on the hockey fight dataset and 91.60% on the test set. Results also significantly outperformed the other CNN-LSTM models (t-test, $p < 0.05$). The MobileNetV2-LSTM spent around 21 minutes and 7 seconds for the training and test times, respectively. In contrast, the very deep networks (ResNet50V2 and NASNetMobile) performed worse on accuracy, computation, and biggest model size than others. MobileNetV2-LSTM has the best FLOPS value of 77 and the fewest parameters of 13M, G is 10^9 and M is 10^6 to measure the computing performance.

Table 4 *The average accuracy (%) and the standard deviation of CNN architectures combined with the LSTM network obtained on cross-validation and test sets.*

Models	Learning Rate	5-CV	Test Accuracy (%)	Training Time (~mins)	Testing Time (~sec/video)	Model Size	FLOPS (G)	Params (M)
ResNet50V2-LSTM	0.01	77.33 ±0.0472	77.60	33	11	170	893	9.4
NASNetMobile-LSTM	0.0001	82.67 ±0.0550	87.60	31	34	94	147	
MobileNetV1-LSTM	0.00001	92.00 ±0.0354	92.00	22	5	89	146	
MobileNetV2-LSTM	0.0001	92.76 ±0.0369	91.60	21	7	86	77	

We found that the proposed CNN-LSTM architectures can address the overfitting problem because the accuracies of the 5-cv and test set were not different. With the MobileNetV2 architecture, a very small learning rate value was used to reach the lowest loss value. Further, the computational time decreased when the lightweight CNNs (MobileNetV1 and V2) were performed. In the following experiments, MobileNetV1 is proposed in combination with different RNN architectures: LSTM, BiLSTM, and GRU.

3.5.4 Experiments with Fusion MobileNets and RNN Architectures

To examine the effect of the combination between MobileNets and RNN architectures, we combine the deep features extracted using MobileNetV1 and MobileNetV2 with concatenating and adding operations. Then, the deep combination features were transferred to the RNN architectures and classifier with a softmax function. Furthermore, the proposed model was trained with 1,000 epochs.

We present the accuracy results of the combined operations, including concatenating and adding, as shown in Table 5. We also compared the fusion MobileNet and RNN architecture results with the experiments in Section 3.5.2. The fusion MobileNet and RNN models outperformed the single CNN models by approximately 2% on the test set. However, they spent much more training time, because they had to train on both MobileNet architectures. It can be seen from Table 5 that the concatenating operation created robust deep features with the size of 16x2048 and achieved better accuracy when combining MobileNet models with BiLSTM

architecture. It achieved an accuracy of 95.20% on the test set of the hockey fight dataset.

Furthermore, we use the FLOPS to measure computing recognition performance. The BiLSTM with the concatenating feature has a slightly higher FLOPS value than the others, equal to 252.53G. However, the adding operation created only 16×1024 deep features and achieved 94.80% accuracy when combined with RNNs. The performance was slightly decreased (only around 0.4%) when compared with concatenating operation. Most importantly, the testing time shown was almost equal.

3.5.5 Discussion of Experiments with fusion MobileNets and RNN Architectures.

When using the combined operations: concatenating and adding, the deep feature sizes of the concatenating operation were larger one time than the adding operation. However, the training time was different, by only about one hour. The fusion MobileNet and RNN architectures can be used to classify violence from real-time because it is recognized quickly and with high accuracy. So, extending the complex architecture does not affect the recognition time.

Table 5 *The accuracy (%) and computational times of violence recognition experiments on the hockey fight dataset.*

Combined operations	**RNNs**	**Test Acc (%)**	**Training time (h:m)**	**Testing time (~sec/video)**	**FLOP (G)**	**Params (M)**	**Model Size (MB)**
Concatenating (16 × 2048)	LSTM	94.80	3:38	3	252.26	34	104
	BiLSTM	**95.20**	8:44	5	252.53	69	208
	GRU	94.00	4:6	2	252.26	26	80
Adding (16 × 1024)	LSTM	94.80	3:22	2	252.26	26	72
	BiLSTM	94.80	7:35	4	252.26	52	136
	GRU	94.40	3:36	2	252.26	20	52

3.6 Comparison of the Fusion MobileNets and BiLSTM architecture and the Existing Methods.

This section presents the experimental results of various methods, as shown in Table 6.

Table 6 *The comparison of the proposed method with existing methods.*

Methods	No. of Frames	Data splitting Train:Test (%)	Testing accuracy (%)
Multiscale convolutional features (Ditsanthia et al., 2018)	40	80:20	83.19
salient frame extraction and MobileNet (Khan et al., 2019)	N/A	75:25	87.00
Short-term traffic flow prediction (Soliman et al., 2019)	20	80:20	88.20
Multi-stream CNN (Carneiro et al., 2019)	40	90:10	89.10
Optical flow and AlexNet (Keçeli & Kaya, 2017)	20	80:20	94.40
Our Proposed Method	**16**	**75:25**	**95.20**

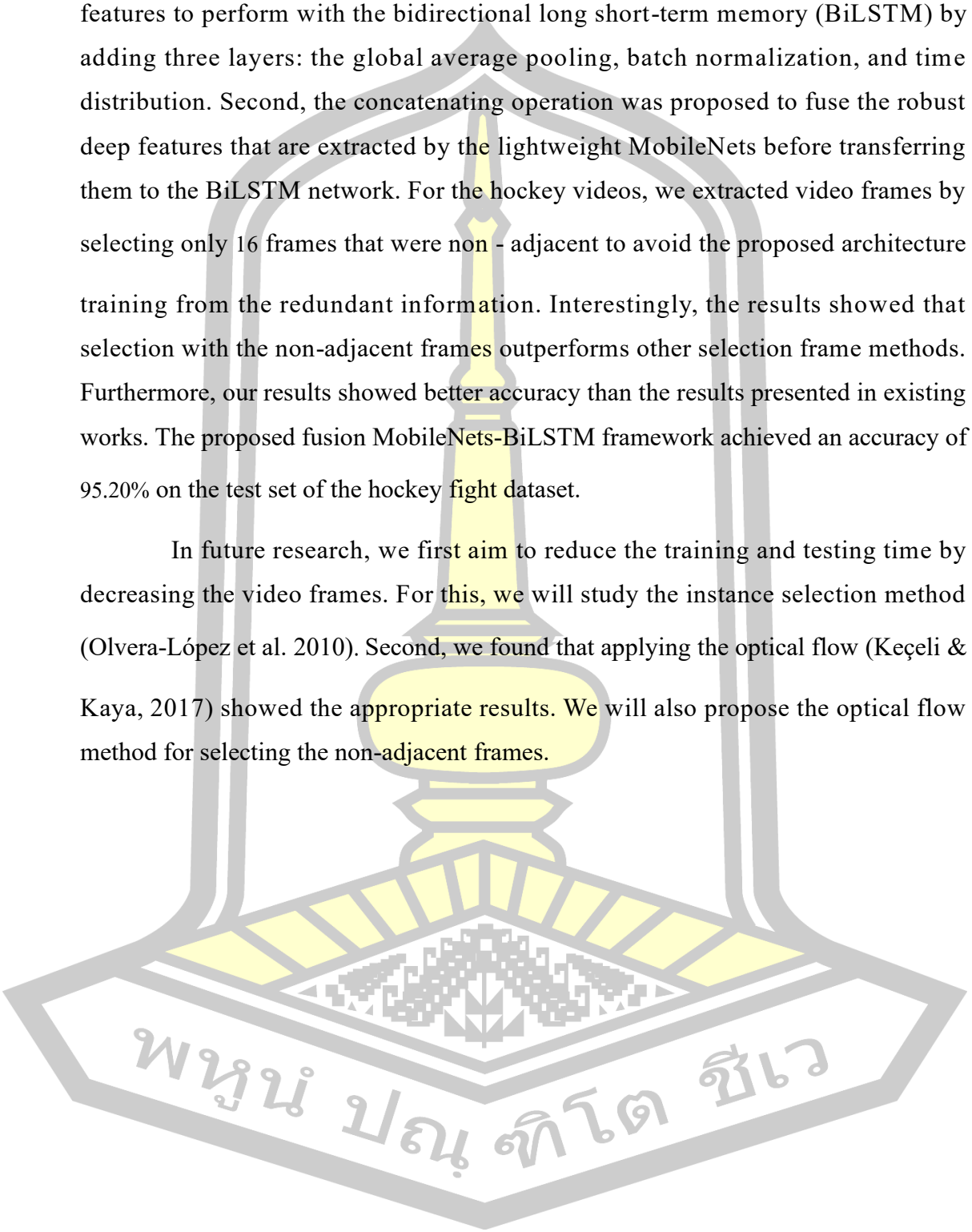
Table 6 compares the results of our proposed method with the existing methods on the hockey fight dataset. It shows that our proposed fusion MobileNets-BiLSTM architecture outperformed the existing methods with an accuracy of 95.20%. As a result, the existing method trained their models with more frames than our proposed method. The existing method trained with 20 and 40 frames, while our model trained with 16 frames. We also trained the model with less training set than the other methods, except research (Khan et al., 2019).

3.7 Conclusions

In this research, we proposed the fusion MobileNets-BiLSTM framework to recognize violent events from the sport of hockey. First, MobileNetV1 and MobileNetV2 were selected, which are lightweight convolutional neural networks

(CNNs), that aim to extract the robust deep features and then convert the deep features to perform with the bidirectional long short-term memory (BiLSTM) by adding three layers: the global average pooling, batch normalization, and time distribution. Second, the concatenating operation was proposed to fuse the robust deep features that are extracted by the lightweight MobileNets before transferring them to the BiLSTM network. For the hockey videos, we extracted video frames by selecting only 16 frames that were non - adjacent to avoid the proposed architecture training from the redundant information. Interestingly, the results showed that selection with the non-adjacent frames outperforms other selection frame methods. Furthermore, our results showed better accuracy than the results presented in existing works. The proposed fusion MobileNets-BiLSTM framework achieved an accuracy of 95.20% on the test set of the hockey fight dataset.

In future research, we first aim to reduce the training and testing time by decreasing the video frames. For this, we will study the instance selection method (Olvera-López et al. 2010). Second, we found that applying the optical flow (Keçeli & Kaya, 2017) showed the appropriate results. We will also propose the optical flow method for selecting the non-adjacent frames.



Chapter 4
Violence recognition with 3D-CNN

The technology of surveillance systems has been developing rapidly. Many places install surveillance cameras for the security of unusual events that may occur. However, monitoring violent events requires manual work and time to analyze historical files, which does not allow immediate action to stop the incident. Deep learning is a powerful technique that can extract important features for discriminative recognition. It can also construct models with high accuracy for application in various domains. In this work, we proposed an effective method for recognizing violent videos using deep learning techniques. The proposed method comprises two main parts, the deep feature extraction and integration part and the 3D convolution part. For the deep feature extraction, we used MobileNetV1 and MobileNetV2 to extract spatial features from individual frames separately. Then, the obtained features are integrated with concatenate operation before passing through the 3D convolution. The 3D convolution considers temporal information between adjacent feature frames and performs violent classification using softmax. The performance of the proposed method is evaluated using three violence datasets, including hockey fight, movie, and violent flow. The result achieves an accuracy of 97.60%, 100%, and 96.77%, respectively. The result indicates that the proposed method is efficient compared to other proposals for violent recognition.

4.1 Introduction

Recently, the surveillance system has been developing rapidly. Various locations have cameras installed to monitor abnormal events, including surveillance of theft in the mall, attacks in the park, patient behavior tracking in hospitals, and detection of elderly falls (Rajavel et al., 2022; X. Yang et al., 2022). However, detecting abnormalities is a human manual for analyzing and detecting visual information. Therefore, detecting abnormal events using humans must be more accurate and impractical (Jahlan & Elrefaei, 2022). Sometimes, anomaly detection occurs after a strange event has occurred, and it is impossible to notify in time.

Unusual circumstances include physical abuse, punching, robbery, robbery, accidents, etc. For the safety of human beings and the avoidance of violence. Therefore, an automated surveillance system is crucial developed to monitor human behavior at risk of violence from surveillance cameras. However, it is challenging to differentiate the violence in the video since similar to a typical gesture. Violent activity contains different activities such as fighting, beating, punching, and attacking people. However, Video recognition differs from image recognition in that each video requires multiple frames to extract features. Therefore, videos with high frame rates are also time-consuming. In addition to different viewpoints, scale, video resolution, the number of people in the area, the crowd scene, and the dynamic scene will significantly affect the recognition performance, making action recognition more challenging to capture practical and discriminative features.

Many researchers proposed methods to improve the effectiveness of video violence recognition (Das et al., 2019; Gao et al., 2016; Souza et al., 2010). In literature, the basic process of violent recognition is divided into feature extraction and classification. Several years ago, feature extraction used a hand-craft method consisting of local and global feature extraction to recognize the violence in surveillance video. For example, Souza et al. (2010) proposed a violence detector based on the local spatiotemporal feature. Das et al. (2019) used a histogram of oriented gradients method to extract the edges of gradient and orientation in localized portions of an image. Some studies proposed methods for global feature extraction. For example, Gao et al. (2016) improved the violent flow feature descriptor to use the orientation information of optical flow, namely oriented violent flow which considers both magnitude and orient information. The obtained features are then encoded into the bag of words. Finally, a classifier, such as a support vector machine, is adopted to recognize the violence in the video.

Deep learning is a core technology popularly applied to many fields within machine learning today due to its learning capabilities from the given data (Sarker, 2021). A convolutional neural network (CNN) is one of the most influential networks for deep learning. Many researchers employed CNN for robust deep feature extraction. Khan et al. (2019) proposed lightweight deep learning to extract spatial features of frames and classified them by the softmax function. Carneiro et al. (2019)

proposed pre-trained VGG-16 to generate spatial features, temporal features, rhythm features, and depth information of video for violence detection. The result showed that the method improves the recognition efficiency derived from learning from training models. Also, Soliman et al. (2019)proposed the pre-trained VGG-16 and long short-term memory (LSTM) to extract spatial and temporal features of the video, respectively. In addition, other research applied 3D-CNN for spatiotemporal extraction, such as Ullah et al. (2019) using 3D- CNN to learn complex sequential patterns to predict violence in surveillance video streams to achieve good recognition performance. Li et al. also obtained an effective recognition model when using 3D-CNN for spatiotemporal feature extraction for multiplayer violence.

This research proposed a method to recognize violent video using deep feature integration and three-dimension convolution. First, each frame extracted features with two CNN models, including MobileNetV1 and MobileNetV2, at the last convolution layer. Then, we integrated the features vector with concatenate operation to represent the video feature vector. The video feature was learned with the proposed three-dimensional convolution. Finally, we use a softmax function to classify violent or non-violent videos. We perform on three challenge violent recognition in video datasets, namely hockey fight, movie, and violent flow, to verify the effectiveness of our method.

4.2 Related work

4.2.1 Recognition of violence in surveillance video

In an early study, recognition of violence in surveillance video focused on a handcrafted approach for feature extraction, which can distinguish violence from nonviolence. Then, aggregate the features using encoding strategies and apply machine learning as a classifier (Li et al., 2019). A histogram of oriented gradients (HOG) extracts features from an image. The technique is used to count the occurrences of the gradient in the localized portions of an image. Dalal and Triggs (2005) applies a histogram of gradient orientation features for person detection and uses SVM as a classifier, which achieves good results. Correspondingly, Patil et al. (2017) used a histogram of gradient orientation feature descriptor to extract features and an SVM classifier to recognize human activities, providing good recognition

results with a minimum number of false detections. Sun et al. (2019) proposed a multi-view maximum entropy discriminant model to extract scale-invariant feature transform, histogram of oriented gradient, local binary patterns, and color histogram features from the image and combine various features for violence recognition of static images. Das et al. (2019) proposed a system to detect violence from video, applied HOG as a feature descriptor to extract features from the images, and employed various classifier models and a majority voting technique to decide whether a video clip contains violence. The result shows the system is robust enough to detect violence in different surveillance situations.

With the continuous development of violence recognition, many studies have analyzed motion features to encounter motion in video frames. Souza et al. (2010) presented a violence detector based on local spatiotemporal features with a bag of visual words and a support vector machine. The results confirm that motion patterns are crucial to distinguish violence from regular activities compared to visual descriptors in the space domain. Hassner et al. (2012) present a violent flow feature descriptor based on optical flow magnitude changes between adjacent violent video frames. Gao et al. (2019) improved the violent flow feature descriptor to use the orientation information of optical flow, namely oriented violent flow, which considers both magnitude and orient information. The features are encoded into the bag of word representation and a support vector machine for violence in the video classifier.

Bermejo et al. (2011) introduced a fight dataset and used space-time interest points and motion scale-invariant feature transform method to extract spatial-temporal features. Then, the feature vector is sent to the support vector machine classifier. Similarly, Xu et al. (2014) used the motion scale-invariant feature transform method to extract the low-level description of a query video. The kernel density estimation is exploited for feature selection to obtain the highly discriminative video feature. A sparse coding method with a max pooling procedure generates a discriminative high-level video representation from local features. The result shows that the proposed method outperforms violence detection in crowded and non-crowded scenes.

4.2.2 Deep neural networks

Deep learning has recently been widely used to train deep neural networks as robust feature extractors for violence recognition (Khan et al., 2019; Tian et al., 2021; P. Zhou et al., 2017). A convolutional neural network (CNN) is a deep learning architecture that extracts valuable information using convolution operation (Tyagi et al., 2022). Khan et al. (2019) presented a violence detection scheme for movie. The frame is selected based on the saliency score and applied by MobileNet to classify violence and non-violence. Then, all non-violence scenes are combined sequentially to generate a violence-free movie. This method obtained recognition performance of 87.00%, 99.5%, and 97.0% on the hockey fight, the movie dataset, and violence scene detection datasets, respectively. Some research uses deep learning to learn from various features type (Tian et al., 2021; P. Zhou et al., 2017).

P. Zhou et al. (2017) constructed ConvNets, namely FightNet, to model long-term temporal structures for recognizing violence. The input consists of an RGB image, optical flow, and acceleration field to extract the motion information better. Their approach demonstrates that deep ConvNets could capture more essential features and detect violence accurately. Carneiro et al. (2019)used VGG-16 for a multi-stream that includes spatial, temporal, rhythm, and depth information. The model achieved an accuracy of 89.10% on the hockey fight dataset and 100% on the movie dataset. The results showed that multi-stream methodology increased the efficiency of violent video recognition. Celard et al. (2023) proposed the CNN to recognize and classify violent events. This research evaluated the computational performance of the CNN architectures for automatic violence recognition, such as SqueezeNet, Inception, MobileNetV1, MobileNetV2, and NASNetMobile. The experiment shows that a high classification accuracy of 92.05% can be achieved using mobile architectures compared to VGG16, InceptionV3, and ResNet50 architecture.

Moreover, considering spatial and temporal feature extraction using 2D-CNN followed by long short-term memory (LSTM) (Hanson et al., 2019; Naik & Gopalakrishna, 2021; Soliman et al., 2019; Sudhakaran & Lanz, 2017; Sumon et al., 2019). Sudhakaran and Lanz (2017) use frame difference as input of a 2D-CNN to extract hierarchical features from the video frames and are then aggregated using the convLSTM layer. Then classify violence or non-violence with a fully connected layer.

The experimental result shows that a deep neural network trained on the frame difference performs better than a model trained on raw frames. Soliman et al. (2019) proposed the pre-trained VGG-16 model on ImageNet to extract spatial and LSTM to extract temporal features before being classified by a fully connected layer. Experiments on standard violent data sets show that the model outperforms the state-of-the-art approach. Besides, they created a real-life violence situations (RLVS) dataset for fine-tuning the model, achieving the best accuracy of 88.2% on the hockey fight dataset.

Sumon et al. (2019) demonstrated the efficiency of the deep learning method by using CNN, LSTM and combining CNN with LSTM. The experiment on violent video datasets finds that the CNN model with transfer learning has performed better than LSTM and CNN-LSTM models. Naik and Gopalakrishna (2021) proffered the deep neural network model Mask Region-based Convolutional Neural Network (Mask RCNN) to detect a single person in the video and extract interest points. Then the extracted features were fed to LSTM for feature learning across a time series frame. The results showed that the model had excellent performance. Hanson et al. (2019) proposed the spatiotemporal encoder to detect video violence. First, each video frame was extracted as feature maps with the VGG13 network. Then the feature maps were passed to BiConvLSTM to extract the temporal information by passing forward in time and reverse. Finally, elementwise maximization is applied to represent the video and classified as violent or non-violent in the video. The testing accuracy achieved 96.96% on the hockey fight dataset, 100% on the movie dataset, and 90.6% on the violent flow dataset.

4.2.3 Spatial and temporal feature extraction

Feature extraction is an important step in deep learning, extracting essential features from the input data and reducing the dimension and computational cost (Humeau-Heurtier, 2019). Traditional video recognition is usually based on the geometric features manually extracted from video frames, which are difficult to apply to complex scenarios and cannot gain high accuracy recognition and robustness. Deep learning is outstanding at extracting spatial features of images and extracting features of frames for video when compared to machine learning (Pu et al., 2022). The feature extraction in video recognition tasks differs from image recognition, which can use

spatial feature extraction independently, which needs to be improved to sustain the learning effectively. Thus, temporal features are mainly considered to analyze information regarding the duration of adjacent frames.

Various researchers have proposed several spatial and temporal feature extraction methods for video recognition. Sun et al. (2022) proposed a deep learning model for user-generated content video quality assessment that extracts both a spatial and temporal feature. The spatial features extracted from the end-to-end model employed raw video frame pixels as input. The temporal features extractor uses a pre-trained action recognition network to represent motion information. Then, the multilayer perception layer network is used to regress into chunk-level quality scores, and the temporal average pooling strategy is adopted to obtain the video-level quality score. The experimental results show that the proposed model outperforms five popular user-generated content video quality assessment databases.

J. Yang et al. (2022) introduce a recurrent vision transform framework to achieve the video action recognition task. The recurrent vision transform framework can capture spatial and temporal features by attention gate and recurrent execution. The attention gate can build interaction between the current frame input and the previous hidden state. The result demonstrates that the recurrent vision transform framework can achieve state-of-the-art performance on various datasets for the video recognition task. Some researchers combined convolutional neural networks and long short-term memory models to recognize the video frame sequence. Chen et al. (2023) used the VGG16 and LSTM network to recognize video-recorded actions performed in a traditional Chinese exercise. The result shows that the CNN-LSTM recognition model outperforms manually extracted features in the conventional action recognition model, and the CNN model is more effective in improving classification accuracy.

For violent video recognition, the convolutional network has extraction ability for the deep features from low-level to high-level features. Some research used deep learning to extract the spatial features from video frames after the preprocessing step, such as Sharma et al. Sharma and Sungheetha (2021) proposed a hybrid framework based on a fusion of CNN and SVM to detect abnormal incidents in video surveillance. This proposed method consists of data preprocessing using the

background subtraction technique, spatial features extraction using the CNN architectures, and classification by SVM. The experimental result demonstrates that the proposed method provides good accuracy, higher efficiency, and less loss than other combination and single classifiers. Many researchers proposed an approach for spatial and temporal feature extraction in recognition of violent video.

Vosta and Yow (2022) introduce a model for detecting abnormal events in a surveillance camera using CNN and LSTM. This research divided the video into 20 frames and extracted essential features of each frame with ResNet50. The extracted features are fed into the ConvLSTM network, and normal and abnormal events are classified on the UCF-crime dataset. The results show that the proposed method achieved 81.71% AUC, higher than the C3D model on the same dataset. Jahlan and Elrefaei (2022) proposed a novel approach using the fusion technique to detect violence. First, both Alexnet and SqueezeNet networks are followed by Convolution Long Short-Term Memory (ConvLSTM) to extract robust features from the video. Then, the obtained features were fused and fed into the max-pooling layer, fully connected layer, and softmax classifier.

Recent studies have developed models that extract spatiotemporal features from 3D convolution neural networks (Hu et al., 2020). Maqsood et al. (2021) proposed a framework for recognizing anomaly videos by learning spatiotemporal features using a deep 3-dimensional convolutional network. The experiment was trained on the University of Central Florida (UCF) Crime dataset. The proposed approach consists of 3D feature extraction and spatial augmentation by the proposed 3D ConvNet. The result shows that the 3D ConvNet outperforms the state-of-the-art method on anomalous activity recognition, having 82% AUC. Pratama et al. (2023) proposed the two-stream 3D ResNet-18 network for violence video classification. The two-stream 3D-CNN has two inputs, including RGB and the optical flow frame of the video. Each stream is separately trained with different configurations for extracting temporal information using 3D convolution and 3D pooling. Then, combine the output of both streams, achieving an accuracy of 90.5% on the RWF-2000 dataset. The result indicates that the 3D ResNet-18 shows robust performance in video-based violence classification.

Keçeli and Kaya (2017) proposed an approach for automatically classifying violent video using a combination of a three-dimensional convolutional neural network and transfer learning. The proposed approach consists of three main parts, including person detection, spatial feature extraction, and temporal feature extraction. Person detection involves detecting and removing frames that do not contain a person. Then, AlexNet is used to extract features from a fully connected layer with a dimension of 4096. The extracted features of the individual frame are concatenated to construct a feature volume and are reshaped as two-dimensional before feeding to a 3D-CNN model, designed to capture the temporal features from the video and classify them by softmax layer. The experimental results show that better results are obtained than LSTM and biLSTM.

4.3 Proposed framework

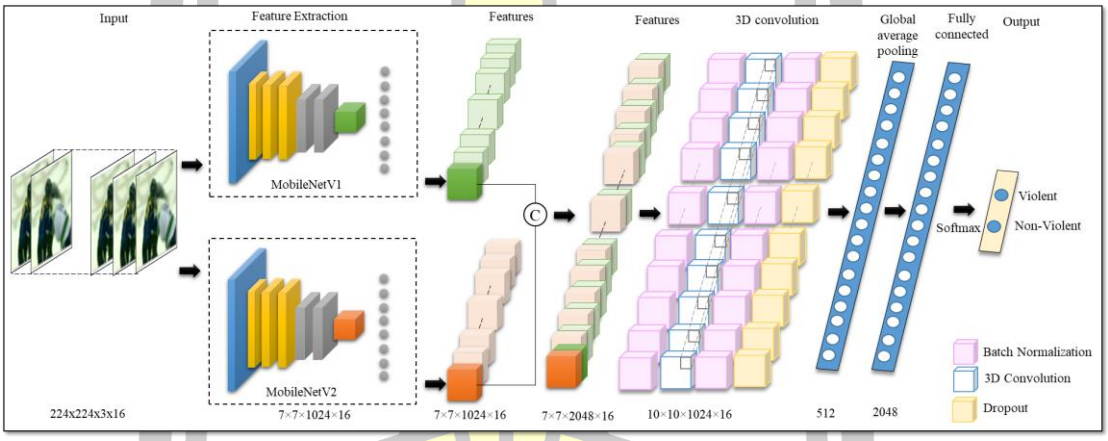


Figure 29 *The proposed framework deep features integration with 3D convolutional to recognize the violent video.*

This section describes the proposed deep features integration with 3D convolution to recognize the violent video. We divided the proposed framework into frame-level deep feature extraction and integration and deep feature learning with 3D convolution. First, the frame-level deep features were extracted with two pre-trained CNN models from the last convolution layer, which are MobileNetV1 and MobileNetV2. Next, the obtained features were integrated with concatenate operation to represent the video-level feature. Then, the video-level features were learned with the proposed 3D convolution consisting of batch normalization, 3D convolution, and

dropout layers. In addition, we employ a global average pooling layer, which is an effective pooling operation to reduce the total number of deep features, followed by a fully connected layer. Finally, the softmax function classified each video-level feature as a violent and nonviolent video. The proposed framework is shown in Figure 31.

4.3.1 Deep features extraction

Deep feature extraction is the main task to find out the robust features. The 2D-CNN model was widely used to be an effective extractor for image and video recognition. We consider the deep feature extraction in the frame level for violent video which each frame was extracted by the two pre-trained 2D-CNN models including MobileNetV1 and MobileNetV2, explained structure as follows.

4.3.1.2 MobileNetV1 (Howard et al., 2017)

MobileNetV1 is a streamlined architecture that uses depthwise separable convolutions to build lightweight deep convolutional neural networks. It provides an efficient model for mobile and embedded vision applications. Standard convolutions both filtering and combining input to produce a new representation. However, MobileNetV1 split convolution into two layers called the depthwise separable convolutions, including depthwise convolution and pointwise convolutions separately. The standard convolution kernels are $D_K \times D_K \times M \times N$ where $D_K \times D_K$ is a dimension of the kernel, M is a number of channels, and N is the number of outputs. The computational cost of the standard convolution is $D_K \times D_K \times M \times N \times D_F \times D_F$ where $D_F \times D_F$ is the feature map size, as illustrated in Figure 30 (a).

The depthwise convolution kernels are $D_K \times D_K \times M$ where M is channels of input. The output of a depthwise convolution is a features map of each input channel, as illustrated in Figure 30(b). In pointwise convolutions, is a combining layer for creating new features by 1×1 convolution kernel, as illustrated in Figure 30(c). The network layer of MobileNetV1 starts with the convolution layer, followed by 13 depthwise separable convolution layers. A Batch Normalization (BN) and Rectified Linear Unit (ReLU) activation function follow all layers in the network. Subsequently, the Global Average Pooling layer is used for reducing extracted feature map size. Finally, the reduced feature map is fed into a fully connected layer and

classification by a softmax activation function. The structure of the MobileNetV1 is illustrated in Figure 31 (a).

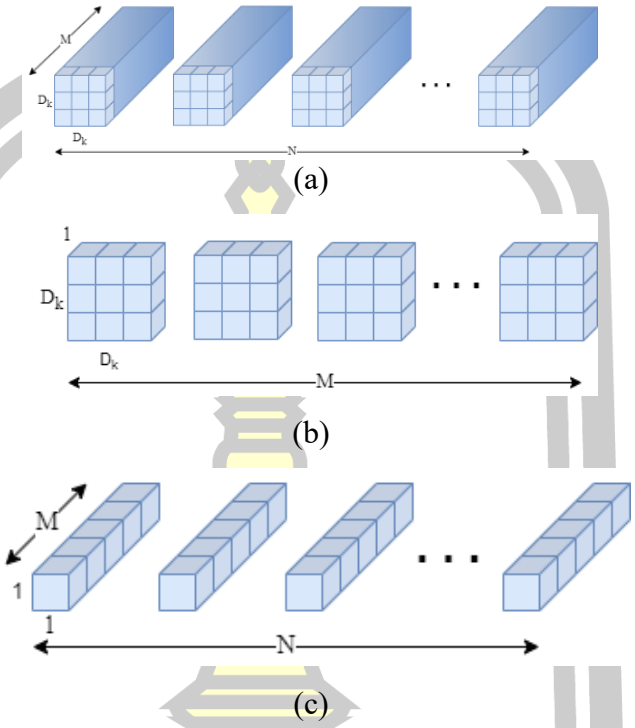


Figure 30 *show (a) the standard convolution kernel (b) the depthwise convolution kernel and (c) pointwise convolution kernel. (Howard et al., 2017)*

4.3.1.2 MobileNetV2 (Sandler et al., 2018)

MobileNetV2 is also a lightweight CNN architecture for mobile devices. MobileNetV2 proposed the concept of inverted residuals and linear bottlenecks based on MobileNetV1. The inverted residual concept has three separate convolutions. First, a pointwise (1×1) convolution is used to expand the dimensional input feature map to a higher dimensional with ReLU6 is applied. Next, a depth-wise convolution is performed using 3×3 kernels, followed by ReLU6 activation. Finally, the spatially filtered feature map is reduced dimensional using another pointwise convolution, and the linear is used instead of ReLU to avoid information loss. Figure 31 (b) show the structure of MobileNetV2. The first layer of MobileNetV2 is the convolution layer, followed by 19 residual bottleneck layers that consist of expanding convolution, depthwise convolution, and projection convolution. The Batch Normalization layer and ReLU6 activation function are applied to all layers except the

projection convolution layer. The last two layers are the global average pooling to reduce feature map size and softmax classifier.

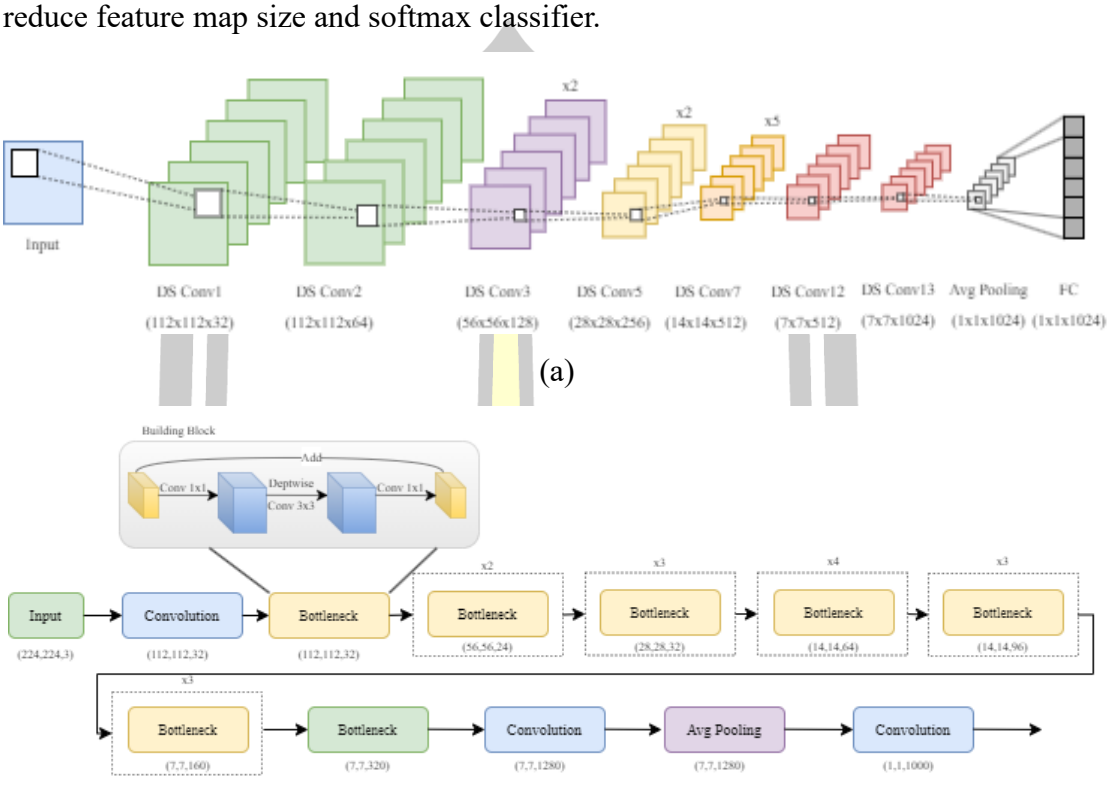


Figure 31 *The structural of CNN (a) MobileNetV1 and (b) MobileNetV2 (Howard et al., 2017; Sandler et al., 2018)*

In addition, 3D-CNN is used as the extractor for video-level features to observe the difference in performance. 3D-CNN has the ability to model temporal information better than 2D-CNN due to 3D convolution and pooling operations. 3D convolution and pooling operations are performed spatial and temporal, whereas 2D-CNN only learn spatially.

3D convolution is performed over multiple frames cascaded in the temporal dimension. The 3D convolution operation as shown in (1) and the obtained feature map is shown in equation (1).

$$conv(I,K)_{x,y,z} = \sum_{i=1}^{n_F}\sum_{j=1}^{n_H}\sum_{k=1}^{n_W}\sum_{l=1}^{n_C} K_{i,j,k,l}, I_{x+i-1,y+j-1,z+k-1,k} \quad (1)$$

where the kernel $K(f_f, f_h, f_w, n_C)$ convolve with the image $I(n_F, n_H, n_W, n_C)$ of different size but of similar number of channels n_C and generate a feature map $Feat\_map(o_F, o_H, o_W, \mathcal{Z})$. The f_f, f_h, f_w represent the frame, height, and width of the

kernel. The $o_F, o_H, o_W, \mathcal{Z}$ represent the frame, height, width, and number of filters of output feature map. And n_F, n_H, n_W denote the frame, height, and width of the given image.

$$Feat_{map(o_F,o_H,o_W,\mathcal{Z})} = \left(\left\lfloor \frac{n_F+2_p-f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_H+2_p-f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_W+2_p-f}{s} + 1 \right\rfloor, \mathcal{Z}\right) \quad (2)$$

Convolutional three-dimension (C3D) architecture is the preferred architecture for video recognition. The details of the architecture C3D are as follows.

4.3.1.3 C3D (Tran et al., 2014)

C3D is a deep three-dimension convolutional neural network for spatiotemporal feature learning of video data. The C3D can learn both spatial features and temporal features from continuous frames by using 3D convolution and 3D pooling operation. The architecture of C3D consists of 8 convolutions, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are size 3×3×3 with stride 1 in both spatial and temporal dimensions. The convolution has a number of filters such as 64, 128, 256, 256, 512, 512, 512 and 512 respectively. All pooling kernels are 2×2×2, except for pool1 is 1×2×2. The fully connected layer has 4096 output units. The architecture of C3D as shown in Figure 4. The number of parameters and FLOPS of C3D pre-trained model which were trained on Sports-1M dataset (Maqsood et al., 2021) with input size 16×112×112 as shown in Table 7.

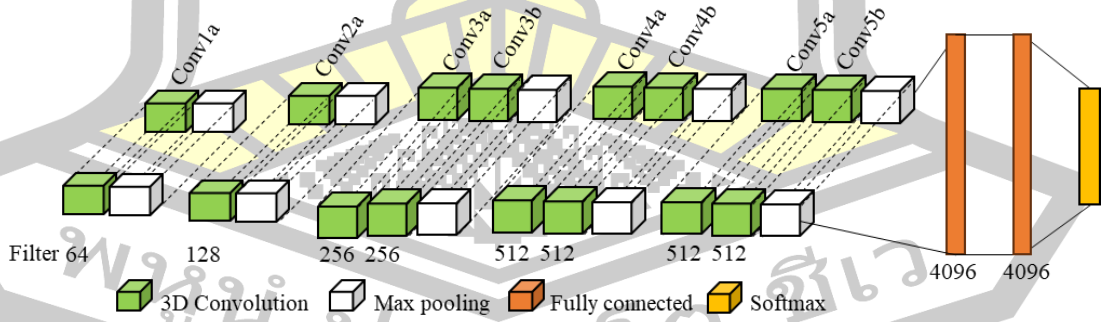


Figure 32 The architecture of C3D (Tran et al., 2014).

***Table** 7 The number of the parameters in all layers of C3D architecture.*

Layer	Input size	Output Size	Parameter (M)
Conv1	16×112×112×3	16×56×56×64	0.005
Conv2	16×56×56×64	8×28×28×128	0.22

Layer	Input size	Output Size	Parameter (M)
Conv3a,3b	8×28×28×128	4×14×14×256	2.65
Conv4a,4b	4×14×14×256	2×14×14×512	10.62
Conv5a,5b	2×14×14×512	2×7×7×512	14.16
Fc6		4096	33.56
Fc7		4096	16.78
Total			78
FLOPS (G)			521

4.3.2 Three - dimensional convolution neural network (3D-CNN)

Generally, 2D-CNN is suitable for image processing and practical in extracting only spatial features. When applied to video, it works with RNN-based networks to understand sequential data, while 3D-CNN is designed for video analysis, providing both spatial and temporal information in video. Therefore, we proposed a 3D-CNN architecture consisting of the batch normalization layer, 3D convolution layer, dropout layer, and global average pooling layer. The kernel sizes in the 3D convolution layers are 1×2×2, a stride of 1, and the filters are 1024. Next, we use the global average pooling layer to reduce the feature size to 512, followed by a fully connected layer. Finally, the output of the last layer is then passed to a dense layer of 2 neurons with a softmax activation function for violent or non-violent classification. The proposed 3D convolutional neural network structure, parameter, and FLOPS are shown in Table 8.

Table 8 *Network architecture of the proposed 3D convolutional neural network.*

Layer	Kernel size	Input size (F×W×H×D)	Output Size (F×W×H×D)	Parameter (M)
Batch Normalization	-	16×7×7×2048	16×7×7×2048	0.008
3D Conv	1×2×2	16×7×7×2048	16×6×6×1024	8.389
Batch Normalization	-	16×6×6×1024	16×6×6×1024	0.004
Dropout	-	16×6×6×1024	16×6×6×1024	-
Global Average Pooling	-	16×6×6×1024	512	-
Fully connected	-	512	2048	2.099
Softmax	-	2048	2	0.004

Layer	Kernel size	Input size (F×W×H×D)	Output Size (F×W×H×D)	Parameter (M)
Total				10.504
FLOPS (G)				521

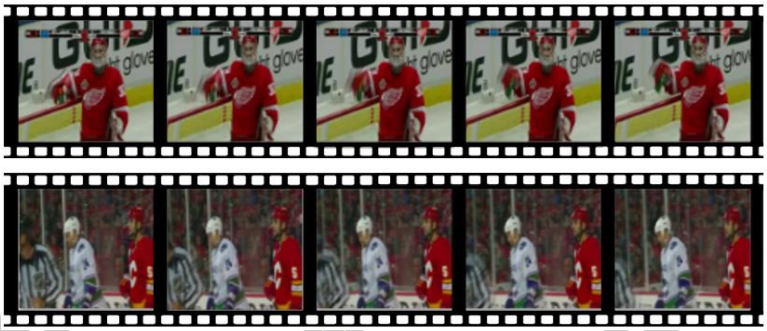
4.4 Violence Datasets

We evaluate our proposed approach on three benchmark violent video datasets, including hockey fight, movie, and violent flow datasets. The datasets were categorized into two classes: violent and non-violent classes. The hockey fight and violent flow are collected from sporting events such as hockey and soccer. At the same time, the movie dataset is collected from movie that have violent scenes. We describe the details of each data set as shown below.

4.4.1 Hockey fight dataset (Souza et al., 2010)

The hockey fight dataset contains 500 video clips for the fight and 500 without the fight. It was collected from hockey games of the National Hockey league in which each video consists of 41 frames and a resolution of 720×576 pixels. The dataset is categorized into training and testing from the perspective of two classes, including violence and non-violence. The sample frame from the hockey fight dataset is shown in Figure 33.

(a)

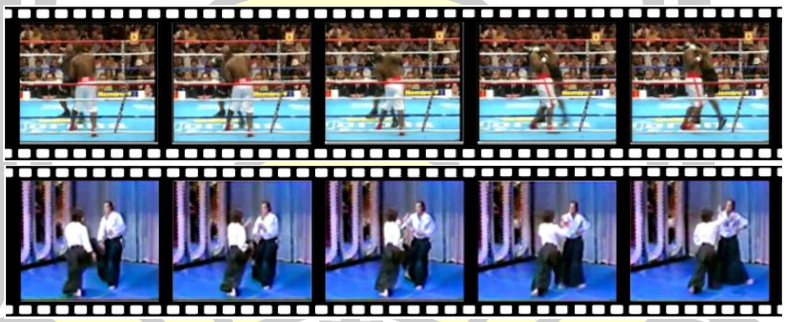


(b)

Figure 33 *Samples of hockey fight dataset, (a) violence video and (b) non-violence video.*

4.4.2 Movie dataset (Souza et al., 2010)

The movie dataset consists of 200 videos collected from action movie. The violence class 100 videos were collected from action movie scenes, while the non-violence was collected from other publicly available action recognition datasets that do not contain violent action. The duration of each video clip is around 2s. The sample frame from the movie dataset is shown in Figure 34.



(a)

(b)

Figure 34 *Samples of movie dataset, (a) violence video and (b) non-violence video.*

4.4.3 Violent flow dataset (Bermejo et al., 2011)

The violent flow dataset consists of 246 videos that contain crowds of scenes of fight between persons. The videos were collected from violent situations that occur in football matches. The sample frame from the violent flow dataset is shown in Figure 35.

(a)

(b)

Figure 35 *A samples of violent flow dataset, (a)violence and (b) nonviolence video.*

4.5 Experimental results and discussion

This section explains the proposed deep features concatenate and 3D convolution for violent recognition. This research works on violence datasets consisting of hockey fight, movie and violent flow. First, we exploit the effectiveness of feature extraction with a pre-trained CNN model, including MobileNetV1, MobileNetV2, and C3D. Second, we assume that combining the deep features and learning through 3D-CNN will enable violent video recognition effectively. Accordingly, we combine the frame-level deep feature to produce robust deep features at the video level. Third, our proposed 3D convolution for violent recognition learned the deep features obtained. Finally, we compare our method with other violence recognition methods.

4.5.1 Experimental setting

The proposed method is implemented by the Python programming language based on Keras API with TensorFlow as backend. All experiments were performed on Intel(R) Xeon(R) 2.00 GHz CPU, Tesla T4 GPU, and RAM 26 GB. We trained our models using the Stochastic Gradient Descent (SGD) optimizer with different learning rates (0.01, 0.001, and 0.0001). The momentum of the SGD was set to 0.9. The batch size is set to 4 and 8, and the model training 500 epochs. For violence recognition on each video, we decided to use 16 frames as input to reduce computational time. The datasets are randomly divided into 75% training and 25% testing. The following metrics are used for defining the performance of classification success:

$$R = \frac{TP}{TP+FN}, \tag{3}$$

$$S = \frac{TN}{TN+FP}, \tag{4}$$

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

above equations TP is true positive, FP is false positive, FN is false negative, R is true positive rate (sensitivity), S is true negative rate (specificity) and Acc is accuracy. Accuracy is the ratio of the number of correct predictions to the total number of test samples (between 0–1). It indicates how well a model performs, as in equation 5.

Moreover, we used the receiver operating characteristic (ROC) and area under the curve (AUC) techniques to evaluate the classification performance. The ROC represents the relation between true positive rate (sensitivity) and false positive rate (1- specificity). True positive rate defines a classifier test performance as accurately categorizing positive instances among all available positive samples throughout the test step, as in equation 3. The false positive rate determines the proportion of false-positive findings compared to the total negative samples available through the test step, as in equation 4. The AUC is used for binary classification and indicates how well a model discriminates between positive and negative target classes. This value is the area under the ROC curve. The best optimal classifier has a value of AUC close to 1.

4.5.2 The result of violent recognition with MobileNetV1, MobileNetV2, and C3D

To assess the efficiency of the pre-trained CNN model in recognizing violence within video content, we experimented with MobileNetV1 and MobileNetV2 on three violent video datasets. First, we use 16 non-overlapping frames and select frames by skipping one frame at a time to reduce data redundancy. The selected frames are sized 16×224×224×3, where 16 represents the number of frames, 224 represents width and height, and 3 describe channels. We retrained both MobileNetV1 and MobileNetV2 models on the three datasets separately and replaced the final layer with softmax for classifying violent or nonviolent video. We compared the results in different batch sizes of 4 and 8 and learning rates of 0.01, 0.001, 0.0001, and 0.00001.

First, MobileNetV1 are evaluated on three datasets. On the hockey fight, the model can achieve performance with 95.99% accuracy when using a batch size of 8 and a learning rate of 0.01,0.001, and 0.0001. On the movie, the model can achieve performance with 98.00% when using a batch size of 4 and a learning rate of 0.001. On violent flow, the model can achieve performance with 91.94% accuracy for all batch sizes and a learning rate setting, as shown in Table 9.

Table 9 *Evaluation of the violent recognition results using MobileNetV1*

Dataset	Batch size	Learning rate	Accuracy	Training Time (hr.)	Testing Time (ms)
Hockey fight	4	0.01	94.80	0.58	2
		0.001	95.99	0.59	
		0.0001	95.20	0.58	
		0.00001	95.20	0.58	
	8	0.01	95.99	0.53	2
		0.001	95.99	0.53	
		0.0001	95.99	0.53	
		0.00001	95.60	0.53	
Movie	4	0.01	95.99	0.11	1
		0.001	98.00	0.11	
		0.0001	93.99	0.11	
		0.00001	93.99	0.11	

Dataset	Batch size	Learning rate	Accuracy	Training Time (hr.)	Testing Time (ms)
	8	0.01	92.00	0.11	1
		0.001	93.99	0.11	
		0.0001	95.99	0.11	
		0.00001	92.00	0.11	
Violent flow	4	0.01	91.94	0.14	1
		0.001	91.94	0.14	
		0.0001	91.94	0.14	
		0.00001	91.94	0.14	
	8	0.01	91.94	0.13	1
		0.001	91.94	0.13	
		0.0001	91.94	0.13	
		0.00001	91.94	0.13	

shows the experiment results with MobileNetV2. On the hockey fight, the model achieved an accuracy of 95.99% when using a batch size of 4 and a learning rate of 0.00001. On the movie, the model can achieve performance with 98.00% for all batch sizes and a learning rate setting. On violent flow, the model can achieve performance with 91.94% accuracy when using a batch size of 8 and a learning rate of 0.01.

Table 10 *Evaluation of the violent recognition results using MobileNetV2*

Dataset	Batch size	Learning rate	Accuracy	Training Time (hr.)	Testing Time (ms)
Hockey fight	4	0.01	95.60	0.68	3
		0.001	95.20	0.69	
		0.0001	95.20	0.68	
		0.00001	95.99	0.53	
	8	0.01	95.20	0.60	3
		0.001	95.20	0.61	
		0.0001	95.20	0.61	
		0.00001	95.20	0.59	
Movie	4	0.01	98.00	0.17	2
		0.001	98.00	0.17	

Dataset	Batch size	Learning rate	Accuracy	Training Time (hr.)	Testing Time (ms)
		0.0001	98.00	0.17	
		0.00001	98.00	0.16	
	8	0.01	98.00	0.16	2
		0.001	98.00	0.15	
		0.0001	98.00	0.16	
		0.00001	98.00	0.17	
Violent flow	4	0.01	87.10	0.17	2
		0.001	88.71	0.16	
		0.0001	87.10	0.16	
		0.00001	88.71	0.16	
	8	0.01	91.94	0.15	2
		0.001	82.26	0.15	
		0.0001	87.10	0.15	
		0.00001	88.71	0.15	

In addition, we merged all datasets to create a larger dataset. Also, we split the data into 75% training and 25% testing and used a random split to avoid bias. We also trained MobileNetV1 and MobileNetV2 to classify violent videos separately. As in the experiment above, we configure the batch size and learning rate parameters. The models are evaluated with three testing datasets, including hockey fight, movie, and violent - flow. Table 11 shows that MobileNetV1 achieved the highest accuracy of 96.40% when using a batch size of 4 and a learning rate of 0.0001. For the movie dataset, the model achieved the highest accuracy of 98.00% when using a batch size of 8 and a learning rate of 0.00001. Finally, the experiment result in Table 12 shows that MobileNetV2 achieved the highest accuracy of 95.59%, 92%, and 83.87% on hockey fight, movie, and violent flow datasets, respectively.

Table 11 *Testing accuracy of feature-extraction with the MobileNetV1, trained with merging all datasets and testing with separate datasets.*

Dataset	Batch size	Learning rate	Accuracy (%)	Training Time (hr.)	Testing Time (ms)

Dataset	Batch size	Learning rate	Accuracy (%)	Training Time (hr.)	Testing Time (ms)
Hockey fight	4	0.01	94.80	0.81	3
		0.001	94.40	0.81	
		0.0001	96.40	0.79	
		0.00001	94.40	0.78	
	8	0.01	94.40	0.76	2
		0.001	95.20	0.76	
		0.0001	95.20	0.75	
		0.00001	94.40	0.75	
Movie	4	0.01	93.99	0.81	2
		0.001	89.99	0.81	
		0.0001	87.99	0.79	
		0.00001	92.00	0.78	
	8	0.01	95.99	0.76	1
		0.001	89.99	0.76	
		0.0001	93.99	0.75	
		0.00001	98.00	0.75	
Violent flow	4	0.01	80.65	0.81	1
		0.001	82.26	0.81	
		0.0001	79.03	0.79	
		0.00001	82.26	0.78	
	8	0.01	79.03	0.76	1
		0.001	79.03	0.76	
		0.0001	79.03	0.75	
		0.00001	82.26	0.75	

Table 12 *Testing accuracy of feature-extraction with the MobileNetV2, trained with merging all datasets and testing with separate datasets*

Dataset	Batch size	Learning rate	Accuracy	Training Time (hr.)	Testing Time (ms)
Hockey fight	4	0.01	93.99	0.93	3

Dataset	Batch size	Learning rate	Accuracy	Training Time (hr.)	Testing Time (ms)
		0.001	95.59	0.93	
		0.0001	94.40	0.92	
		0.00001	95.59	0.91	
	8	0.01	95.20	0.84	3
		0.001	95.20	0.85	
		0.0001	94.80	0.84	
		0.00001	95.20	0.86	
Movie	4	0.01	89.99	0.93	2
		0.001	92.00	0.93	
		0.0001	89.99	0.92	
		0.00001	89.99	0.91	
	8	0.01	87.99	0.84	2
		0.001	87.99	0.85	
		0.0001	92.00	0.84	
		0.00001	89.99	0.86	
Violent flow	4	0.01	82.26	0.93	2
		0.001	82.26	0.93	
		0.0001	77.42	0.92	
		0.00001	80.65	0.91	
	8	0.01	82.26	0.84	2
		0.001	82.26	0.85	
		0.0001	83.87	0.84	
		0.00001	79.03	0.86	

In addition, we also experimented with pre-trained 3D-CNN models to discover the performance of spatial and temporal features. We selected the pre-trained C3D architecture, which was trained with a large video dataset such as the sportM1 dataset. The input of C3D was fixed to be a sequence of 16 frames and a size of 112×112×3, where 112×112 represents width and height, and 3 represents dimension. Then, we define the classification layer according to the dataset into two classes violence and

nonviolence. Next, the C3D model was trained on three violent video datasets, including the hockey fight, the movie, and the violent flow dataset. Finally, the model was trained with different batch sizes of 4 and 8 and learning rates of 0.0001 and 0.00001, as shown in Table 13.

Table 13 *The accuracy results with C3D on three datasets.*

Dataset	Batch size	Learning rate	Training time (hr.)	Testing time (ms.)	Testing accuracy (%)	Model size
Hockey fight	4	0.0001	1.87	13	76.40	297.7MB
		0.00001	1.75		72.80	
	8	0.0001	1.48		70.80	
		0.00001	1.95		72.80	
Movie	4	0.0001	0.24	13	86.00	
		0.00001	0.37		82.00	
	8	0.0001	0.24		84.00	
		0.00001	0.38		84.00	
Violent - flow	4	0.0001	0.52	13	70.97	
		0.00001	0.50		72.58	
	8	0.0001	0.48		70.97	
		0.00001	0.47		75.81	

Table 13 shows the testing accuracy of the C3D model for video-level feature extraction. The C3D achieved 76.40%, 86.00%, and 75.81% testing accuracy on the hockey fight, movie, and violent flow datasets, respectively. Unfortunately, C3D experimental results are less accurate than 2D-CNN feature extraction. Therefore, comparing the accuracy between violent video recognition with MobileNetV1, MobileNetV2 and C3D model, it was found that MobileNetV1 and MobileNetV2 model was still more accurate than the C3D model.

4.5.3 Deep features integrate with 3D convolutional neural network (3D-CNN).

Leverage the robust spatial feature extraction capability of 2D-CNN and 3D-CNN to learn temporal information between adjacent frames. In this

experiment, we employed MobileNetV1 and MobileNetV2 to extract spatial features from individual frames separately. The obtained features were extracted from the last convolution layer before the pooling layer with a size of 7×7×1024. We integrated the features from MobilNetV1 and MobileNetV2 with concatenate for representing individual frames. The integrated feature is larger than the original size of 7×7×2048. The combined features were passed through the proposed 3D convolution for spatial and temporal feature learning. To find the most suitable 3D convolution for violence recognition, we experimented with the proposed five different 3D convolutions, as shown in Table 14.

Table 14 *The five-difference 3D convolution structures.*

Model	**Model1**	**Model2**	**Model3**	**Model4**	**Model5**
	Input Deep Feature (16×7×7×2048)				
	Batch Normalization (16×7×7×2048)				
	Conv3D (1024) K (1×2×2)	Conv3D (512) K (1×2×2)	Conv3D (1024) K (1×2×2)	Conv3D (1024) K (1×2×2)	Conv3D (1024) K (1×2×2)
	Batch normalization	Conv3D (512) K (1×2×2)	Conv3D (512) K (1×2×2)	Conv3D (512) K (1×2×2)	Conv3D (512) K (1×2×2)
	Dropout (0.2)	Batch normalization	Batch normalization	Batch normalization	Batch normalization
	GAP (1024)	Dropout (0.2)	Dropout (0.2)	GAP (512)	Dropout (0.2)
	Dense (2048)	GAP (512)	GAP (512)	Dense (2048)	GAP (512)
	Dense (2)	Dense (2048)	Dense (1024)	Dense (2)	Dense (2048)
		Dense (2)	Dense (2)		Dense (2)
Params	10,499,074	6,303,746	11,019,778	11,547,138	11,547,138
FLOPS (G)	521	431	575	575	575

The hyperparameters are set with different values to achieve optimal performance, including batch size (4 and 8) and learning rate (0.01, 0.001, and 0.0001). The models were trained with 500 epochs. We reported the evaluation metrics regarding testing accuracy, training time, testing time, and model size for

different 3D convolution on the hockey fight, movie, and violent flow datasets, respectively.

Table 15 *Performance of the 3D convolution with integrated deep features on hockey fight dataset.*

Model	Learning rate	Batch size of 4			Batch size of 8			Model size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	
Model 1	0.01	2.06	10	97.20	2.20	10	96.00	80.17
	0.001	2.05	10	96.00	2.11	10	95.20	
	0.0001	2.06	10	95.60	2.03	10	96.40	
Model 2	0.01	1.27	6	95.60	1.26	6	96.40	48.17
	0.001	1.27	6	96.00	1.26	6	95.20	
	0.0001	1.29	6	96.00	1.26	6	96.00	
Model 3	0.01	2.06	11	95.60	2.27	11	95.60	84.15
	0.001	2.07	11	96.40	2.29	11	95.60	
	0.0001	2.09	11	96.00	2.29	11	95.60	
Model 4	0.01	2.03	11	96.00	2.26	11	95.60	88.17
	0.001	2.06	11	95.60	2.29	11	95.60	
	0.0001	2.06	11	96.40	2.12	10	95.60	
Model 5	0.01	2.38	11	96.00	2.29	10	96.00	88.17
	0.001	2.39	11	95.60	2.31	10	96.00	
	0.0001	2.38	11	95.60	2.36	10	95.60	

Table 15 presents the recognition performance of five different 3D convolution structures (model1 – model5) on the hockey fight dataset. The result shows that model1 achieved the highest accuracy with 97.20% when using a batch size of 4 and a learning rate set of 0.01. The model takes training time about 2 hours, and the testing time is about 10 milliseconds.

Table 16 *Performance of the 3D convolution with integrated deep features on movie dataset.*

Model	Learning	Batch size of 4	Batch size of 8	Model size

	rate	Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	(MB)
Model 1	0.01	0.47	9	97.37	0.47	11	97.37	80.17
	0.001	0.48	9	97.37	0.47	11	100.00	
	0.0001	0.48	9	97.37	0.47	11	97.37	
Model 2	0.01	0.26	6	96.00	0.26	6	96.00	48.17
	0.001	0.26	5	96.00	0.25	6	94.00	
	0.0001	0.27	6	96.00	0.27	6	96.00	
Model 3	0.01	0.52	11	96.00	0.52	11	96.00	84.15
	0.001	0.52	11	96.00	0.52	11	96.00	
	0.0001	0.53	11	94.00	0.53	11	96.00	
Model 4	0.01	0.42	11	96.00	0.42	11	96.00	88.17
	0.001	0.42	11	96.00	0.42	11	96.00	
	0.0001	0.43	11	94.00	0.47	11	94.00	
Model 5	0.01	0.45	9	96.00	0.44	11	96.00	88.17
	0.001	0.45	9	96.00	0.45	11	94.00	
	0.0001	0.47	9	92.00	0.48	11	96.00	

Table 16 presents the recognition performance on the movie dataset. The result shows that the 3D convolution with the integrated deep feature can achieve the highest accuracy of 100% on model 1 using a batch size of 4 and a learning rate set of 0.01. The model takes about 0.47 hours to train, and the test time for a sample is about 11 milliseconds.

Table 17 *Performance of the 3D convolution with integrated deep features on violent flow dataset.*

Model	Learning rate	Batch size 4			Batch size 8			Model Size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	
Model 1	0.01	0.47	9	95.65	0.46	11	93.48	80.17
	0.001	0.46	9	95.65	0.46	11	95.65	
	0.0001	0.47	9	93.48	0.46	11	**96.77**	
Model 2	0.01	0.32	6	87.10	0.30	5	91.94	48.17
	0.001	0.32	6	93.55	0.31	6	93.55	
	0.0001	0.34	6	90.32	0.32	6	91.94	

Model	Learning rate	Batch size 4			Batch size 8			Model Size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	
Model 3	0.01	0.54	11	91.94	0.52	11	93.55	84.15
	0.001	0.53	11	91.94	0.53	11	91.94	
	0.0001	0.55	11	91.94	0.53	11	93.55	
Model 4	0.01	0.54	11	93.55	0.53	11	91.94	88.17
	0.001	0.55	11	93.55	0.52	11	93.55	
	0.0001	0.53	11	93.55	0.53	11	93.55	
Model 5	0.01	0.56	9	91.94	0.55	11	93.55	88.17
	0.001	0.56	9	90.32	0.55	11	93.55	
	0.0001	0.56	9	93.55	0.58	11	93.55	

Table 17 presents the performance of the 3D convolution with an integrated deep feature model on the violent flow dataset. The result shows that Model1 achieved the highest accuracy of 96.77% when using a batch size of 8 and learning rate 0.0001. The model takes about 0.46 hours to train, and the test time for a sample is about 11 milliseconds.

Moreover, we extract deep features from the 2D-CNN model, which is trained by merging all datasets. The obtained deep features were integrated and learned spatial and temporal features by 3D convolution also. The difference in batch size and learning rate are compared. Then, the models were evaluated recognition performance with testing split on hockey fight, movie, and violent flow. Table 18 presents the performance of 3D convolution with integrated deep features (merged all dataset) on the hockey fight dataset. The result shows that model1 achieved the highest accuracy of 97.60% when using a batch size of 8 and a learning rate of 0.001. The model takes about 2.24 hours to train, and the test time for a sample is about 11 milliseconds.

Table 18 *Performance of the 3D convolution with integrated deep features(merged all datasets) on the hockey fight dataset.*

Model	Learning rate	Batch size 4			Batch size 8			Model size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	

Model	Learning rate	Batch size 4			Batch size 8			Model size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	
Model 1	0.01	2.35	11	95.60	2.06	11	96.80	80.17
	0.001	2.35	11	96.40	2.24	11	**97.60**	
	0.0001	2.39	11	96.00	2.32	11	96.80	
Model 2	0.01	1.27	11	96.00	1.28	8	96.40	48.17
	0.001	1.27	11	95.60	1.25	8	96.40	
	0.0001	1.31	11	96.40	1.28	8	96.40	
Model 3	0.01	2.05	11	96.00	2.37	11	94.40	84.15
	0.001	2.07	11	89.60	2.36	11	84.40	
	0.0001	2.11	11	88.80	2.37	11	87.20	
Model 4	0.01	2.03	11	96.40	2.07	11	95.20	88.17
	0.001	2.11	11	96.40	2.12	11	96.40	
	0.0001	2.10	10	96.00	2.21	11	96.00	
Model 5	0.01	2.38	11	95.60	2.29	11	95.60	88.17
	0.001	2.39	11	95.60	2.31	11	96.00	
	0.0001	2.38	11	96.00	2.33	11	96.40	

Table 19 presents the performance of 3D convolution with deep features integrated on the movie dataset. The model1 achieved accuracy of 100% with batch size 4 and a learning rate of 0.0001. The model takes about 2.24 hours to train, and the test time for a sample is about 11 milliseconds. Whereas the recognition performance of the proposed 3D convolution model with features extracted from merging all dataset on the violent flow dataset shows that the highest accuracy obtained was 93.55%. The model was trained with a batch size of 4 and learning rate 0.01, as shown in Table 20.

Table 19 *Performance of the 3D convolution with integrated deep features(merged all datasets) on movie dataset.*

Model	Learning rate	Batch size 4			Batch size 8			Model size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	
Model 1	0.01	0.44	11	98.00	0.47	11	98.00	80.17
	0.001	0.47	11	96.00	0.47	11	96.00	
	0.0001	0.47	11	**100.00**	0.47	11	96.00	

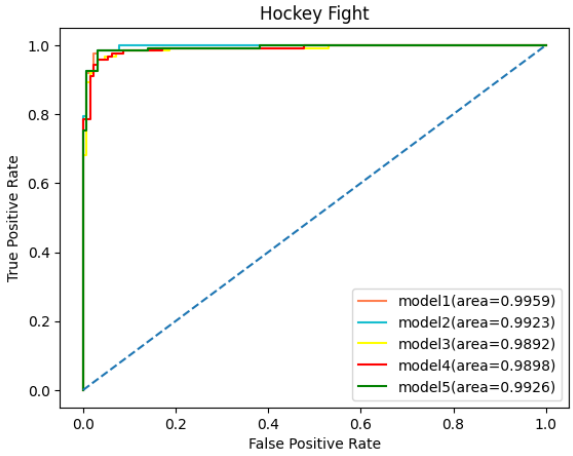
Model	Learning rate	Batch size 4			Batch size 8			Model size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	
Model 2	0.01	0.26	6	98.00	0.26	6	96.00	48.17
	0.001	0.26	6	98.00	0.26	6	96.00	
	0.0001	0.27	6	96.00	0.27	5	96.00	
Model 3	0.01	0.42	11	98.00	0.42	11	96.00	84.15
	0.001	0.43	11	98.00	0.42	11	98.00	
	0.0001	0.44	11	96.00	0.45	11	96.00	
Model 4	0.01	0.42	11	98.00	0.42	11	98.00	88.17
	0.001	0.42	11	98.00	0.42	11	96.00	
	0.0001	0.44	11	96.00	0.46	11	96.00	
Model 5	0.01	0.43	11	98.00	0.43	11	96.00	88.17
	0.001	0.43	11	98.00	0.43	11	96.00	
	0.0001	0.44	11	98.00	0.47	11	98.00	

Table 20 *Performance of the 3D convolution with integrated deep features (merged all datasets) on violent flow dataset.*

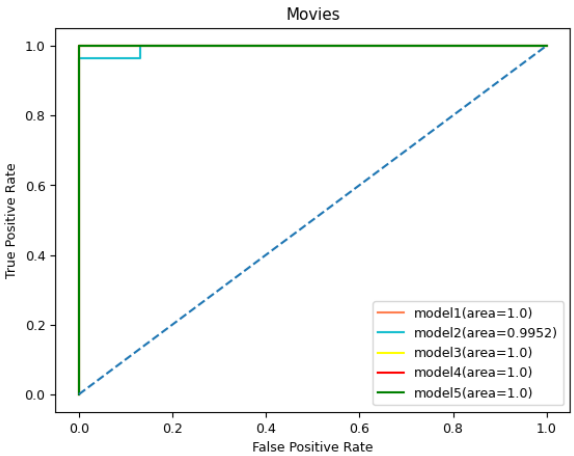
Model	Learning rate	Batch size 4			Batch size 8			Model size (MB)
		Training time (hr.)	Testing time (ms.)	Acc. (%)	Training time (hr.)	Testing time (ms.)	Acc. (%)	
Model 1	0.01	0.51	11	**93.55**	0.72	11	88.71	80.17
	0.001	0.51	11	88.71	0.72	11	90.32	
	0.0001	0.51	11	85.48	0.72	11	87.10	
Model 2	0.01	0.32	5	87.10	0.31	6	88.71	48.17
	0.001	0.33	6	88.71	0.31	5	88.71	
	0.0001	0.34	5	87.10	0.32	6	88.71	
Model 3	0.01	0.52	11	90.32	0.52	11	90.32	84.15
	0.001	0.52	11	85.48	0.52	11	87.10	
	0.0001	0.52	11	87.10	0.53	11	87.10	
Model 4	0.01	0.52	10	93.55	0.52	11	90.32	88.17
	0.001	0.52	11	87.10	0.52	11	90.32	
	0.0001	0.52	11	87.10	0.53	11	90.32	
Model 5	0.01	0.53	11	91.94	0.53	11	88.71	88.17
	0.001	0.53	11	88.71	0.53	11	88.71	
	0.0001	0.53	11	87.10	0.55	11	88.71	

4.5.4 Performance metrics

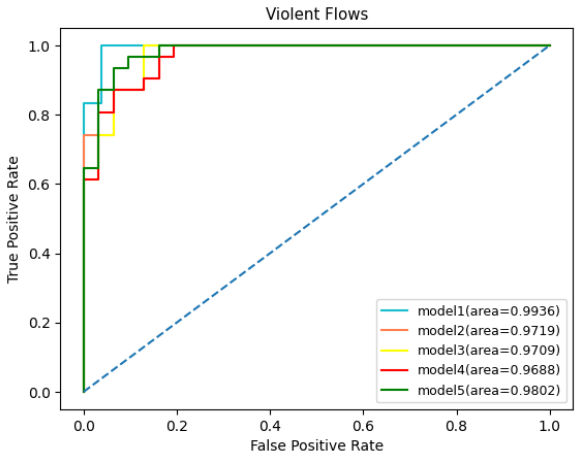
The proposed method was evaluated using well-known classification metrics, including receiver operating characteristics(ROC), area under curve (AUC), precision-recall curve, area under (precision-recall) curve(AUC-PR), training and validation loss, and confusion matrix. The receiver operating characteristics (ROC) curve and area under the curve (AUC) are performed for accuracy of five different 3D-CNN models are illustrated in Figure 36. The AUC value of model 1 is better than other models, 0.99, 1.00, and 0.98 on hockey fight, movie, and violent flow respectively.



(a)



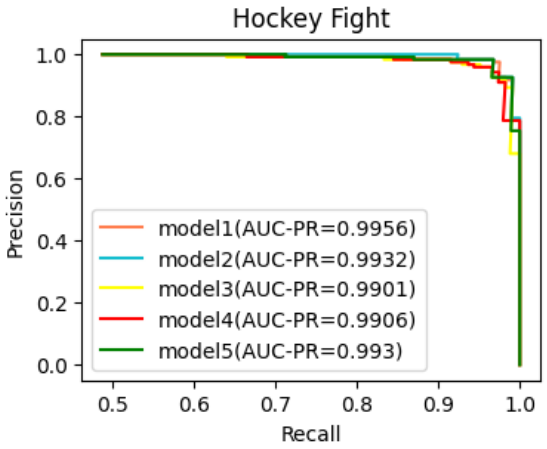
(b)



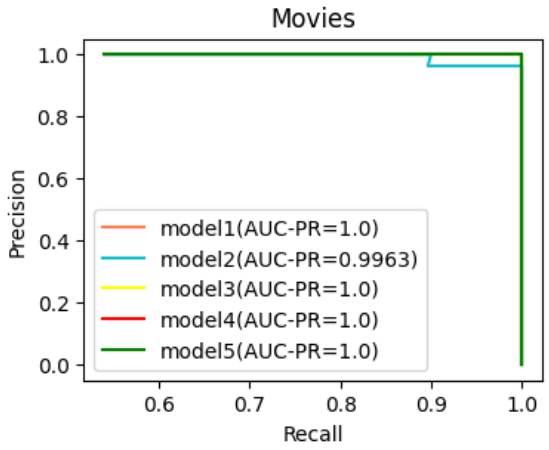
(c)

Figure 36 *Receiver operating characteristic (ROC) curve and area under the curve (AUC) for each 3D-CNN model (a) on the hockey fight dataset, (b) on the movie dataset, and (c) on violence flow dataset.*

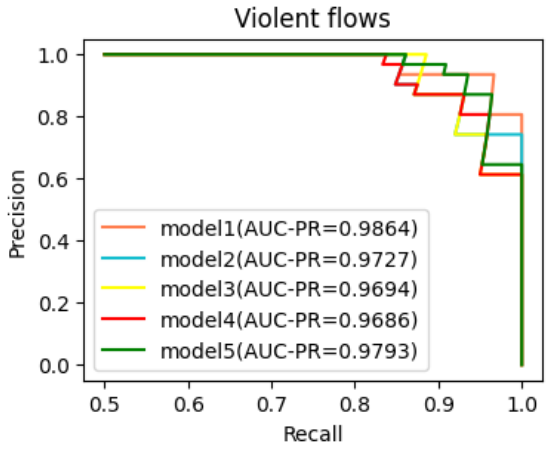
Also, we used the precision-recall curve to evaluate the performance of violent video recognition. Figure 37 demonstrates that the model 1 is a better classifier than other models, with the area under the precision-recall curve (AUC-PR) at 0.9956 on the hockey fight dataset. Almost every model performs well for the movie dataset, with an AUC-PR equal to 1, except model 2, with the lowest AUC-PR equal to 0.9963. Finally, for the violent flow dataset, the graph shows that model 1 has the highest AUC-PR of 0.9864 compared to the other models.



(a)

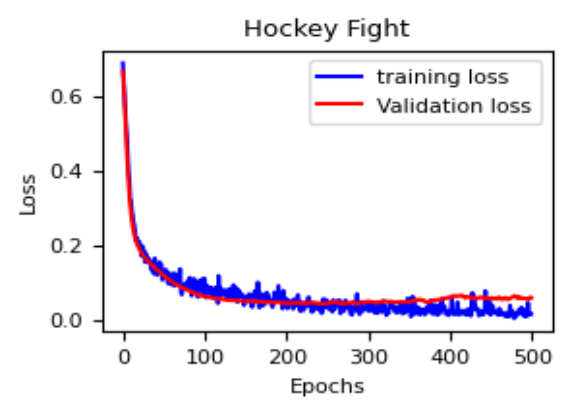


(b)

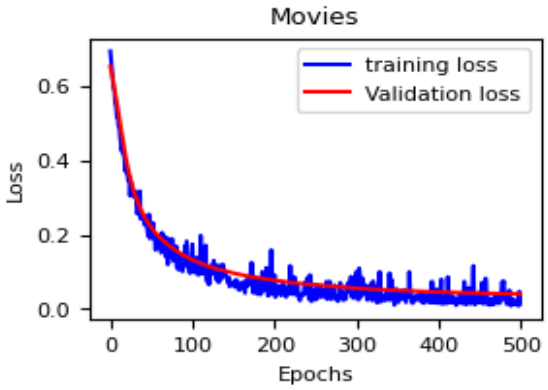


(c)

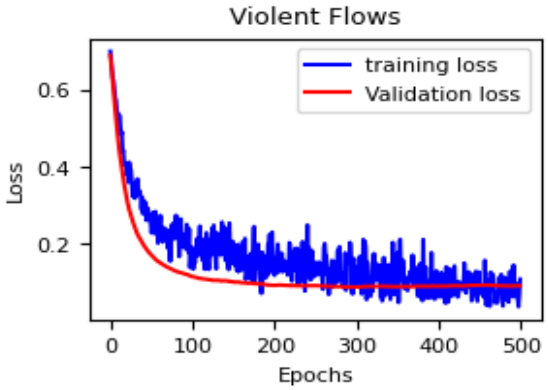
Figure 37 *Precision recall curve and area under the precision recall curve (AUC-PR) for each 3D-CNN model (a) on hockey fight dataset, (b) on movie dataset, and (c) on violent flow dataset.*



(a)



(b)



(c)

Figure 38 *Training and validation loss of our proposed model on the (a) hockey fight dataset, (b) movie dataset, and (c) violent flow dataset.*

The confusion metrics for the test dataset, in Figure 39, represents the correct and incorrect classification of each class on hockey fight, movie, and violent flow datasets. From the confusion metric, the proposed model on the movie dataset performed well in classified violent videos. Besides, the confusion metric on the hockey fight and the violent flow dataset obtained high true positives and true negatives. We have presented some examples of videos that wrongly predicted (false positives and false negatives) the hockey fight and violent flow dataset in Figure 40 and Figure 41 respectively.

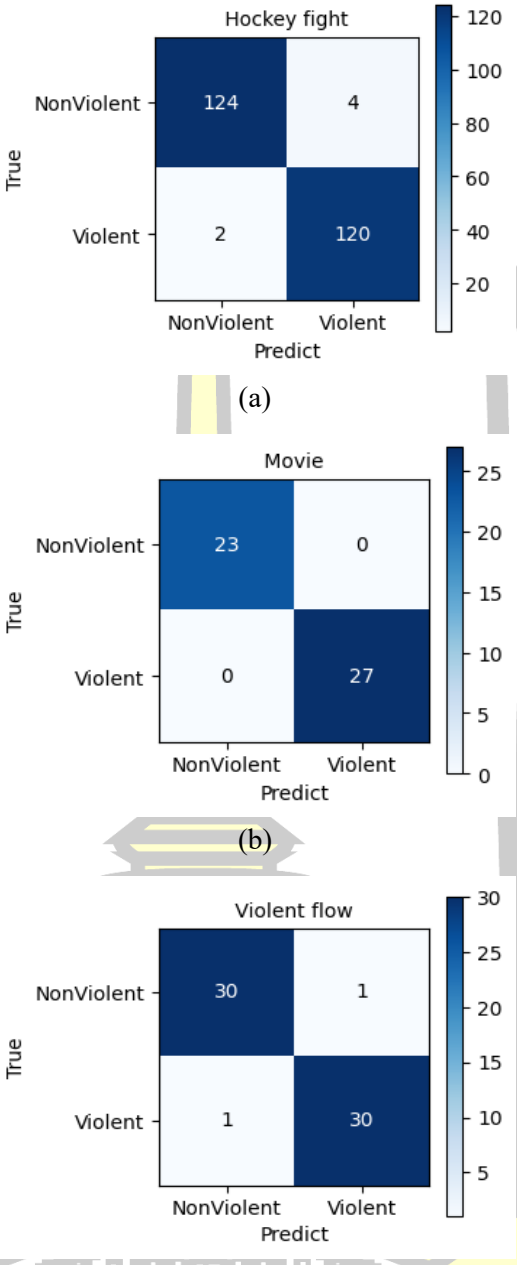
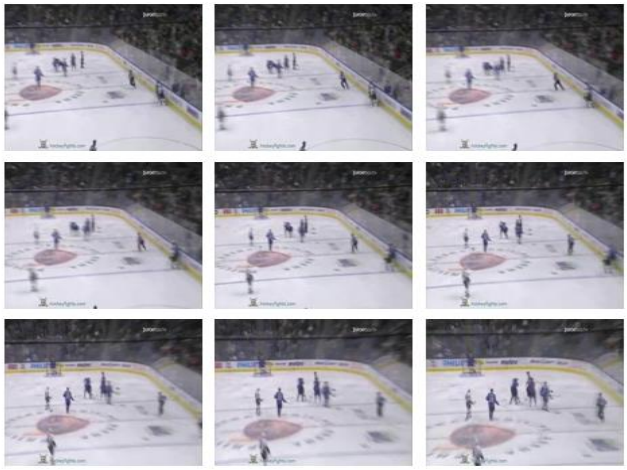
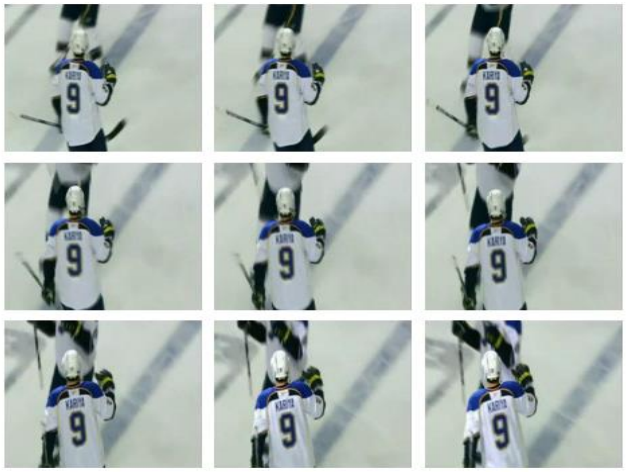


Figure 39 *The confusion matric of test datasets (a) on hockey fight dataset, (b) movie dataset, and (c) violent flow dataset.*



(a)



(b)

Figure 40 *Example of missing video prediction on hockey fight dataset, (a) false negative prediction and (b) false positive prediction.*

(a)

(b)

Figure 41 *Example of missing video prediction on violent flow dataset, (a) false negative prediction and (b) false positive prediction.*

4.6 Discussion

4.6.1 Violent recognition with MobilebileNetV1, MobileNetV2, and C3D

We utilized CNN to evaluate the effectiveness of recognizing violent videos. The result shows that MobileNetV1 and MobileNetV2 can accurately recognize violent videos with high accuracy values on three datasets. The results were obtained by training the model with different batch sizes and learning rates to find an optimal model. For MobileNetV1, Setting the batch size value cannot be confirmed to affect the classification accuracy and testing time.

However, the training time may be reduced using a larger batch size. The learning rate of the model affects learning on hockey fight and movie datasets. In contrast, the different learning rates do not affect the violent recognition performance on the violent flow dataset. For the testing time, MobileNetV1 takes less time than MobileNetNetV2 with all datasets, which are 1, 2, and 2 milliseconds on hockey fight, movie, and violent flow, respectively. Therefore, MobileNetV1 is faster than MobileNetV2 for violent video recognition with the same accuracy.

Then, we experiment with training the model by merging all datasets to compare the training model with a single dataset. The results show that the performance of MobileNetV1 on hockey fight is better for merging all datasets than the training model without merging. For the movie dataset, the recognition results of

merging all datasets were not different from using training separate datasets. For violent flow, the recognition performance of the merging dataset was lower than that of the model without the merging dataset. On the other hand, the recognition performance of MobileNetV2 using training by merging all datasets decreased in all datasets. We discussed the differences in the number of videos of each dataset with those different characteristics. When the datasets were merged for model training and then tested with separate datasets, it caused the recognition performance. The hockey fight dataset comprised 1,000 videos, whereas movie and violent flow had 200 and 246, respectively.

Additionally, we search for a pre-trained model that can learn spatial and temporal features by importing video data from multiple frames simultaneously. We used the C3D model for the training model on three datasets. Unfortunately, C3D experimental results are less accurate than 2D-CNN. Therefore, comparing the accuracy of violent video recognition with MobileNetV1, MobileNetV2, and the C3D model, it was found that the MobileNetV1 and MobileNetV2 models were still more accurate than the C3D model. This may be because the C3D model requires more frame or spatial features to learn, which further increases computational time and model size.

4.6.2 Deep feature integration with 3D-CNN

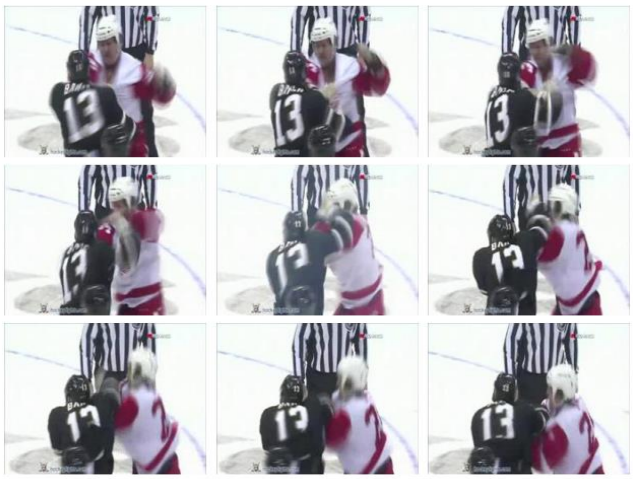
We used the capabilities of MobileNetV1 and MobileNetV2 for significant spatial feature extraction from the video frame at the last convolution layer. The spatial features of individual frames extracted from different CNNs are combined to create a more discriminative feature representation. We proposed 3D convolution to learn spatial and temporal from the concatenated features that produced an excellent performance.

The experimental results clearly show that the performance recognition is better in the feature integration and 3D convolution method than individual MobileNetV1 and MobileNetV2 for all datasets. We conclude that the proposed 3D convolution with the concatenation of features from various 2D-CNNs benefits overcoming the limitation of a single 2D-CNN and producing outstanding performance.

When exploring the impact of object size on the proposed model, the training dataset encompasses diverse object sizes depicted in videos captured from varying perspectives, including high angles, side angles, and close-up angles is shown in Figure 42. The findings of the experimental results reveal that the model exhibits the ability to recognize violent videos captured from high angles, except for videos where the camera angle is so far away that it is impossible to distinguish it visually. Therefore, the size of the objects in the video does not affect the recognition performance of violent videos.

(a)

(b)



(c)

Figure 42 *Example of violent video with different camera angles: (a) very long shot, (b) medium-close-up shot, and (c) close-up shot.*

When considering the overall experimental results, it is indicated that the proposed approach achieved high accuracy on all datasets. Thus, we assume that the proposed approach can be applied to unseen datasets categorized as violent or non-violent video. At the same time, additional training is required when applied to differently classified datasets.

4.6 Comparison

The comparison of the proposed method with state-of-the-art methods based on the accuracy of violent recognition on three standard datasets is presented in this section. Table 21, Table 22 and Table 23 show the experimental results of the hockey fight, movie and violent flow datasets that our proposed outperformed the state-of-the-art method. The proposed method achieved 97.60%, 100% and 96.77% accuracy, respectively. The research in Table 21 has divided the dataset into training 80% and testing 20%. Therefore, an extended experiment was shown that split the dataset for training and testing to 80% and 20%, respectively. The accuracy of the proposed slightly increased is 98% from 96.77%, more than research (Jahlan & Elrefaei, 2022).

Table 21 *A comparison of the proposed method with the state-of-the-art methods on hockey fight dataset.*

Ref.	Method	No. of Frame	Classifier	Data Splitting (Train/Test) (%)	Testing Accuracy (%)
(Khan et al., 2019)	MobileNet	N/A	softmax	75/25	87.00
(Soliman et al., 2019)	VGG16+LSTM	20	LSTM	80/20	88.20
(Carneiro et al., 2019)	Multi-Stream	40	SVM	90/10	89.10
(Ullah et al., 2019)	3D-CNN	16	softmax	75/25	96.00
(Hanson et al., 2019)	VGG13+BiConvLSTM	20	FC	80/20	96.96
(Jahlan & Elrefaei, 2022)	AlexNet,SqueezNet and ConvLSTM	20	softmax	80/20	97.00
Proposed		16	softmax	75/25	97.60

Table 22 *A comparison of the proposed method with the state-of-the-art methods on movie dataset.*

Ref.	Method	No. of Frame	Classifier	Data Splitting (Train/Test) (%)	Testing Accuracy (%)
(Carneiro et al., 2019)	Multi-Stream	40	SVM	90/10	100.00
(Hanson et al., 2019)	VGG13+BiConvLSTM	20	FC	80/20	100.00
(Atallah Almazroey & Kammoun Jarraya, 2021)	Keyframe+AlexNet	50	SVM	80/20	100.00
(Jahlan & Elrefaei, 2022)	AlexNet,SqueezNet and ConvLSTM	20	softmax	80/20	100.00
Proposed		16	softmax	75/25	100.00

Table 23 *A comparison of the proposed method with the state-of-the-art methods on violent flow dataset.*

Ref.	Method	No. of Frame	Classifier	Data Splitting (Train/Test) (%)	Testing Accuracy (%)
(Soliman et al., 2019)	VGG16+LSTM	20	LSTM	80/20	90.01
(Hanson et al., 2019)	VGG13+BiConvLSTM	20	FC	80/20	90.60
(Jahlan & Elrefaei, 2022)	AlexNet,SqueezNet and ConvLSTM	20	softmax	80/20	96.00
Proposed		16	softmax	80/20	98.00

Ref.	Method	No. of Frame	Classifier	Data Splitting (Train/Test) (%)	Testing Accuracy (%)
				75/25	96.77

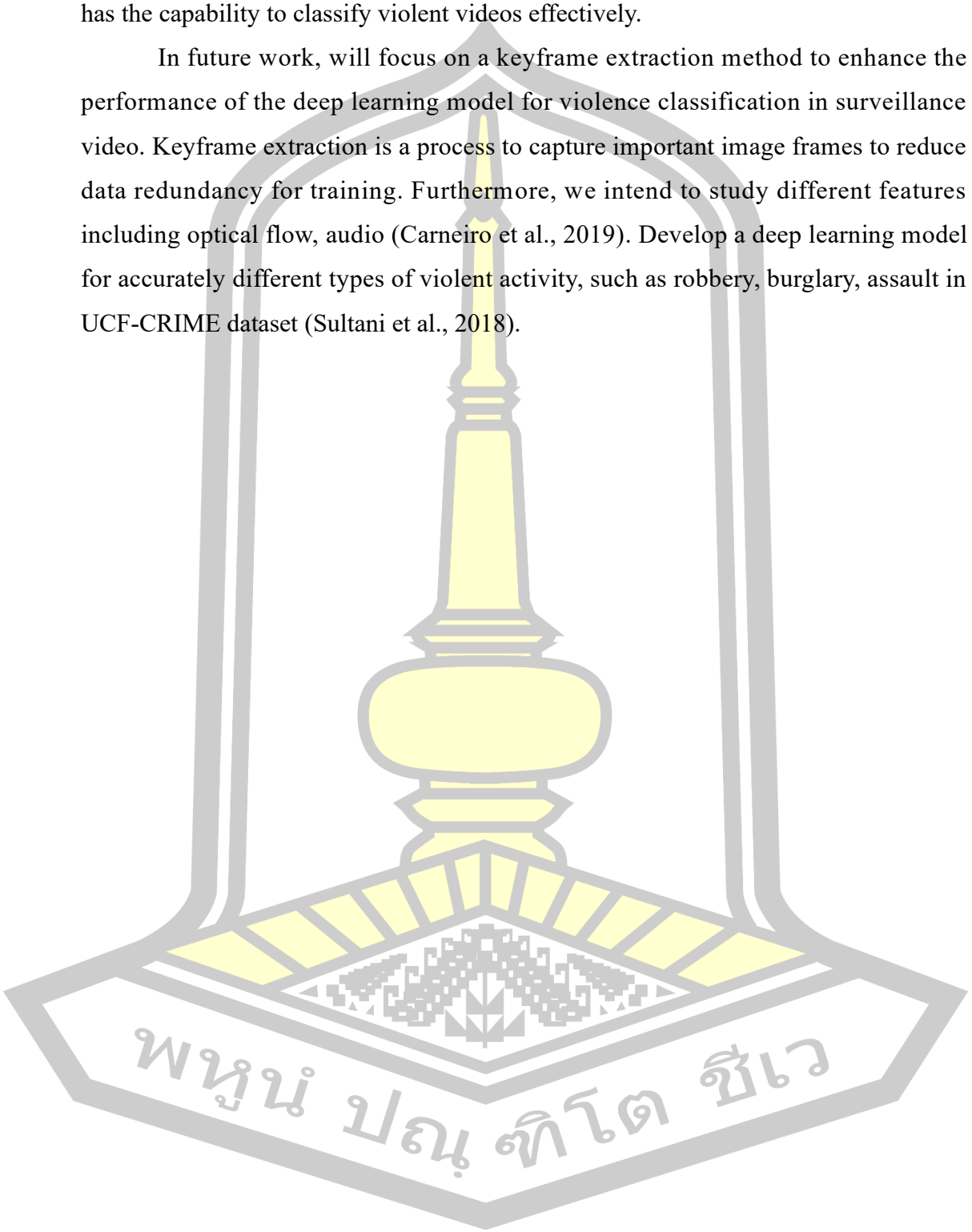
4.7 Conclusions

In this research, proposed a deep feature integration and three-dimensional convolution for violence recognition. The 16 non-overlapping were captured for representative frames of each video as input to the CNN model for feature extraction. The proposed method was tested on three benchmark datasets, including hockey fight, movie, and violent flow. The recognition accuracy, confusion metric, ROC, and precision recall curve are presented to evaluate classifier models. In the first part, we used the pre-trained CNN model on the ImageNet dataset, including MobileNetV1 and MobileNetv2, to classify violent video. The results show that the two CNN models can be no different in classifying video violence with an accuracy of 95.99%, 98.00%, and 91.94% for the hockey fight, movie, and violent flow datasets, respectively. Then, we merged all the datasets to retrain the above CNN model to classify video violence. Unfortunately, the results of this experiment were not better than using separate datasets. In addition, we also used the C3D to recognize violent video, which is the pre-trained model using 3D convolutional networks. The experimental results show that the model achieved a lower accuracy score than MobileNetV1 and MobileNetV2. The model size is large, and it is also the most time-consuming part of network training.

We leverage the advantages of the 2D-CNN model and the 3D-CNN to extract robust spatial and spatiotemporal features from the video. The video frames extracted the robust features from the last convolution layer in MobilNetV1 and MobileNetV2 separately. Then, we integrated the obtained features with concatenate to create explicit feature representation. The integrated features were used as input for the proposed 3D convolution instead of the video frame and classified by softmax to violent video. We experiment with the different 3D convolution structures. In our work, we proved that the proposed method performs better than a single use of MobileNetV1, MobileNetV2, and C3D with 97.60%, 100% and 96.77% accuracy on hockey fight, movie, and violent flow datasets, respectively. The experimental results

demonstrated that the proposed method performs better than the existing methods and has the capability to classify violent videos effectively.

In future work, will focus on a keyframe extraction method to enhance the performance of the deep learning model for violence classification in surveillance video. Keyframe extraction is a process to capture important image frames to reduce data redundancy for training. Furthermore, we intend to study different features including optical flow, audio (Carneiro et al., 2019). Develop a deep learning model for accurately different types of violent activity, such as robbery, burglary, assault in UCF-CRIME dataset (Sultani et al., 2018).



Chapter 5
Discussion

This thesis aims to propose deep learning approaches to improving violent video recognition. In the findings of this research, we contribute two main types of research. I first proposed an approach that integrates lightweight CNNs and sequence learning, employing MobileNetV1 and MobileNetV2 for robust deep feature extraction. The deep features were combined through concatenation and processed using bidirectional long short-term memory (BiLSTM) to discern violent or non-violent videos. Additionally, a focused effort was made to enhance efficiency by reducing the number of frames for training and feature extraction on the hockey fight dataset. Second, a novel method was introduced to enhance the accuracy of violent video recognition by integrating deep features with a 3D Convolutional Neural Network (3D-CNN), leveraging the advantages of MobileNets for spatial feature extraction at the frame level. The proposed 3D-CNN was designed for spatiotemporal feature learning to preserve crucial temporal information between frames. The effectiveness of the proposed method is evaluated across three benchmark datasets: hockey fight, movie, and violent flow.

I will now briefly describe and discuss the challenges of violent video recognition using a deep learning approach.

Chapter 3. Achieving precision in identifying violence within videos requires high accuracy and a significant amount of computational time due to processing multiple video frames. I concurrently optimized for accuracy and recognition time conditions to solve this challenge. I propose the fusion lightweight CNN and sequence learning approach for violent video recognition. First, I propose five different frame selections using MobileNet and LSTM to classify violent videos. The experimental results found that using 16 non-adjacent frames resulted in the highest accuracy. Second, I propose four different CNN architectures for deep feature extraction and sent the deep feature to LSTM to classify violent video. The experimental results show that MobileNet+LSTM achieved the highest accuracy on the hockey fight dataset. Moreover, MobileNet+LSTM had the smallest model size

and the least computation time. Third, I used MobileNetV1 and MobileNetV2 to extract the robust deep feature from the video frame. Then, the deep features are fused with concatenating and adding operations to generate the video representation features before transfer into RNN architecture.

I proposed three different RNN architectures: LSTM, BiLSTM, and GRU. I trained the combination of fusion MobileNet and RNN architecture with 1,000 epochs. The experimental results indicated that the fusion MobileNet and RNN model outperform the single CNN models by approximately 2% on the hockey fight dataset. Consequently, the concatenating operation achieved better accuracy when combining MobileNet with BiLSTM because the deep feature size of the concatenating operation was larger one time than the adding operation.

Although the proposed architecture requires a longer time due to the training from both MobileNets, the testing time remains the same. Therefore, it can be concluded that the fusion MobileNet and RNN architecture can be applied to classify violence because it is recognized quickly with high accuracy, and extending the complex architecture does not affect the recognition time.

Chapter 4. I focus on enhancing the efficiency of violent video recognition by integrating deep features with the 3D-CNN approach. First, I propose MobileNetV1 and MobileNetV2 for leveraging spatial feature extraction from video frames. The spatial features were integrated with concatenate operation for a more robust video feature representation, encouraged by the fusion MobileNets-BiLSTM described in Chapter 3. Second, the integrated spatial features were transferred into the proposed 3D-CNN to capture spatial and temporal features within the video data. I proposed five different 3D-CNN architectures with different structures with the same input feature. I trained the proposed architecture with 500 epochs, different batch sizes, and learning rate settings. I evaluate the proposed model with three benchmark datasets: hockey fight, movie, and violent flow.

The proposed 3D-CNN architecture achieved the highest accuracy values, comprising batch normalization, 3D convolution, dropout, and a global average pooling, fully connected layers followed by a softmax function to classify violent or non-violent videos. The experimental result shows that the integrated deep feature

with the 3D-CNN approach outperformed the single MobileNet by approximately 2% on three datasets. Also, the proposed method can improve the performance of violent video recognition by approximately 3% compared to the method MobileNet-BiLSTM.

5.1 Answers to The Research Questions

According to the research questions (RQ) in Chapter 1, I explain the improvement of violent video recognition using deep learning with two solutions. In this section, I briefly answer each research question.

Objective 1. I aim to research deep learning studies that improve violent video recognition performance by combining CNN and RNN with deep feature fusion techniques.

Research Question 1. Generally, violent video understanding applies Recurrent Neural Networks (RNN) such as LSTM, BiLSTM, or GRU to learn the feature from sequential frames within the video data. RNN can distinguish patterns and movements, accurately classifying actions, or activities in video. However, some research used Convolutional Neural Networks (CNN) to extract deep features from the individual frame, which received high accuracy for violent recognition (Karisma et al., 2021) and (Irfanullah et al.,2022). Therefore, if I utilize the CNN to extract the deep features from video frames and then transfer the received deep features to RNN to learn information within the video, will this improve the performance of understanding violent videos?

To find the answer to RQ1, I focused on a state-of-the-art CNN and RNN model. The frame selections and number of input frames are also considered to reduce redundant information. The deep feature fusion technique was applied to combine the deep features from different architectures to understand violent video and improve the performance of violent video recognition. Will these methods encourage enhancing the performance of violent video recognition?

To answer RQ1, I first focused on the video frame selection to reduce the number of frames and redundant information for training and feature extraction. Using 16 non-adjacent frames resulted in the highest accuracy. Second, lightweight MobileNets were used for deep feature extraction due to the lower computation time and higher accuracy than ResNet50V2 and NASNetMobile. The obtained deep

features were fused with concatenating operations to leverage information from both MobileNets before being transferred to RNN architecture, including LSTM, BiLSTM, and GRU. Finally, the fused deep feature was transferred to BiLSTM to learn features from violent videos and classify them into violent or non-violent videos. The result showed that the accuracy increased by approximately 2% on the hockey fight dataset when combining the deep feature with concatenating and sending it to BiLSTM. Considering testing time indicates that network expansion does not affect the performance of violent video recognition. Consequently, I can combine lightweight MobileNet and Bi-LSTM with the deep feature fusion technique to improve the performance of violent recognition while maintaining recognition time.

Objective 2. I aim to research deep learning approaches that improve violent video recognition by deep feature integration with three-dimensional convolution neural network (3D-CNN)

Research Question 2. the 2D-CNN outperforms in extracting spatial features within individual frames, making it well-suited for tasks where static visual patterns hold pivotal significance, such as image classification and object detection. Conversely, 3D-CNN surpasses 2D-CNN in tasks requiring the incorporation of necessary temporal dimensions, as it can directly understand spatiotemporal features from video sequences. This renders 3D-CNN notably advantageous for applications like action recognition, wherein comprehending temporal alterations and motion is imperative. Although 2D-CNN demonstrates computational efficiency and is commonly employed for image-based tasks, 3D-CNN extends its functionalities to video analysis by seamlessly incorporating temporal information into the learning process. Therefore, If 2D-CNN is used to extract spatial features from frames and integrate the obtained features. Then, the features are transferred to 3D-CNN for spatiotemporal learning and classified into violent or nonviolent videos. Can the proposed approach improve the performance of violent video recognition?

To find the answer to RQ2, I will use 2D-CNN to extract spatial and integrated features with concatenating operations. Then, these features are transferred to 3D-CNN to learn spatiotemporal features and classify violent or non-violent videos.

To answer RQ2, I proposed a deep features integration and 3D convolution for violence recognition. First, I extracted the spatial features with CNN from the video frame. These features were integrated with concatenating operations to enhance more information representations. I focus on the 3D convolution architecture, which capability captured spatiotemporal information by considering multiple frames in video. The effectiveness of the proposed method was evaluated on three benchmark datasets hockey fight, movie, and violent flow datasets.

The proposed approach demonstrated an approximate 2% increase in the recognition performance of violent videos across all datasets when compared to a single CNN. Moreover, compared to a single 3D-CNN, the proposed approach exhibited significant improvements, with performance improvements of approximately 27%, 16%, and 29% on the hockey fight, movie, and violent flow datasets, respectively. These results emphasize the capability of the 3D-CNN to enhance the recognition performance of violent videos. Moreover,

5.2 Future work

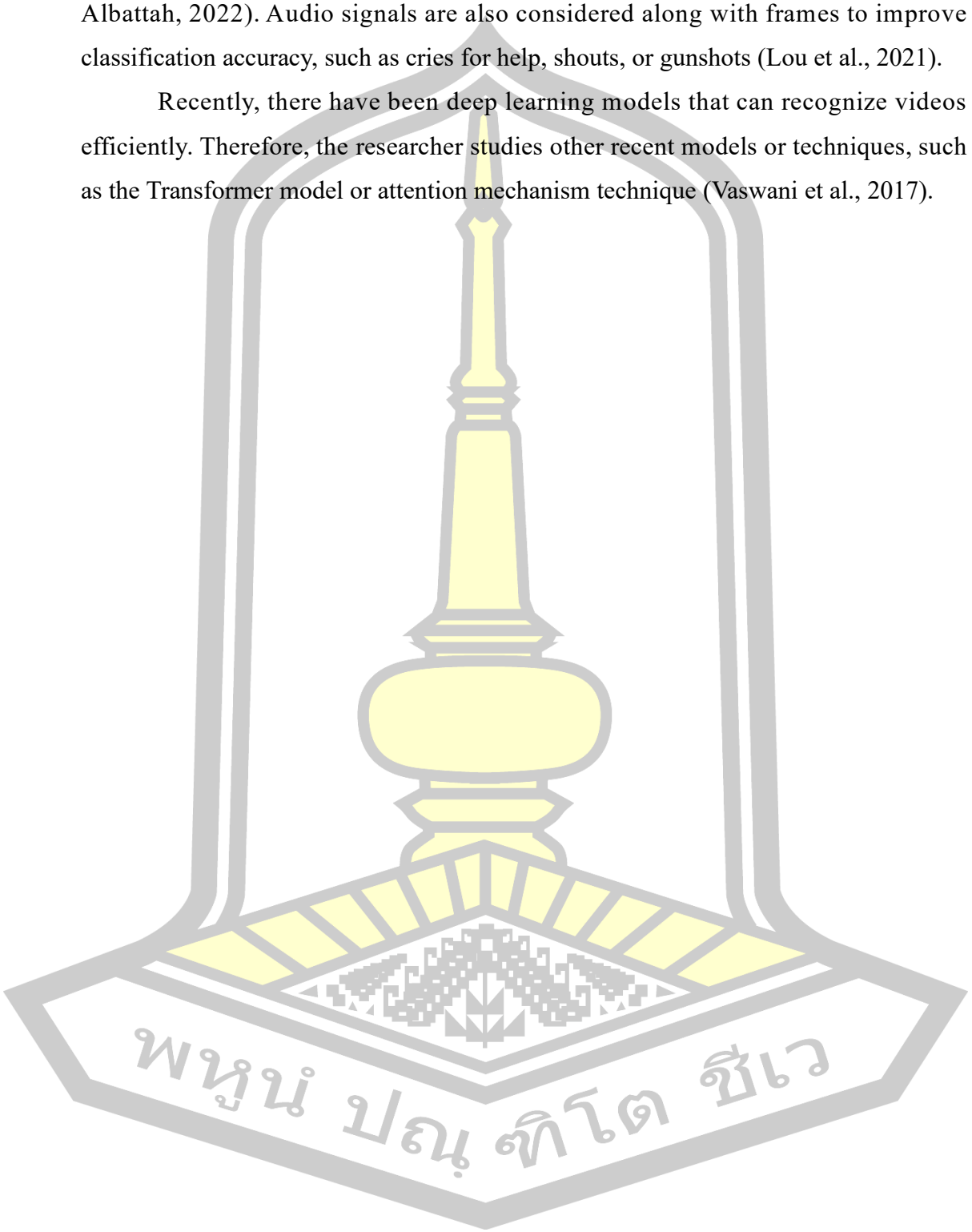
In this dissertation, I proposed novel deep feature extraction techniques to improve the performance of video understanding based on violent video. Several future works present in the following could be used as a direction for recognizing video violence tasks. I described frame selection, challenges, and applications of up-to-date deep learning models.

The frame selection is a critical procedure for selecting essential input data for training in the deep learning model. Researchers considered reducing the number of sample frames for selecting keyframes to reduce the redundant information and computation cost while making the developed model efficient such as adaptive frame selection (Tao & Duan, 2023).

Moreover, researchers can further improve responses to many challenges of violence video recognition, such as recognition of different types of violence, human occlusion, or audio signals in conjunction with image frames. For different types of violence, it is challenging to differentiate violent behavior including robbery, burglary, assault in the UCF-CRIME dataset (Sultani et al., 2018) . For human occlusion, an object may be obscuring a violent incident, resulting in difficulty distinguishing

whether it is violent or not due to only part of the scene being visible (Aldayri & Albattah, 2022). Audio signals are also considered along with frames to improve classification accuracy, such as cries for help, shouts, or gunshots (Lou et al., 2021).

Recently, there have been deep learning models that can recognize videos efficiently. Therefore, the researcher studies other recent models or techniques, such as the Transformer model or attention mechanism technique (Vaswani et al., 2017).



REFERENCES

Aldayri, A., & Albattah, W. (2022). Taxonomy of anomaly detection techniques incrowd scenes. *Sensors*, *22*(16), 6080. https://doi.org/https://doi.org/10.3390/s22166080

Alharthi, R., Alhothali, A., Alzahrani, B., & Aldhaheri, S. (2023). Massive crowd abnormal behaviors recognition using C3D. IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53-53. https://doi.org/10.1186/s40537-021-00444-8

Atallah Almazroey, A., & Kammoun Jarraya, S. (2021). *Fight detection in crowd scenes based on deep spatiotemporal features.* International Conference on Artificial Intelligence, Robotics and Control, New York, NY, USA. https://doi.org/10.1145/3448326.3448329

Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., & Amirat, Y. (2015). Physical human activity recognition using wearable sensors. *Sensors*, *15*(12), 31314-31338. https://doi.org/10.3390/s151229858

Bermejo, E., Deniz, O., Bueno, G., & Sukthankar, R. (2011). Violence detection in video using computer vision techniques. International Conference on Computer Analysis of Images and Patterns, Berlin, Heidelberg.

Boureau, Y. L., Ponce, J., & LeCun, Y. (2010, June). A theoretical analysis of feature pooling in visual recognition. International Conference on Machine Learning (ICML),, Haifa, Israel.

Carneiro, S. A., da Silva, G. P., Guimarães, S. J. F., & Pedrini, H. (2019). Fight detection in video sequences based on multi-stream convolutional neural networks. Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil.

Celard, P., Iglesias, E. L., Sorribes-Fdez, J. M., Romero, R., Vieira, A. S., & Borrajo, L. (2023). A survey on deep learning applied to medical images: from simple artificial neural networks to generative models. *Neural Computing and Applications*, *35*(3), 2291-2323. https://doi.org/10.1007/s00521-022-07953-4

Chen, H., Hu, C., Lee, F., Lin, C., Yao, W., Chen, L., & Chen, Q. (2021). A supervised video hashing method based on a deep 3D convolutional neural network for large-scale video retrieval. *Sensors*, *21*(9). https://doi.org/10.3390/s21093094

Chen, J., Wang, J., Yuan, Q., & Yang, Z. (2023). CNN-LSTM model for recognizing video-recorded actions performed in a traditional chinese exercise. *IEEE Journal of Translational Engineering in Health and Medicine*, *11*, 351-359. https://doi.org/10.1109/JTEHM.2023.3282245

Cheng, M., Cai, K., & Li, M. (2021, 2021/01//). RWF-2000: an open large scale video database for violence detection. International Conference on Pattern Recognition (ICPR), Los Alamitos, CA, USA.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase pepresentations using RNN encoder-decoder for statistical machine translation. *arXiv*, *abs/1406.1078*, arXiv:1406.1078. https://doi.org/10.48550/arXiv.1406.1078

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA.

Dallel, M., Havard, V., Baudry, D., & Savatier, X. (2020). InHARD - industrial human action recognition dataset in the context of industrial collaborative robotics. IEEE International Conference on Human-Machine Systems (ICHMS), Rome, Italy.

Das, S., Sarker, A., & Mahmud, T. (2019). Violence detection from videos using HOG features. International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh.

Ditsanthia, E., Pipanmaekaporn, L., & Kamonsantiroj, S. (2018). Video representation learning for CCTV-based violence detection. Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), Bangkok, Thailand.

Gao, R., Oh, T.-H., Grauman, K., & Torresani, L. (2019). Listen to look: action recognition by previewing audio. *CoRR*, *abs/1912.0*. http://arxiv.org/abs/1912.04487

Gao, Y., Liu, H., Sun, X.-h., Wang, C., & Liu, Y. (2016). Violence detection using oriented violent flows. *Image and Vision Computing (IMAVIS)*, *48*(1), 37-41. https://doi.org/10.1016/j.imavis.2016.01.006

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *International Neural Network Society*, *18*(5-6), 602-610. https://doi.org/10.1016/j.neunet.2005.06.042

Hanson, A., Pnvr, K., Krishnagopal, S., & Davis, L. (2019). Bidirectional convolutional LSTM for the detection of violence in videos. the European Conference on Computer Vision (ECCV) workshops, Cham.

Hassner, T., Itcher, Y., & Kliper-Gross, O. (2012). Violent flows: real-time detection of violent crowd behavior. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Rhode Island, USA.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. European Conference on Computer Vision (ECCV), Cham.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Howard, A. G., Zhu, M., Chen, B. D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications. *CoRR*, *abs/1704.04861*, arXiv:1704.04861. https://arxiv.org/abs/1704.04861

Hu, Z.-p., Zhang, L., Li, S.-f., & Sun, D.-g. (2020). Parallel spatial-temporal convolutional neural networks for anomaly detection and location in crowded scenes. *Journal of Visual Communication and Image Representation*, *67*, 102765-102765. https://doi.org/10.1016/j.jvcir.2020.102765

Humeau-Heurtier, A. (2019). Texture feature extraction methods: a survey. *IEEE Access*, *7*, 8975-9000. https://doi.org/10.1109/ACCESS.2018.2890743

Irfanullah, Hussain, T., Iqbal, A., Yang, B., & Hussain, A. (2022). Real time violence detection in surveillance videos using Convolutional Neural Networks. *Multimedia Tools and Applications*, *81*(26), 38151-38173. https://doi.org/10.1007/s11042-022-13169-4

Jahlan, H. M. B., & Elrefaei, L. A. (2022). Detecting violence in video based on deep features fusion technique. *arXiv*, *abs/2204.07443*, arXiv:2204.07443. https://arxiv.org/abs/2204.07443

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221-231. https://doi.org/10.1109/TPAMI.2012.59

Jiaxin, Y., Fang, W., & Jieru, Y. (2021). A review of action recognition based on Convolutional Neural Network. *Journal of Physics: Conference Series (JPCS)*, *1827*(1), 12138-12138. https://doi.org/10.1088/1742-6596/1827/1/012138

Karisma, Imah, E. M., & Wintarti, A. (2021). Violence classification using support vector machine and deep transfer learning feature extraction. International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia.

Keçeli, A. S., & Kaya, A. (2017). Violent activity detection with transfer learning method. *Electronics Letters*, *53*(15), 1047-1048. https://doi.org/10.1049/el.2017.0970

Khan, S. U., Haq, I. U., Rho, S., Baik, S. W., & Lee, M. Y. (2019). Cover the violence: a novel deep-learning-based approach towards violence-detection in movies. *Applied Sciences*, *9*(22), 4963-4963. https://doi.org/10.3390/APP9224963

Kreuter, D., Takahashi, H., Omae, Y., Akiduki, T., & Zhang, Z. (2020). Classification of human gait acceleration data using convolutional neural networks. *International Journal of Innovative Computing, Information and Control (IJICIC)*, *16*(2), 609-619. https://doi.org/10.1109/ JSEN.2019.2928777

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

Lee, J., Abu-El-Haija, S., Varadarajan, B., & Natsev, A. (2018). Collaborative deep metric learning for video understanding. International Conference on Knowledge Discovery and Data Mining, New York, NY, USA.

Lejmi, W., Ben Khalifa, A., & Mahjoub, M. (2020). A novel spatio-temporal violence classification framework based on material derivative and LSTM neural network. *Traitement du Signal*, *37*, 687-701. https://doi.org/10.18280/ts.370501

Li, C., Zhu, L., Zhu, D., Chen, J., Pan, Z., Li, X., & Wang, B. (2018). End-to-end multiplayer violence detection based on deep 3D CNN.*ICNCC 2018* International Conference on Network, Communication and Computing (ICNCC), New York, NY, USA.

Li, J., Jiang, X., Sun, T., & Xu, K. (2019). Efficient violence detection using 3d convolutional neural networks. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan.

Lin, J., Gan, C., & Han, S. (2019). TSM: Temporal Shift Module for efficient video understanding. Proceedings of the IEEE/CVF international conference on computer vision,

Liu, J., Sun, W., Zhao, X., Zhao, J., & Jiang, Z. (2022). Deep feature fusion classification network (DFFCNet): Towards accurate diagnosis of COVID-19 using chest X-rays images. *Biomedical Signal Processing and Control*, *76*, 103677-103677. https://doi.org/10.1016/j.bspc.2022.103677

Liu, X., Liu, L., Simske, S. J., & Liu, J. (2016). Human daily activity recognition for healthcare using wearable and visual sensing data. International Conference on

Healthcare Informatics (ICHI), Chicago, IL, USA.
Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., & Hu, H. (2022). *Video swin transformer* IEEE/CVF Conference on Computer Vision and Pattern Recognition,
Lou, J., Zuo, D., Zhang, Z., & Liu, H. (2021). Violence recognition based on auditory-visual fusion of autoencoder mapping. *Electronics*, *10*(21). https://doi.org/10.3390/electronics10212654
Lu, S., Ding, Y., Liu, M., Yin, Z., Yin, L., & Zheng, W. (2023). Multiscale feature extraction and fusion of image and text in VQA. *International Journal of Computational Intelligence Systems*, *16*(1), 54-54. https://doi.org/10.1007/s44196-023-00233-6
Ma, W., Zhu, X., Xiang, W., Li, W., & Wang, Z. (2023). Human Behavior Recognition and Detection Technology Based on Video Stream. International Conference on Intelligent Media, Big Data and Knowledge Mining (IMBDKM),
Maqsood, R., Bajwa, U. I., Saleem, G., Raza, R. H., & Anwar, M. W. (2021). Anomaly recognition from surveillance videos using 3D convolution neural network. *Multimedia Tools and Applications*, *80*(12), 18693-18716. https://doi.org/10.1007/s11042-021-10570-3
Mercioni, M. A., & Holban, S. (2023). A brief review of the most recent activation functions for neural networks. International Conference on Engineering of Modern Electric Systems (EMES), Oradea, Romania.
Morshed, M. G., Sultana, T., Alam, A., & Lee, Y.-K. (2023). Human action recognition: a taxonomy-based survey, updates, and opportunities. *Sensors*, *23*(4). https://doi.org/10.3390/s23042182
Mumtaz, N., Ejaz, N., Aladhadh, S., Habib, S., & Lee, M. Y. (2022). Deep multi-scale features fusion for effective violence detection and control charts visualization. *Sensors*, *22*(23). https://doi.org/10.3390/s22239383
Naik, A., & Gopalakrishna, M. T. (2021). Deep-violence: individual person violent activity detection in video. *Multimedia Tools and Applications*, *80*, 1-16. https://doi.org/10.1007/s11042-021-10682-w
Nirthika, R., Manivannan, S., Ramanan, A., & Wang, R. (2022). Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. *Neural Computing and Applications*, *34*(7), 5321-5347. https://doi.org/10.1007/s00521-022-06953-8
Patil, C. M., Jagadeesh, B., & Meghana, M. N. (2017). An approach of understanding human activity recognition and detection for video surveillance using HOG descriptor and SVM classifier. International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India.
Peixoto, B., Lavi, B., Bestagini, P., Dias, Z., & Rocha, A. (2020). Multimodal violence detection in videos. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain.
Pratama, R. A., Yudistira, N., & Bachtiar, F. A. (2023). Violence recognition on videos using two-stream 3D CNN with custom spatiotemporal crop. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023-15599-0
Pu, S., Chu, L., Hou, Z., Hu, J., Huang, Y., & Zhang, Y. (2022). Spatial-Temporal Feature Extraction and Evaluation Network for Citywide Traffic Condition

Prediction. In.
Rajavel, R., Ravichandran, S. K., Harimoorthy, K., Nagappan, P., & Gobichettipalayam, K. R. (2022). IoT-based smart healthcare video surveillance system using edge computing. *Journal of Ambient Intelligence and Humanized Computing (JAIHC)*, *13*(6), 3195-3207. https://doi.org/10.1007/s12652-021-03157-1
Ramesh, M., & Mahesh, K. (2022). Sports video classification framework using enhanced threshold based keyframe selection algorithm and customized CNN on UCF101 and Sports1-M dataset. *Computational Intelligence and Neuroscience*, *2022*, 3218431-3218431. https://doi.org/10.1155/2022/3218431
Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: inverted residuals and linear bottlenecks. IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA.
Sarker, I. H. (2021). Deep Learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 420-420. https://doi.org/10.1007/s42979-021-00815-1
Sharifani, K., & Amini, M. (2023). Machine learning and deep learning: a review of methods and applications. *World Information Technology and Engineering Journal*, *10*(07), 3897-3904.
Sharma, D. R., & Sungheetha, D. A. (2021). An efficient dimension reduction based fusion of CNN and SVM model for detection of abnormal incident in video surveillance. *Journal of Soft Computing Paradigm*, *3*(2), 55-69.
Singh, P., Chaudhury, S., & Panigrahi, B. K. (2021). Hybrid MPSO-CNN: multi-level particle swarm optimized hyperparameters of convolutional neural network. *Swarm and Evolutionary Computation*, *63*, 100863-100863. https://doi.org/10.1016/j.swevo.2021.100863
Siregar, A. F., & Mauritsius, T. (2021). ULOS fabric classificatino using android-based convolution neural network. *International Journal of Innovative Computing, Information and Control (ICIC)*, *17*(3), 753-766.
Soliman, M. M., Kamal, M. H., El-Massih Nashed, M. A., Mostafa, Y. M., Chawky, B. S., & Khattab, D. (2019). Violence recognition from videos using deep learning techniques. International Conference on Intelligent Computing and Information Systems (ICICIS) Cairo, Egypt.
Souza, F. D. M., Chávez, G. C., Valle Jr, E. A., & A. Araujo, A. (2010). Violence detection in video using spatio-temporal features. Conference on Graphics, Patterns and Images (SIBGRAPI), Gramado, Brazil.
Su, J., Her, P., Clemens, E., Yaz, E., Schneider, S., & Medeiros, H. (2022, 2022). Violence Detection using 3D Convolutional Neural Networks. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Madrid, Spain.
Sudhakaran, S., & Lanz, O. (2017). Learning to detect violent videos using convolutional Long Short-Term Memory. *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 1-6. https://doi.org/10.1109/ AVSS.2017. 8078468
Sultani, W., Chen, C., & Shah, M. (2018). Real-world anomaly detection in surveillance videos. *CoRR*, *abs/1801.04264*, 6479-6488. https://doi.org/10.48550/arXiv.1801.04264
Sumon, S., Shahria, T., Goni, R., Hasan, N., Almarufuzzaman, A. M., & Rahman, M.

(2019). *Violent crowd flow detection using deep learning* Asian Conference, Intelligent Information and Database Systems (ACIIDS), Yogyakarta, Indonesia.

Sun, S., Liu, Y., & Mao, L. (2019). Multi-view learning for visual violence recognition with maximum entropy discrimination and deep features. *Information Fusion*, *50*, 43-53. https://doi.org/10.1016/j.inffus.2018.10.004

Sun, W., Min, X., Lu, W., & Zhai, G. (2022). A deep learning based no-reference quality assessment model for UGC videos.*MM '22* ACM International Conference on Multimedia, New York, NY, USA.

Suryadevara, N. K., & Mukhopadhyay, S. C. (2014). Determining wellness through an ambient assisted living environment. *IEEE Intelligent Systems*, *29*(3), 30-37. https://doi.org/10.1109/MIS.2014.16

Tahir, R., Ahmed, F., Saeed, H., Ali, S., Zaffar, F., & Wilson, C. (2020). Bringing the kid back into YouTube kids: detecting inappropriate content on video streaming platforms.*ASONAM '19* IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, New York, NY, USA.

Tao, H., & Duan, Q. (2023). An adaptive frame selection network with enhanced dilated convolution for video smoke recognition. *Expert Systems with Applications*, *215*, 119371. https://doi.org/https://doi.org/10.1016/j.eswa.2022.119371

Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J. W., & Carneiro, G. (2021). Weakly-supervised video anomaly detection with contrastive learning of long and short-range temporal features. *arXiv*, *abs/2101.1*, arXiv:2101.2101. https://arxiv.org/abs/2101.10030

Toharudin, T., Pontoh, R., Caraka, R., Zahroh, S., Lee, Y., & Chen, R. (2020). Employing long short-term memory and facebook prophet model in air temperature forecasting. *Communication in Statistics- Simulation and Computation*, *24*, 1-24. https://doi.org/10.1080/03610918.2020.1854302

Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., & Paluri, M. (2014). Learning spatiotemporal features with 3D convolutional networks. *CoRR*, *abs/1412.0*. https://doi.org/10.48550/arXiv.1412.0767

Tyagi, B., Nigam, S., & Singh, R. (2022). A review of deep learning techniques for crowd behavior analysis. *Archives of Computational Methods in Engineering*, *29*(7), 5427-5455. https://doi.org/10.1007/s11831-022-09772-1

Ullah, F. U. M., Ullah, A., Muhammad, K., Haq, I. U., & Baik, S. W. (2019). Violence detection using spatiotemporal features with 3D convolutional neural network. *Sensors*, *19*(11), 2472-2472. https://doi.org/10.3390/s19112472

ur Rehman, A., Belhaouari, S. B., Kabir, M. A., & Khan, A. (2023). On the use of deep learning for video classification. *Applied Sciences*, *13*(3). https://doi.org/10.3390/app13032007

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*, 6000–6010.

Vosta, S., & Yow, K.-C. (2022). A CNN-RNN combined structure for real-world violence detection in surveillance cameras. *Applied Sciences*, *12*(3). https://doi.org/10.3390/app12031021

Wang, X., Yang, J., & Kasabov, N. K. (2023). Integrating spatial and temporal information for violent activity detection from video using deep spiking neural networks. *Sensors*, *23*(9). https://doi.org/10.3390/s23094532

Wimolsree Getsopon, & Surinta, O. (2022). Fusion lightweight convolutional neural networks and sequence learning architectures for violence classification. *International Journal of Innovative Computing, Information and Control(ICIC) : Express Letter Part B : Applications*, *13*(10), 1027-1035. https://doi.org/10.24507/icicelb.13.10.1027

Wu, Z., Yao, T., Fu, Y., & Jiang, Y.-G. (2017). Deep learning for video classification and captioning. *Frontiers of Multimedia Research*, 3-29. https://doi.org/10.1145/3122865.3122867

Xie, S., Sun, C., Huang, J., Tu, Z., & Murphy, K. (2017). Rethinking spatiotemporal feature learning for video understanding. *arXiv*, *abs/1712.0*(1), 5-5. https://arxiv.org/abs/abs/1712.0

Xu, L., Gong, C., Yang, J., Wu, Q., & Yao, L. (2014). Violent video detection based on MoSIFT feature and sparse coding. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy.

Yadav, A., & Vishwakarma, D. K. (2020). A unified framework of deep networks for genre classification using movie trailer. *Applied Soft Computing*, *96*, 106624-106624. https://doi.org/10.1016/j.asoc.2020.106624

Yang, J., Dong, X., Liu, L., Zhang, C., Shen, J., & Yu, D. (2022). Recurring the transformer for video action recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA.

Yang, X., Tang, S., guo, T., Huang, K., & Xu, J. (2022). Design of indoor fall detection system for the elderly based on ZYNQ. *IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, *9*, 1174-1178. https://doi.org/10.1109/ITAIC49862.2020.9338837

Yao, H., & Hu, X. (2023). A survey of video violence detection. *Cyber-Physical Systems*, *9*(1), 1-24. https://doi.org/10.1080/23335777.2021.1940303

Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, *12*(17). https://doi.org/10.3390/app12178643

Zhou, B., Andonian, A., & Torralba, A. (2017). Temporal relational reasoning in videos. *CoRR*, *abs/1711.0*, arXiv:1711.08496-arXiv:01711.08496. https://arxiv.org/abs/abs/1711.0

Zhou, P., Ding, Q., Luo, H., & Hou, X. (2017). Violent interaction detection in video based on deep learning. *Journal of Physics: Conference Series (JPCS)*, *844*, 12044-12044. https://doi.org/10.1088/1742-6596/844/1/012044

Zolfaghari, M., Singh, K., & Brox, T. (2018). Efficient convolutional network for online video understanding. *European Conference on Computer Vision (ECCV)*, *abs/1804.09066*, 695-712. https://doi.org/10.1007/978-3-030-01216-8_43

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA.

พหุน ปณฺ ฑิโต ชีเว

BIOGRAPHY

NAME	Wimolsree Getsopon
DATE OF BIRTH	29 September 1984
PLACE OF BIRTH	Khon Kaen, Thailand
ADDRESS	164/274 Village No.4, Nai Mueang Subdistrict, Mueang District, Khon Kaen, 40000, Thailand
POSITION	Lecturer
PLACE OF WORK	Department of Information System Faculty of Business Administration and Information Technology Rajamangala University of Technology Isan Khon Kaen Campus, Thailand
EDUCATION	2006 Bachelor of Science (B.Sc.) Computer Science, Khon Kaen University, Khon Kaen, Thailand 2008 Master of Science (M.Sc.) Information Technology, Khon Kaen University, Khon Kaen, Thailand 2024 Doctor of Philosophy (Ph.D.) Information Technology, Mahasarakham University, Mahasarakham, Thailand
Research output	Getsopon, W., & Surinta, O.(2022). Fusion Lightweight Convolutional Neural Networks and Sequence learning architectures for violence classification. ICIC Express Letters, Part B: Applications, 1027-1035, Volume 13, Number 10, October 2022.