Detecting Potential Violent Behavior Using Deep Learning

OWOH, Dalton Chukwuezugo

Master's thesis
2024

Tomas Bata University in Zlín
Faculty of Applied Informatics

Tomas Bata University in Zlín
Faculty of Applied Informatics
Department of Informatics and Artificial Intelligence

Academic year: 2023/2024

ASSIGNMENT OF DIPLOMA THESIS

(project, art work, art performance)

Name and surname:	**Dalton Chukwuezugo Owoh**
Personal number:	**A22547**
Study programme:	**N0613A140023 Information Technologies**
Specialization:	**Software Engineering**
Type of Study:	**Full-time**
Work topic:	**Detekce možného násilného chování pomocí hlubokého učení**
Work topic in English:	**Detecting Potential Violent Behavior Using Deep Learning**

Theses guidelines

1. Create a literature review focusing on A.I. techniques for video and image processing.
2. Select available appropriate public datasets for experiments.
3. Select the appropriate methodology with specific focus on the deep learning models.
4. Implement the most suitable one or more models and provide experimental results for selected datasets and models configurations.
5. Provide the analysis of results in terms of accuracy and effectivity.
6. Discuss the capabilities and applicability of A.I. models.

Form processing of diploma thesis: **printed/electronic**
Language of elaboration: **English**

Recommended resources:

1. ROSEBROCK, Adrian. *Deep learning for computer vision with Python*. PyImageSearch, 2017. ISBN 9781986538138.
2. *Advanced methods and deep learning in computer vision*. Computer vision and pattern recognition. London, United Kingdom: Elsevier Academic Press, [2022]. ISBN 9780128221495.
3. GÉRON, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Third edition. Beijing: O'Reilly, [2023].
4. RAMSUNDAR, Bharath a ZADEH, Reza Bosagh. *TensorFlow for deep learning: from linear regression to reinforcement learning*. Beijing: O'Reilly Media, 2018. ISBN 9781491980422.
5. UDUMA, Nikhil. *Fundamentals of deep learning: designing next-generation machine intelligence algorithms*. Beijing: O'Reilly, 2017. ISBN 9781491925614.
6. TUREČKOVÁ, Alžběta. *Deep Learning Methods Applied in Computer Vision*. Zlín: Univerzita Tomáše Bati ve Zlíně, 2023. ISBN 978-80-7678-191-7.
7. AL-DHAMARI, Ahlam; SUDIRMAN, Rubita; MAHMOOD, Nasrul Humaimi. Transfer deep learning along with binary support vector machine for abnormal behavior detection. *IEEE Access*, 2020, 8: 61085-61095.
8. FOUNTA, Antigoni Maria, et al. A unified deep learning architecture for abuse detection. In: *Proceedings of the 10th ACM conference on web science*. 2019. p. 105-114.

Supervisors of diploma thesis: **prof. Ing. Roman Šenkeřík, Ph.D.**
Department of Informatics and Artificial Intelligence

Date of assignment of diploma thesis: **November 5, 2023**
Submission deadline of diploma thesis: **May 13, 2024**

Tomas Bata University in Zlín
Faculty of Applied Informatics
L.S.

doc. Ing. Jiří Vojtěšek, Ph.D. m.p.
Dean

prof. Mgr. Roman Jašek, Ph.D., DBA m.p.
Head of Department

In Zlín January 5, 2024

I hereby declare that:

- I understand that by submitting my Master´s thesis, I agree to the publication of my work according to Law No. 111/1998, Coll., On Universities and on changes and amendments to other acts (e.g. the Universities Act), as amended by subsequent legislation, without regard to the results of the defence of the thesis.
- I understand that my Master´s Thesis will be stored electronically in the university information system and be made available for on-site inspection, and that a copy of the Master´s Thesis will be stored in the Reference Library of the Faculty of Applied Informatics, Tomas Bata University in Zlín, and that a copy shall be deposited with my Supervisor.
- I am aware of the fact that my Master´s Thesis is fully covered by Act No. 121/2000 Coll. On Copyright, and Rights Related to Copyright, as amended by some other laws (e.g. the Copyright Act), as amended by subsequent legislation; and especially, by §35, Para. 3.
- I understand that, according to §60, Para. 1 of the Copyright Act, TBU in Zlín has the right to conclude licensing agreements relating to the use of scholastic work within the full extent of §12, Para. 4, of the Copyright Act.
- I understand that, according to §60, Para. 2, and Para. 3, of the Copyright Act, I may use my work - Master´s Thesis, or grant a license for its use, only if permitted by the licensing agreement concluded between myself and Tomas Bata University in Zlín with a view to the fact that Tomas Bata University in Zlín must be compensated for any reasonable contribution to covering such expenses/costs as invested by them in the creation of the thesis (up until the full actual amount) shall also be a subject of this licensing agreement.
- I understand that, should the elaboration of the Master´s Thesis include the use of software provided by Tomas Bata University in Zlín or other such entities strictly for study and research purposes (i.e. only for non-commercial use), the results of my Master´s Thesis cannot be used for commercial purposes.
- I understand that, if the output of my Master´s Thesis is any software product(s), this/these shall equally be considered as part of the thesis, as well as any source codes, or files from which the project is composed. Not submitting any part of this/these component(s) may be a reason for the non-defence of my thesis.

I herewith declare that:

- I have worked on my thesis alone and duly cited any literature I have used. In the case of the publication of the results of my thesis, I shall be listed as co-author.
- That the submitted version of the thesis and its electronic version uploaded to IS/STAG are both identical.

In Zlín; dated: May 13, 2024 ....

Student´s Signature

ABSTRAKT

V této diplomové práci byly implementovány čtyři modely hlubokého učení - DenseNet-121, Inception-v3, ResNet50 a VGG-16 - k detekci potenciálního násilného chování pomocí principů transfer learningu. V teoretické části byl proveden rozsáhlý přehled literatury v oblasti detekce násilí na lidech s cílem identifikovat převažující silné stránky a mezery ve stávajících výzkumných pracích. Výsledky experimentů provedených v této práci ukázaly nejlepší výsledky s hodnotami přesnosti 98 %. Tato práce kromě jiných klíčových zjištění doporučuje, aby se budoucí výzkum zaměřil na zkoumání zobecnění výsledků tohoto experimentu na větší soubory dat s přizpůsobením širší doméně. Velkým přínosem pro budoucí práci bude také další ladění hyperparametrů modelů s více konfiguracemi.

Klíčová slova: Detekce násilí, Umělá inteleigence, Ladění hyperparametrů, Rozpoznávání vzorů, Konvoluční neuronové sítě

ABSTRACT

In this master's thesis, four deep learning models – DenseNet-121, Inception-v3, ResNet50, and VGG-16 were implemented to detect potential violent behavior by applying transfer learning principles. In the theoretical part, a comprehensive review of literature in the field of human violence detection was conducted to identify prevalent strengths and gaps in existing research work. The results of the experiments conducted in this work showed the best performance with accuracy values of 98%. This work recommends, among other key findings, that future research be geared towards exploring the generalization of results from this experiment across larger datasets with adaptations to a broader domain. The future work will also benefit greatly from further hyperparameter tuning of models with more configurations.

Keywords: Violence detection, AI, Hyperparameter tuning, Pattern recognition, Convolutional Neural Networks

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to the Almighty God for guiding me through my Masters program in Software Engineering at this esteemed institution.

I am deeply indebted to my project supervisor, Doc. Ing. Roman Šenkeřík, Ph.D., whose unwavering support and expertise have been instrumental in shaping my research and academic journey.

I also express appreciation to the Software Engineering Department at Tomas Bata University for providing invaluable opportunities for professional development.

Furthermore, I am grateful to my parents for their unwavering support and sacrifices throughout my graduate studies. Their love and encouragement have been my driving force.

DECLARATION

I hereby declare that the print version of my Master's thesis and the electronic version of my thesis deposited in the IS/STAG system are identical.

This declaration assures that both versions contain the same content, formatting, and structure. By ensuring this consistency, I uphold the integrity and reliability of my academic work, adhering to the standards of scholarly excellence and maintaining transparency in my research process.

CONTENTS

INTRODUCTION

Artificial Intelligence (AI) is a computing concept designed to enable machines to think, solve problems, and learn from mistakes like humans [1]. While traditional robots are limited by fixed programming, AI aims to simulate human-like creative thinking and problem-solving abilities. The field of AI encompasses machine learning, deep learning, and more. Machine learning includes supervised, unsupervised, and reinforcement learning. AI applications range from personal assistants to cybersecurity and healthcare. Despite concerns about AI's impact on jobs and ethics, it offers significant opportunities for new specialized roles and business growth [1].

Within the landscape of AI, the field of deep learning as a subset of machine learning is currently undergoing a lot of development and new 'intelligent' systems are now utilising this technology. There is also a new advancement towards finding the intersection between machine learning and behavioral analysis. This intersection point examines various concepts such as feature engineering, predictive modelling, anomaly detection, development of recommendation systems, user profiling and segmentation systems, sentiment analysis as well as other systems requiring continuous learning and adaptation.

Deep learning itself stems from architectures like deep neural networks, deep belief networks, recurrent networks, convolutional neural networks and transformers [2]. These deep learning architectures have been applied to research areas of computer vision, natural language processing, machine translation, speech recognition, bioinformatics and many other prominent fields.

Deep learning then utilises various machine learning algorithms to progressively extract higher-level features from raw input [2]. This is commonly seen in image processing where lower layers may identify edges, and higher layers identifying human concepts like digits and faces.

Building intelligent machines is at the core of deep learning. The human brain in its innate form can be considered an intelligent system [3]. This is largely because the brain dictates how we see, small, taste and hear. The brain then stands out as the most remarkable component of the body, governing our interpretation of the world through our senses, memory retention, emotional experiences, and even the phenomenon of dreaming. Its absence would reduce us to primitive organisms devoid of any complexity, solely reliant on instinctive responses. This is the objective of deep learning: to replicate an intelligent system.

Detecting potential violent incidents for purpose of assisting crime investigations and consequently preventing re-occurrences remains an ever-present challenge in the world we live in today. Processing visual information is natural for humans, but the manual analysis involved makes manual process a daunting task [4]. Deep learning can play a significant role in automating the detection of potential violent incidents as we have seen with many intelligent systems in use today [5].

Traditional methods of identifying cases of violence rely heavily on reactive measures, only responding to incidents after they occur as opposed to proactively identifying and addressing the risk factors. This reactive way of potential violent behavior detection underscores the importance of adopting deep learning as a means to better anticipate and prevent violent behavior before it escalates into a more severe harm [6].

Various methods for violence detection have been developed in past few years [6]. Convolutional Neural Networks (CNN) are well suited for feature recognition as they can categorize image frames extracted from video clips based on data set and retrieved features.

Figure 1 below shows the fundamental stages of most video-based detection systems.

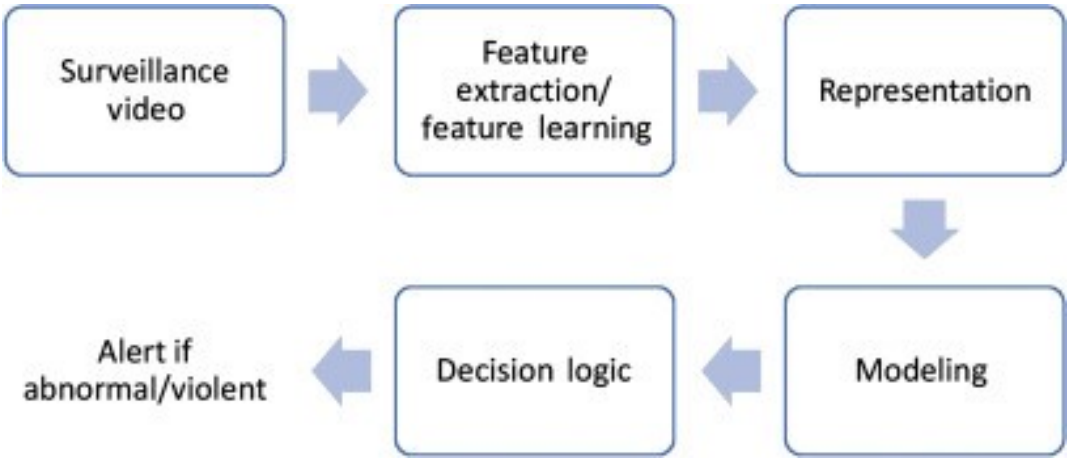

Figure 1 Fundamental stages of video-based violence detection[1]

Spatio-Temporal Interest Points (STIP) represent a specific category of local invariant features utilised in video analysis. These features can withstand alterations such as rotation, changes in scale, affine transformations, and shifts in viewpoint [7]. Among the most commonly employed local invariant features are Harris corners, Scale Invariant Feature

[1] https://ars.els-cdn.com/content/image/3-s2.0-B9780128163856000118-f11-01-9780128163856.jpg

Transform (SIFT), and Speeded Up Robust Features (SURF). These features demonstrated widespread and effective utilization across tasks including object detection and recognition, image registration, image classification, and image analysis [7].

Histogram of Oriented Gradients (HOG) is a feature descriptor widely used in image processing and computer vision tasks. It quantifies the distribution of gradient orientations; HOG captures information about the local edge and texture patterns in an image [8].

CNNs have disrupted the field of computer vision, offering powerful tools for image recognition and analysis [3]. The four model architectures being considered in this work ResNet (Residual Network), VGG (Visual Geometry Group), DenseNet and Inception-v3 have been instrumental in spatial data processing and feature extraction.

This project work focuses on an examination of the effectiveness and accuracy of the four selected CNNs – specifically, ResNet, VGG, DenseNet, and Inception-v3 in detecting potential violent behavior by processing frames from video clips.

In this work, the Real-Life Violence Situations Dataset from Kaggle was utilized for evaluating and comparing different models. This dataset consists of 1,000 videos depicting violence and 1,000 videos showing non-violent situations, sourced from various public platforms.

During the training phase, data augmentation techniques were applied, and consistent optimal weights were employed. Comparative analysis was conducted to assess the performance of four different models in classifying these video frames. A web-based application was developed in Python using Streamlit framework with the primary objective of evaluating the trained models through analysis of the sample videos and facilitating the visualisation of corresponding predictions.

I THEORY

1 ARTIFICIAL NEURAL NETWORKS

ANNs are like computer brains inspired by how our brains work. They help solve tough problems. This section looks at history of ANNs, how ANNs are built, what they do, and where they're used in the real world [9].

The history of ANNs goes back to the 1940s. That was when people first started talking about them [10]. One of the first ideas was the McCulloch-Pitts neuron, suggested by Warren McCulloch and Walter Pitts in 1943 [10]. But things really started moving in the 1950s and 1960s. That was when Frank Rosenblatt came up with the perceptron in 1957 [11]. It was an important moment because it showed that ANNs could be good at recognizing patterns.

During the period spanning 1970s to the 1980s, the interest in ANNs reduced within the academic and scientific circles [12]. This decline in ANN adoption was largely because computers then were not as powerful as they are today. Many scientists during that period also did not have rudimentary understanding of ANNs yet so there was not much drive to advance ANN in this period.

In the late 1980s and early 1990s, we witnessed a growth in the adoption of ANNs. This time, it was because computers were getting better, and we had new ways to teach ANNs how to learn [12]. One important thing was the backpropagation training algorithm. This was invented by a few different people, like Paul Werbos, David Rumelhart, Geoffrey Hinton, and Ronald Williams [12]. Backpropagation made it easier to train ANNs with many layers, so they could solve more complicated problems [12].

The 21st century saw ANNs really take off. This was because we had big sets of data to train them on, and computers were super powerful. We also got better at using deep learning, which means training ANNs with lots of layers. This made ANNs good at tasks like recognizing pictures, understanding language, and even driving cars [13].

Table 1 below summarises the evolution of major events in AI history.

Table 1 Major Events in AI History

Year	Key Researchers	Events in AI History
1943	Warren McCulloch and Walter Pitts	Simplified model of a neuron
1957	Frank Rosenblatt	Perceptron [11]
1970s – 1980s	-	Decline in interest in ANNs
Late 1980s	Paul Werbos, David Rumelhart, Geoffrey Hinton and Ronald Williams	Backpropagation allowing training of multi-layer ANNs
21st Century	-	Big Data, Deep learning

ANNs are pivotal in modern computing, functioning similarly to the human brain and solving very complex problems. The basic form of an ANN is the Neuron which is instrumental in processing information and producing outputs. The Neurons act as decision makers that take in data as input, perform computations on the data and then produce the processed output [3].

In an ANN, neurons are arranged in layers: input layer, multiple hidden layers, and an output layer. Each neuron receives input signals from neurons in the previous layer, processes them using an activation function such as the Sigmoid function shown in Figure 2, and produces and output.

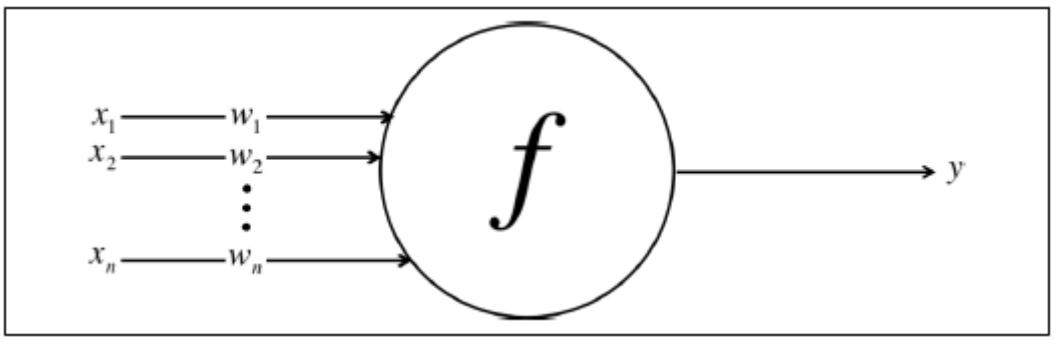

Figure 2 Schematic for a neuron in an ANN [3]

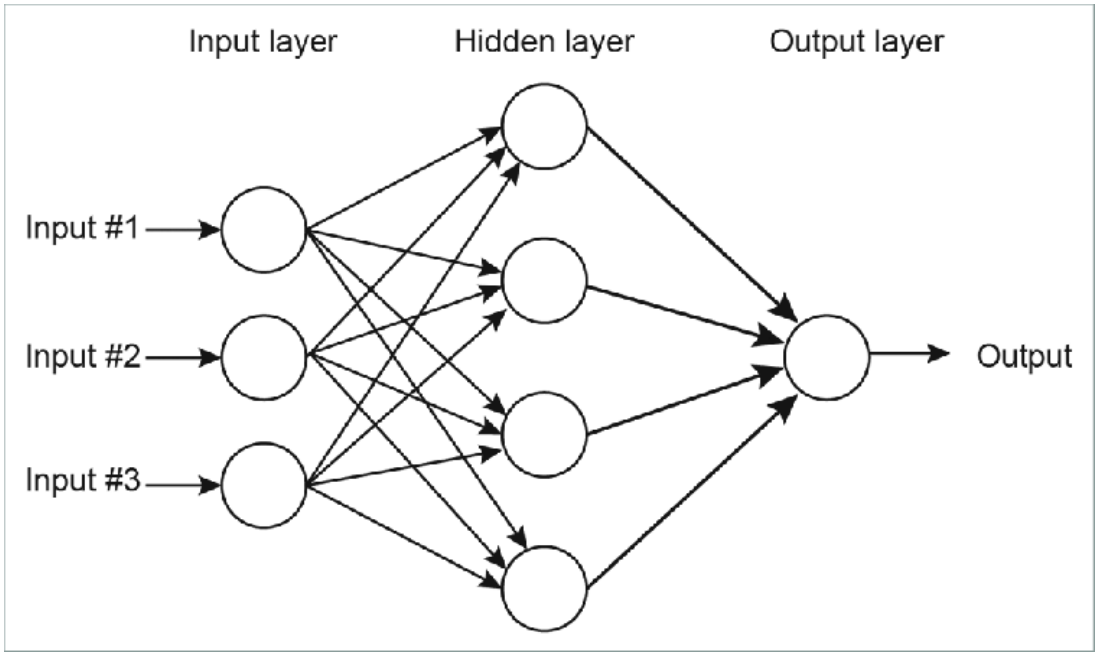

Figure 3 Simple example of a feed-forward neural network with three layers and three neurons per layer[2]

Figure 3 above describes the structure of a feed-forward neural network with three layers and three neurons per layer. These connections between neurons are weighted. The weights represent the strength of each connection. During the training, the network adjusts these weights to minimise the difference between its predicted outputs and the actual outputs, a process known as optimization. This adjustment is typically done using algorithms like backpropagation, where the network iteratively corrects its predictions based on the errors it makes.

The equation labelled (1) below describes the sigmoid activation function with the graph in Figure 4 taking the form of the characteristic S-shape.

$$f(x) = \frac{1}{1 + e^{-z}} \tag{1}$$

[2] https://www.researchgate.net/figure/Structure-of-a-typical-3-layer-feed-forward-multilayer-perceptron-artificial-neural_fig3_259319882

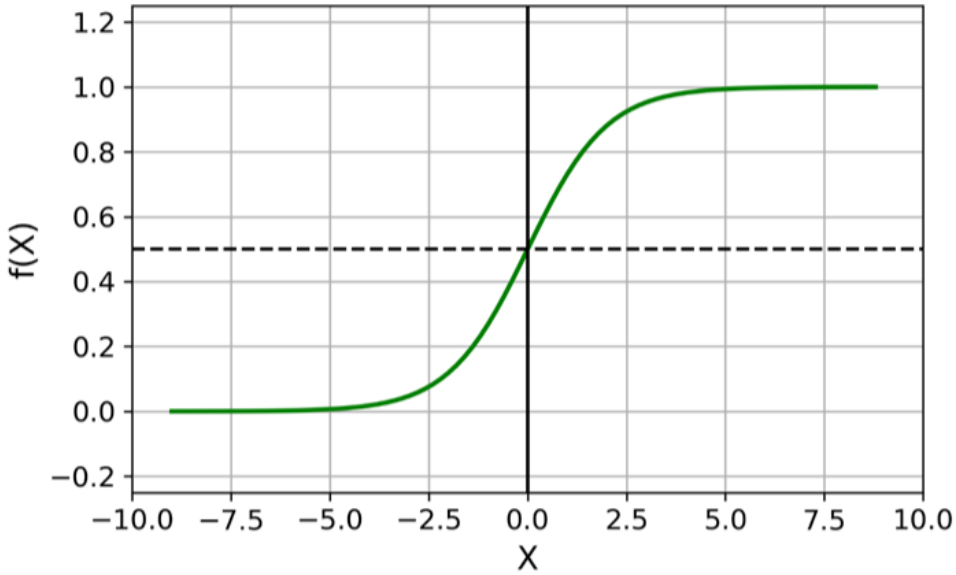

Figure 4 The plot of a sigmoid activation function[3]

In the field of ANN, there are other common activation functions such as Tanh (Hyperbolic Tangent) and ReLU (Rectified Linear Unit).

ReLU is most popular due to its simplicity and effectiveness. It solves the vanishing gradient problem which remains a problem associated with sigmoid and tanh activation functions [3]. ReLU is computationally efficient and most commonly used in the hidden layers of deep neural networks.

ReLU uses a different type of nonlinearity as represented in the equation (2) below and the resulting plot in Figure 5 takes a hockey-stick-shaped output.

$$f(x) = \max(0, z) \tag{2}$$

[3] https://ambrapaliaidata.blob.core.windows.net/ai-storage/articles/Untitled_design_13.png

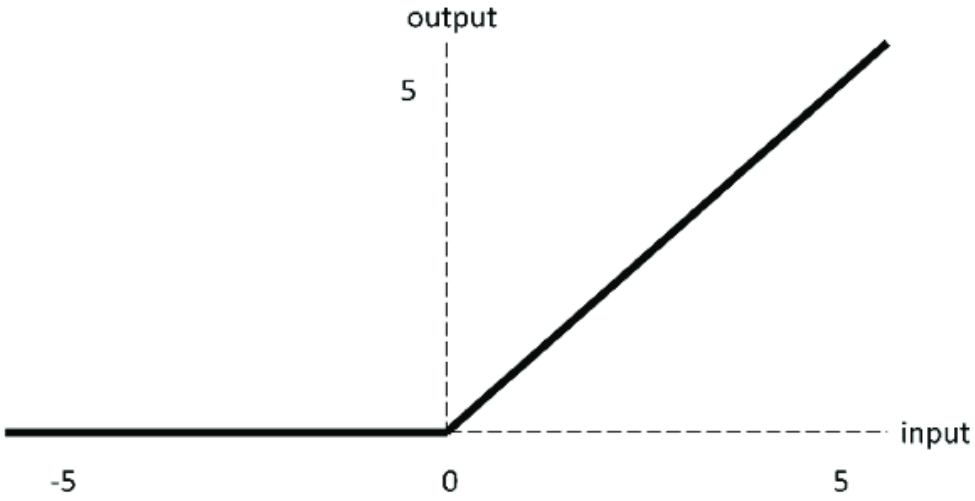

Figure 5 The plot of a ReLU activation function[4]

Tanh is similar to the sigmoid function and use a similar S-shaped nonlinearity in Figure 4, but instead of ranging from 0 to 1, the output of tanh neurons range from -1 to 1 [3].

The mathematical equation for the tanh activation function is given below in equation (3) with the resulting relationship between the output y and the logit z shown in Figure 6 below.

$$f(x) = \tanh(z) \tag{3}$$

[4] https://vidyasheela.com/web-contents/img/post_img/40/ReLU-activation-function-new.png

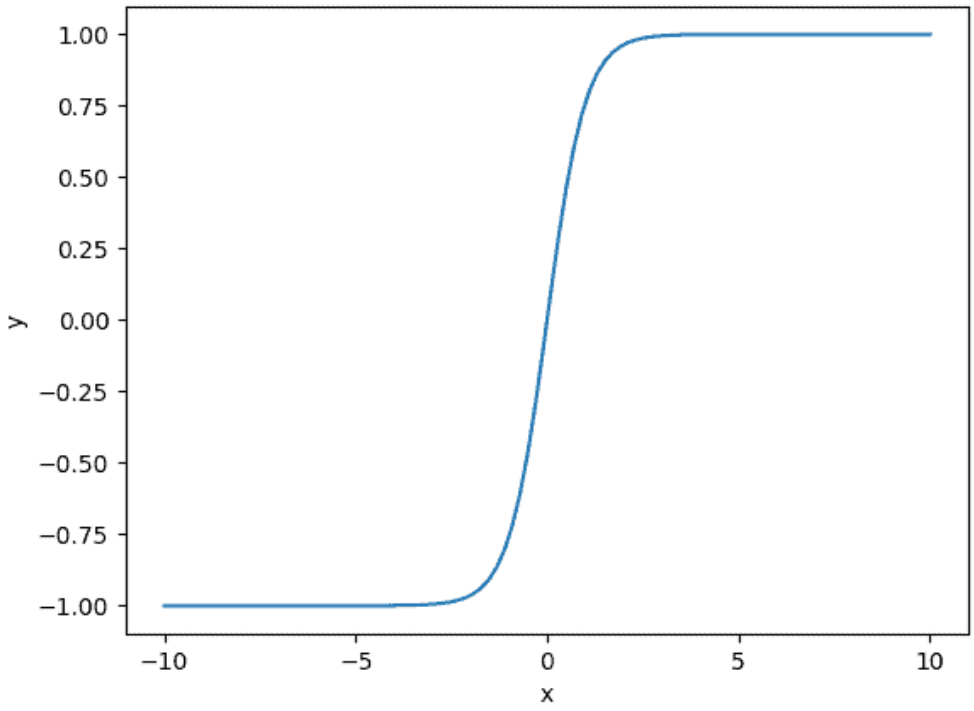

Figure 6 The plot of a tanh activation function[5]

While ReLU is commonly used in hidden layers due to the simplicity and effectiveness in solving the vanishing gradient problem,

I have focused mainly on the Sigmoid activation function for this research due to some reasons below:

a. The sigmoid activation function forces the input values to fall between 0 and 1 which is very helpful in this work because violent behavior detection is a binary classification problem where outputs are interpreted as probabilities

b. Despite the drawbacks from the vanishing gradient problem, it still functions well as the output layer of binary classification problems where the goal is simply to predict probabilities for event occurrence.

[5] https://images.squarespace-cdn.com/content/v1/5acbdd3a25bf024c12f4c8b4/1524687495762-MQLVJGP4I57NT34XXTF4/TanhFunction.jpg

The adoption of the Sigmoid activation function in this work has helped me to normalise prediction outputs between the range of 0 and 1. When the value of *x* is large and positive, the sigmoid function approaches 1, indicating a high probability, while *x* is large and negative, the sigmoid function approaches 0, indicating a low probability.

The derivative of the sigmoid function with respect to its input *x* is computed in the equation (4) below:

$$\frac{d}{dx} sigmoid(x) = sigmoid\ (x)\ .(1 - sigmoid(x)) \tag{4}$$

The gradient of the sigmoid function approach 0 for large positive or negative values of input x, and this affects the rate of convergence during training for Deep Neural Networks (DNNs).

ANN learns input data by adjusting the weights and improving its ability to make correct predictions and classification.

The history of this approach is marked by several milestones as summarised in Table 1 in the earlier chapter. The works of McCulloch and Pitts were pivotal in pushing the breakthroughs in Deep Learning (DL). Since the beginning of AI research, ANNs have seen a lot of changes in its framework for tackling computational problems.

The advancements seen in ANN today is largely due to the efforts of the researchers from the various academic institutions. Their joint efforts towards new research areas and discovery of new approaches in computer vision, natural language processing, healthcare and robotic engineering presents a bright future for ANN.

1.1 NEURAL NETWORKS FOR IMAGE PROCESSING

CNNs are widely used in various domains of image processing [14]. CNNs stand out as a crucial tool in various domains including image analysis, natural language processing and image classification tasks. What sets CNN apart is its remarkable capability to discern and interpret intricate patterns within both visual and textual data.

In the area of image classification, CNNs have reached high levels of accuracy nearly human-level performance. CNNs have be run against benchmark datasets like ImageNet,

allowing for performance of computationally demanding tasks such as object and scene detection, and disease identification in medical scans [15].

Yann LeCun in his groundbreaking paper in 1998 introduced the LetNet architecture [16]. The inception of the LeNet architecture marked a pivotal moment in the quest to efficiently classify 2D images [16]. It emerged as a response to the challenges faced by both overly simplistic neural networks struggling to grasp complex training sets and unwieldy, parameter-heavy networks. LeNet aimed to strike a balance, offering a streamlined convolutional network capable of navigating intricate data landscapes [16].

Two important ideas behind LeNet are feature map system and cascading local convolutional feature maps applied to several hidden layers.

As shown in Figure 7 below, a feature map is generated by initiating a convolution operation on the initial matrix. Equation (5) shows the relationship between the feature map and the input and kernel matrices per the LeNet system architecture.

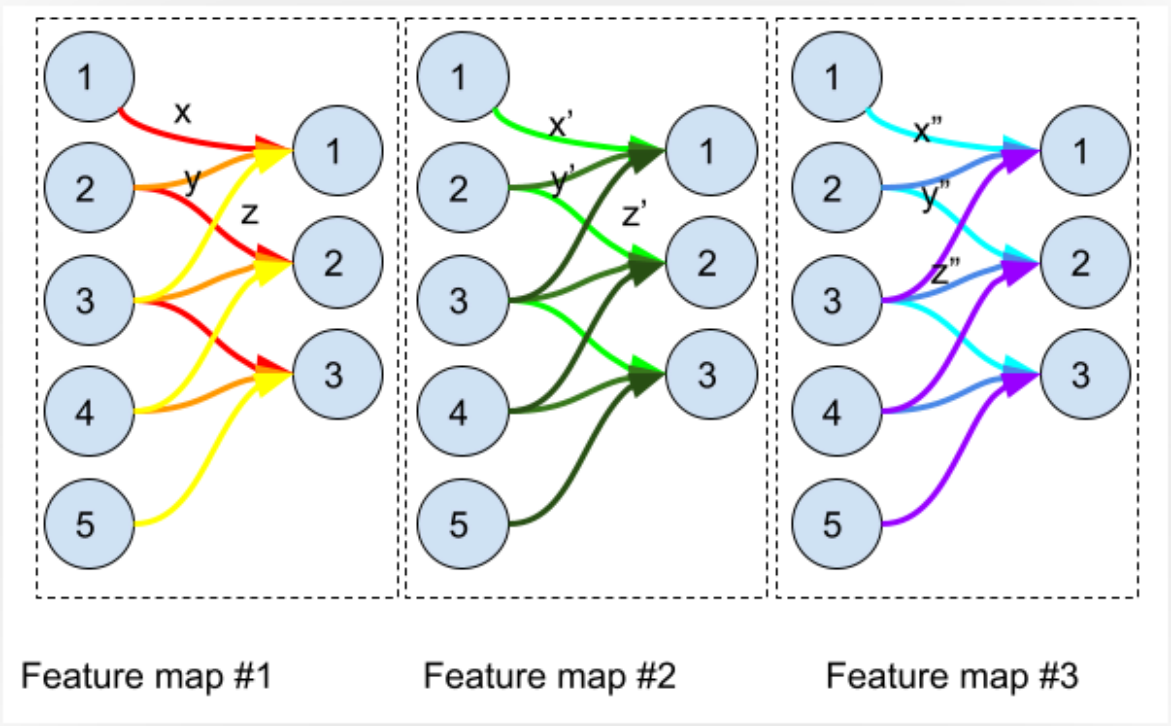

Figure 7 Feature map system for LeNet architecture [6]

[6] https://acodez.in/anatomy-of-the-lenet-1-neural-network/

$$\text{Feature map} = \text{input matrix} * \text{kernel matrix} \tag{5}$$

In the equation (5) above, the convolution operator is represented as '*'. For this convolution operation, only weights inside the kernel matrix will be considered.

The rationale behind favoring local connectivity in neural network architectures, particularly for classifying 2D maps like images, stems from the inefficiency of fully connected layers in capturing the spatial intricacies of the data [16].

Unlike fully connected networks that indiscriminately consider all input pixels regardless of their spatial proximity, architectures like LeNet-1 employ localized connections, emphasizing specific regions through local receptive fields, essentially convolution kernels [16]. By focusing on these small, adjacent subsets of the input, convolutional neural networks (CNNs) efficiently extract relevant features, effectively addressing the spatial nature of the data while reducing computational overhead [16].

In the realm of image recognition, bespoke neural network designs tailored specifically for 2D maps have proven invaluable [16]. These specialized architectures excel at sifting through noise, tackling distortions, and smoothing out fluctuations within input data.

Convolutional networks stand out as the cornerstone of such tailored approaches. They possess the unique ability to hone in on localized patterns, seamlessly integrating these insights to construct a holistic understanding of the input. LeNet embodies this concept, leveraging the power of convolutional networks to distil intricate visual data into actionable insights [16].

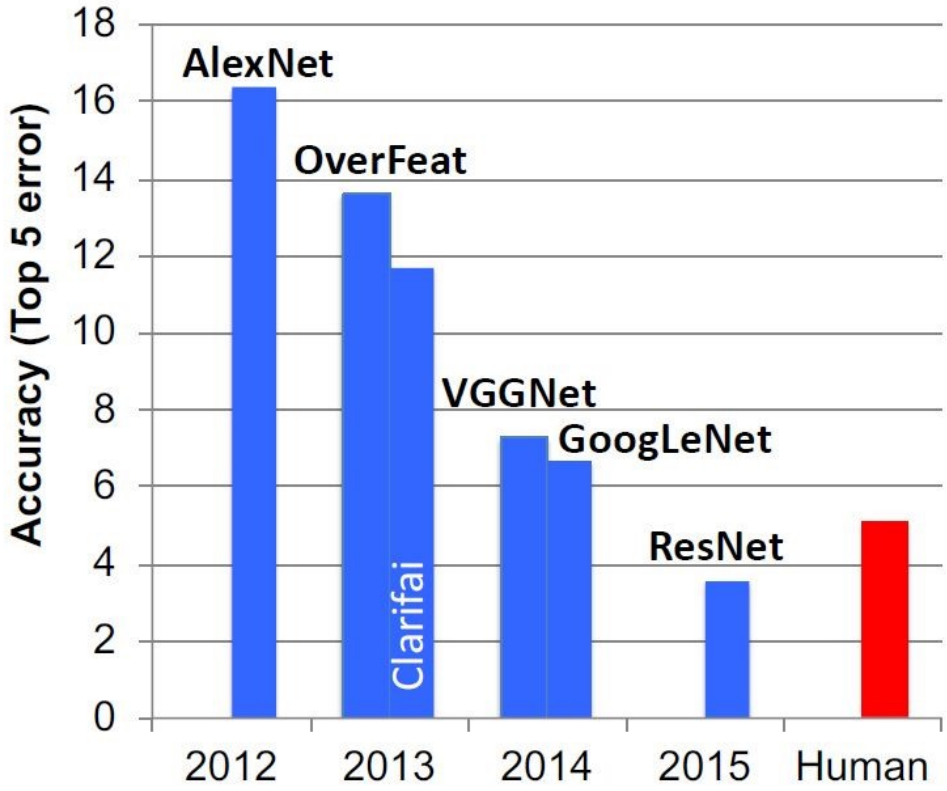

Figure 8 Large Scale Visual Recognition Challenge (LSVRC) 2012 Results [17]

The LSVRC for which results are shown in Figure 8 above is an annual competition aimed at advancing the state-of-the-art in visual recognition tasks, especially in the field of object detection and image classification [17].

Organized by academic and industry leaders in the field, including institutions like Stanford University and Google, LSVRC attracts participation from researchers worldwide. The challenge typically involves tasks such as object localization, where algorithms must identify and precisely locate objects within images, and object classification, where algorithms categorize objects into predefined classes. Participants are provided with large datasets for training their models, which often include millions of labeled images spanning numerous object categories. The competition evaluates submissions based on their accuracy in recognizing objects in unseen images, with a focus on robustness, efficiency, and scalability. LSVRC serves as a benchmark for assessing the progress of computer vision techniques and fostering innovation in the development of algorithms for real-world visual recognition applications.

AlexNet emerged victorious with its innovative architecture featuring five convolutional layers and three fully connected layers. With a total weight of 61 million parameters and 724 million multiply-accumulate operations (MACs), AlexNet showcased its prowess in handling large-scale visual recognition tasks [17].

The convolutional layer configuration for AlexNet is summarised in Table 2 below.

Table 2 AlexNet Convolutional Layer Configurations [17]

Layer	**Filter Size**	**# Filters (M)**	**# Channels (C)**	**Stride**
1	11x11	96	3	4
2	5x5	256	48	1
3	3x3	384	256	1
4	3x3	384	192	1
5	3x3	256	192	1

One of its key strengths lay in its use of ReLU activation functions, which introduced non-linearity to the network, enhancing its ability to capture complex patterns in image data. This groundbreaking achievement solidified AlexNet's status as a milestone in the field of computer vision, setting the stage for further advancements in deep learning and image recognition technologies [17].

VGG-16, a prominent deep convolutional neural network, is renowned for its robust architecture designed for image classification tasks. With a staggering 13 convolutional layers as shown in Figure 9 below, followed by three fully connected layers, VGG-16 excels in extracting hierarchical features from input images at various levels of abstraction [17]. Its substantial weight of 138 million parameters enables it to capture intricate details within images, contributing to its impressive performance [17].

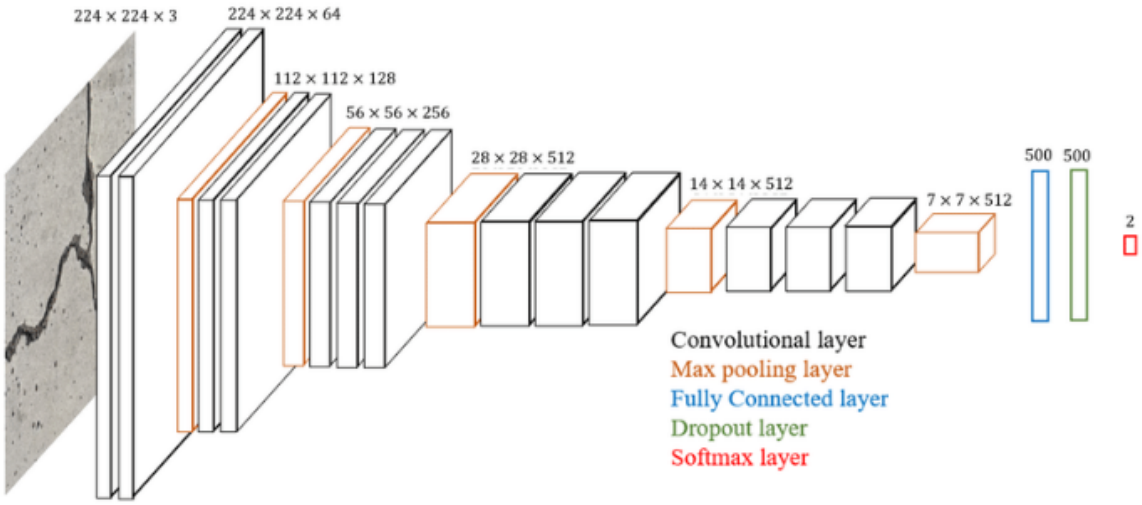

Figure 9 Architecture of VGG-16 CNN model [7]

Moreover, with a remarkable computational efficiency reflected in its 15.5 billion multiply-accumulate operations (MACs), VGG-16 demonstrates its capability to handle large-scale visual recognition tasks with speed and accuracy. This combination of depth, parameterization, and computational efficiency has positioned VGG-16 as one of the leading models in the field of deep learning, inspiring further advancements in image analysis and computer vision applications [17].

The future of neural networks in image processing holds promise, with ongoing research focusing on improving model efficiency, interpretability, and generalization. Techniques like attention mechanisms and self-supervised learning aim to enhance the performance of neural networks in understanding complex visual content. However, challenges such as data privacy, ethical considerations, and the environmental impact of large-scale training remain pertinent.

[7] https://www.researchgate.net/figure/Architecture-of-the-modified-VGG16-model_fig1_350828239

1.2 LIMITATIONS OF TRADITIONAL NEURAL NETWORKS IN IMAGE PROCESSING

The advent of traditional neural networks, particularly CNNs, has significantly reshaped the landscape of image processing. However, amid their notable achievements, these models exhibit inherent limitations that hinder their efficacy in specific contexts [18].

Traditional neural networks operate hierarchically, progressively extracting abstract features from input images. Despite their proficiency in capturing local patterns, they often fail in achieving a comprehensive understanding of spatial relationships within the image [18]. Consequently, tasks requiring nuanced spatial reasoning, such as semantic segmentation or image manipulation, present formidable challenges [17].

A prominent limitation of traditional neural networks lies in their susceptibility to variations and distortions in input images. Minor disruptions, such as rotations or occlusions, can significantly affect model performance. This vulnerability emanates from the fixed receptive fields of convolutional filters, thereby impeding the models' generalization capacities to novel data or variations in image characteristics [19].

The opaque and intricate nature of internal representations learned by traditional neural networks presents a formidable challenge in interpretation. While good at extracting hierarchical features from raw pixel data, discerning the semantic underpinnings of these representations remains difficult to catch. This dearth of interpretability creates doubts regarding the reliability of neural network predictions, particularly in domains requiring high interpretative fidelity, such as medical diagnosis or autonomous driving [14].

The training and deployment of traditional neural networks, notably deep CNNs, exact substantial computational overheads and memory requisites [19]. The magnitude of parameters and layers mandates adequate computational resources, constraining accessibility to researchers and practitioners lacking access to high-performance computing infrastructure. Additionally, the deployment of such models on resource-constrained devices poses logistical challenges due to memory and power constraints [3].

Traditional neural networks are susceptible to adversarial attacks, wherein small changes to input images can induce misclassification or erroneous predictions. This susceptibility, rooted in the linear and non-robust nature of neural network activations, enables attackers to exploit small deviations in the input to manipulate model outputs. Consequently, traditional neural networks may not be reliable in situations where security is very important [19].

While traditional neural networks have sparked big changes in how we process images, their limits show that we still need to keep coming up with new ideas in this area. To deal with these limits, we need to work together across different fields like machine learning, computer vision, and cognitive science. If we can solve these problems, we can make models for image processing that are stronger, easier to understand, and use less computing power. This would make them useful in many different real-world situations [19].

2 OVERVIEW OF AI TECHNIQUES IN VIDEO AND IMAGE PROCESSING

Artificial Intelligence (AI) has witnessed a lot of advancements in recent years, significantly impacting video and image processing domains. This work addresses the main AI techniques used in these fields, highlighting their principles, applications, and implications [15].

In the following chapters, CNNs, Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), Transfer Learning, and Attention Mechanisms are discussed, describing their roles in reshaping visual information processing [15]. As reviewed in [15], CNNs have been instrumental in image classification tasks, marking a pivotal advancement in AI-driven image processing.

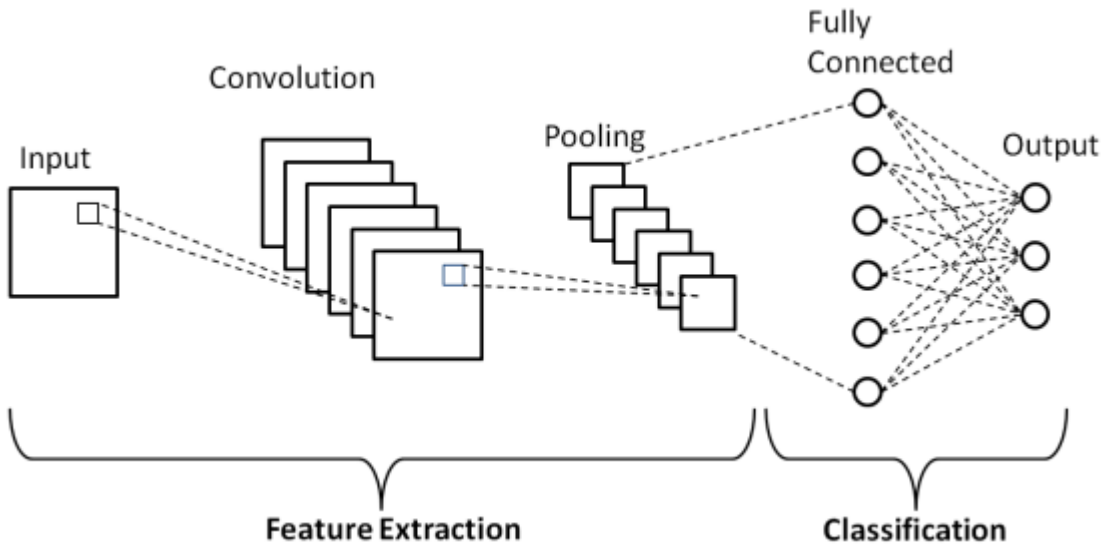

Figure 10 Diagram of a convolutional neural network architecture[8]

Figure 10 above describes a basic CNN consisting of several layers that process input data in a hierarchical manner, extracting features at different levels. The input layer is where we feed raw input into the network. In the context of image processing, each pixel from the image dataset is considered an input neuron. The convolutional layer serves to apply a set of filters known as kernels to the input data. Convolution is performed on this layer to produce

[8] https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26_fig1_336805909

feature maps [18]. The purpose of feature maps is to capture spatial hierarchies of patterns in the input data. Activation functions like ReLU is applied to the network to introduce non-linearity thereby allowing the model learn very complex patterns. The pooling layer serves to reduce the spatial dimensions through a method of downsampling [18]. Fully connected layers allow a strong interconnection between the current layer and all previous layers enabling the model to learn global patterns in the feature maps extracted by the convolutional layers [18].

CNNs have emerged as foundational tools in image processing due to their use in automatically extracting hierarchical features from images [15]. Krizhevsky et al. [15] pioneered this domain with the seminal work on AlexNet, showcasing CNNs' applicability in image classification tasks. Since then, architectures like VGGNet [18] and ResNet [14] have further propelled CNNs' capabilities, fostering breakthroughs in object detection and semantic segmentation [18].

RNNs on the other hand are good at interpreting sequential information. In video processing, RNNs, particularly Long Short-Term Memory (LSTM) networks, play an important role in capturing temporal dependencies. Hochreiter and Schmidhuber [20] introduced LSTM networks, enabling applications such as action recognition and video captioning. By modeling sequential data, RNNs facilitate nuanced analysis of video content, augmenting comprehension and descriptive capabilities.

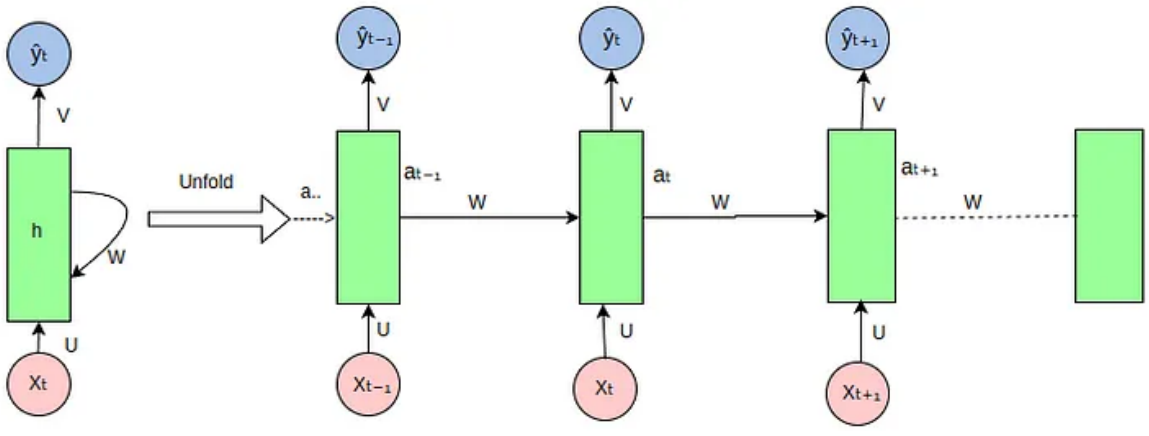

Figure 11 Diagram of a recurrent neural network architecture[9]

[9] https://medium.com/@poudelsushmita878/recurrent-neural-network-rnn-architecture-explained-1d69560541ef

Figure 11 above shows the most basic form of a RNN architecture which is made up of three components: an input layer, hidden layer, and an output layer. The input layer serves to accept input represented in vectors and fed into the system one at a time. The hidden layer is made up of recurrent connections which creates the concept of memory and information retention. The hidden layer takes both current input and output from prior states to compute the hidden state. The hidden state is used in running predictions an fed back into the network for subsequent steps in the network.

GANs have revolutionized content generation tasks by learning to generate realistic images and videos. Goodfellow et al. [9] introduced GANs as a novel framework comprising a generator and a discriminator network trained adversarially. This paradigm has fostered advancements in video synthesis, super-resolution, and prediction, heralding a new era of visual content creation.

Transfer Learning has gained prominence in scenarios with limited labeled data, leveraging pre-trained models to enhance performance on specific tasks. Yosinski et al. [21] demonstrated the efficacy of transfer learning in image processing, reducing computational costs while maintaining competitive performance. By fine-tuning pre-trained models on domain-specific data, practitioners expedite model training and mitigate data scarcity challenges.

Attention Mechanisms is inspired by human visual attention and enable models to focus on salient regions within images and videos. Vaswani et al. [22] introduced Transformer architectures, employing attention mechanisms to selectively weigh different parts of input data. In image processing, attention mechanisms have bolstered tasks like image captioning and object detection, enhancing model interpretability and performance [22].

Table 3 below summarises the various AI techniques and applications.

Table 3 AI Techniques and Applications [13]

AI Technique	Applications
CNNs	Image classification
	Object detection
	Semantic segmentation
	Facial recognition
RNNs	Action recognition
	Video captions
	Video summary
GANs	Image generation
	Video synthesis
	Super-resolution
Transfer learning	Fine-tuning pre-tuned models for specific tasks
	Addressing data securely
Attention Mechanisms	Image caption
	Object detection with attention
	Video summary with attention

AI has brought about significant changes in how we process videos and images. With techniques like CNNs, RNNs, GANs, Transfer Learning, and Attention Mechanisms, machines can now understand, create, and analyze visual content more accurately and efficiently than ever before [14].

CNNs, for instance, help in recognizing objects in images, while RNNs are great at understanding videos by looking at the sequence of frames [18]. Generative Adversarial Networks are excellent at creating realistic images and videos, and Transfer Learning allows machines to learn from one task and apply it to another, saving time and resources. Attention

Mechanisms help models focus on the most important parts of an image or video, improving their performance in tasks like image captioning and object detection [13].

As these AI techniques continue to evolve, we can expect even more breakthroughs in video and image processing. This means better quality images, more accurate object detection, and even more realistic video synthesis. With AI, the possibilities in visual content creation and analysis are endless, promising a future where machines can truly see and understand the world around us in remarkable ways [5].

3 APPLICATION OF AI IN DETECTING VIOLENT BEHAVIOR

AI has transformed many sectors, but most importantly in the safety and security sector, a crucial role is being played in detecting violent behavior. This is possible due to the huge amounts of datasets and advanced deep learning model capable of studying patterns across visual datasets depicting violence behaviors.

AI assists greatly in violent behavior detection through deep learning, a subset of machine learning. Detecting violence can be useful in many places like soccer stadiums, cameras watching streets, and various video sharing platforms such as YouTube and Vimeo [23]. More practical applications of AI in violence detection include monitoring platforms like Facebook, Snapchat, Instagram, TikTok as well as many other popular sites, where AI tools flag potentially violent content for review [24]. There have been cases in the past where AI had spotted violent user-generated content on Facebook and aided law enforcement in responding quickly to de-escalate the situation[10].

However, it is sometimes difficult for people to monitor these videos in real-time because there is so much of these videos to looks at. AI excels at this by quickly spotting violence through automated image processing and informing the authorities so they can act fast in cases that require immediate attention [23].

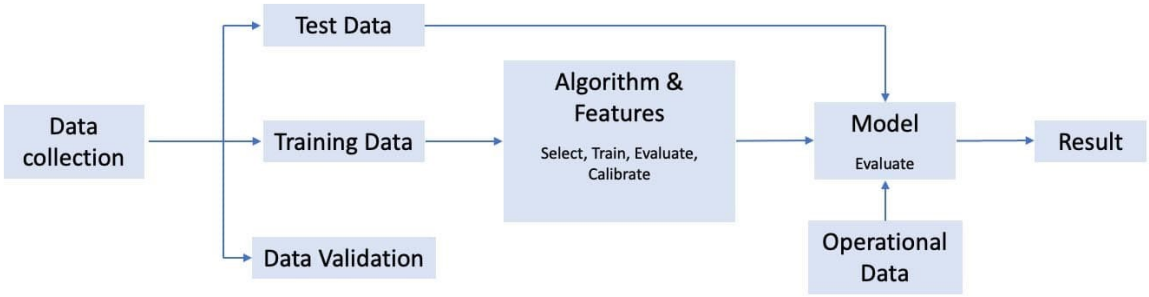

Figure 12 Steps involved in developing AI-based systems[11]

The processes involved in building AI systems as shown in Figure 12 above outlines the sequential steps necessary for system development. These steps are very important for making AI systems work well and be trustworthy [2]. Starting from collecting and preparing

[10] https://about.fb.com/news/2021/12/metas-new-ai-system-tackles-harmful-content/

[11] https://kvalito.ch/taking-shape-artificial-intelligence-regulation-and-its-impact-on-csv-csa-iii/

data to training the model and checking how well it works, each part is crucial for making the system better. Also, after putting the system into use, it's important to keep an eye on it and make improvements over time. Due to the number of steps involved, it is important that we are careful and evaluate every step to get the results we want [3].

AI technologies have shown significant potential in augmenting traditional methods of violence detection, offering innovative solutions to identify patterns and indicators of violent behavior [25]. Violence, whether physical or verbal, poses significant challenges to societies worldwide, making its detection and prevention crucial [26].

4 REVIEW OF RELATED AND EXISTING STUDIES ON VIOLENT BEHAVIOR DETECTION USING AI

This chapter contains extensive research I conducted on existing literature and systems. Previous efforts in visual processing in deep learning were acknowledged, and their strengths and potential gaps examined.

Out of numerous papers of relevance, the following were deemed directly relevant to this work:

I A Machine Learning Approach to Detect Violent Behaviour from Video [5].

In this paper, the authors use a machine learning approach to identify violent behavior in videos by encoding visual features into a vector, processed by a convolutional LSTM network, followed by classification through fully connected layers. Alongside traditional features like angles, velocity, and contact between individuals, the authors incorporated temporal information to construct a feature vector for a binary classification SVM model, aiming to predict violent behavior.

The study utilizes the ISR-UoL 3D Social Activity Dataset, encompassing 93660 RGB images of multi-person actions across 10 sessions, each featuring 8 distinct acts performed by unique pairs of individuals. The dataset captures personal nuances during actions to enhance generalization, with acts sometimes split into four mini-recordings. Various human actions like handshake, hug, fight, push, talk, and draw attention performed by a group of 6 individuals serve as the basis for classification. Notably, actions deemed aggressive or violent, such as push and fight, are considered around 31% of frames, while others are categorized as non-violent.

Potential gaps in the research on action recognition include the need for more focus on real-world applications and practical implementations.

II State-of-the-art Violence Detection Techniques in Video Surveillance Security Systems: A Systematic Review [6].

The systematic review by Omarov et al. provides a comprehensive assessment of video violence detection techniques. The study analyzes 80 research papers from 2015 to 2021, sourced from digital libraries and computer vision conferences. The paper categorizes methods into conventional, deep learning-based, and machine learning-based approaches, highlighting the importance of datasets and evaluation criteria in violence detection. Notably, the review captures the increasing trend of deep learning techniques, especially convolutional neural networks, in addressing violence detection challenges.

In examining the presented datasets, the paper identifies the popular datasets for violence detection and the methods used for abnormal behavior classification. However, a potential gap lies in the limited focus on datasets directly relevant to violence detection, leaving room for a more detailed analysis of dataset suitability and diversity in violence detection research.

III Transfer Deep Learning Along with Binary Support Vector Machine for Abnormal Behavior Detection [27].

Abnormal behavior detection in various scenarios has been a subject of significant research efforts in recent literature. Studies such as the one by Zenati et al. introduce novel frameworks like Bidirectional GAN, emphasizing the importance of encoder E, generator G, and discriminator D in mapping latent representations during training. Furthermore, the work by the authors explores abnormal behavior detection in diverse environments using CNN, showcasing the adaptability of models to different background settings and subject numbers.

On the other hand, Gnouma et al. propose an innovative approach centered around the history of binary motion images (HBMI) for human activity recognition. This method leverages silhouettes of human activities based on characteristics represented through background subtraction techniques like MOF and GMM. Additionally, the utilization of stacked sparse autoencoders (SSAE) in

automating human activity detection underscores the significance of unsupervised feature learning in capturing high-level pixel intensity features efficiently.

Despite the progress made in abnormal behavior detection literature, there exist potential gaps that warrant further investigation. One apparent gap is the need for research focusing on the integration of multiple modalities for enhanced detection accuracy. Incorporating sensor data, textual information, or contextual cues alongside visual data could potentially enrich the existing detection models and improve overall performance.

IV Violent Interaction Detection in Video Based on Deep Learning [26].

The paper by Peipei Zhou et al. ventures into the burgeoning field of automated video surveillance by leveraging deep learning techniques to detect violent interactions in video footage. This study incorporates advanced computational models that exhibit the capability to analyze visual content for aggressive behavior detection, thus enhancing security and surveillance systems. The work primarily integrates deep learning frameworks which are adept at handling large and complex datasets, allowing for more nuanced and accurate interpretations of dynamic scenes.

The paper focuses on the detection of violent interactions using deep learning, but it does not emphasize the efficiency and speed of these detections in real-time applications. Real-time processing is crucial for immediate intervention in security systems.

While the paper by Peipei Zhou et al. marks significant advancements in using deep learning for detecting violent interactions in videos, addressing these gaps can propel the research forward, making it more applicable and robust in real-world scenarios.

V Violent Flows: Real-Time Detection of Violent Crowd Behavior [25].

The paper on violence detection contributes significantly to the field by proposing a method that efficiently labels ViF descriptors as either violent or non-violent using a standard linear Support Vector Machine (SVM). The authors

assembled their own collection of videos, encompassing both violent and non-violent crowd behaviors, to test the accuracy of their method. They compared their approach with existing state-of-the-art techniques on violence classification and detection benchmarks designed using this collection, showcasing a clear performance advantage in favor of their proposed method.

Despite the valuable contributions made by the paper, several potential gaps and areas for further exploration emerge. One such gap revolves around the limited availability of comprehensive video collections for testing violence detection performance, as highlighted by the authors themselves. The lack of focused benchmarks specifically addressing the described problem poses a challenge for researchers in this domain.

While the paper presents a valuable contribution to violence detection methodologies, addressing the identified gaps could further elevate the robustness and applicability of violence detection techniques in diverse real-world setting.

VI Weakly-Supervised Violence Detection in Movies with Audi and Video Based Co-training [28].

In the presented paper by Jian Lin and Weiqiang Wang, the authors introduce a novel methodology for detecting violent content in movie scenes by deploying a dual-view (audio and video) weakly-supervised approach which leverages co-training to increase detection accuracy. This study significantly contributes to the domain of content moderation and multimedia processing by integrating distinct yet complementary sensory data streams—a strategy that remains under-explored in the field.

While the paper outlines a robust framework for violence detection, certain gaps can be explored further. Firstly, the granularity of violence levels is not addressed. All violent incidents are treated with the same severity, which may not be suitable for all applications. Future research could focus on classifying the intensity or type of violence, providing a more nuanced content moderation tool.

VII Person-on-Person Violence Detection in Video Data [29].

The system's methodology and assumptions are backed by a range of studies indexed in the provided references. For example, the work by Kuno et al. on the automated detection of humans is fundamental in formulating algorithms that can discern human figures and activities within a video feed. Similarly, Stauffer and Grimson's exploration of activity patterns using real-time tracking has likely contributed to refining the motion detection algorithms and enhancing temporal consistency, which is key to the detection system's effectiveness in dynamic environments. In the domains of video content characterization and scene recognition, contributions by Vasconcelos and Lippman, along with Nam and Alghoniemy, provide critical frameworks and methodologies that contribute to understanding the semantically meaningful feature spaces necessary for accurate activity detection.

The system tested across various scenarios, including determining the difference between violent and non-violent human interactions, indicates a robust application of the aforementioned studies. The focus on not just static imagery but continuous video feeds allow for a comprehensive approach to surveillance, a distinct improvement over traditional systems that may rely more heavily on static image analysis. The introduction of dual analytical methods further exhibits an innovative approach by enhancing the system's reliability in real-world scenarios, allowing it to handle varied environmental and human factors effectively.

Despite the advancements, the document clearly outlines several limitations and potential areas for future development. One major gap is the system's inability to accurately handle scenarios where individuals are not upright or are engaging in complex interactions like wrestling. This highlights a need for advanced algorithms capable of understanding more complex human postures and interactions beyond basic violent actions.

VIII Audio-Visual Content-based Violent Scene Characterization [30].

The paper by Nam, Alghoniemy, and Tewfik introduces a novel technique for characterizing and indexing violent scenes in TV dramas and movies. The authors address the existing reliance on low-level visual feature analysis in video indexing schemes, emphasizing the need for higher-level features to enable semantically meaningful information retrieval. They highlight the limitations of current approaches in capturing conceptual meanings, particularly in identifying specific events of interest across different film genres.

The authors propose a high-level indexing scheme that merges multiple audio-visual signatures to create a perceptual relation for identifying violent scenes, aiming to support video indexing at a more substantial conceptual level. They underscore the importance of effectively combining different low-level audio-visual features and associating them with conceptually meaningful violent content, illustrating a practical example of query by semantic subject.

One notable gap in the existing research, as highlighted by the authors, is the limited focus on detecting action or violent content using a single source of information (either audio or visual track data alone). For instance, while some studies have used video shot activity and duration as features to categorize movies based on violence, these criteria may not be sufficient to differentiate violent actions from highly active non-violent content, such as sports videos. Similarly, relying solely on audio-based violence detection may lead to false positives due to the complexity of background audio tracks mixing various sounds.

The proposed technique by Nam et al. addresses this gap by integrating cues from both video and audio tracks to characterize violent scenes, acknowledging the high correlation between these modalities during violent events. This approach leverages spatio-temporal dynamic activity signatures in video shots and sound effects embedded in the soundtrack to provide a more comprehensive analysis of violent content.

The work by Nam, Alghoniemy, and Tewfik contributes significantly to the field of audio-visual content-based violent scene characterization by proposing

a novel technique that bridges the gap between existing single-source approaches. Future research could focus on enhancing the robustness and scalability of the proposed method across diverse genres and settings to further advance the field of violent scene identification in audio-visual media.

IX Detection of Violent Events in Video Sequences based on Census Transform Histogram [31].

The paper under review presents a comprehensive analysis of various methods applied to the Hockey Fights Dataset, focusing on different feature descriptors and classification techniques. The study evaluates the performance of methods such as HOG + BoW, HOF + BoW, MoSIFT + BoW, and more, in terms of accuracy percentage. These analyses shed light on the efficacy of the different combinations in accurately classifying instances within the dataset.

The work done in this paper showcases a meticulous exploration of feature extraction and classification methods in the context of analyzing hockey fight videos. By testing various combinations such as MoWLD + SparseCoding, MoIWLD + KDE + SparseCoding, and SRC among others, the researchers provide a detailed comparison of the effectiveness of each method in achieving high accuracy levels.

However, despite the thorough investigation conducted in the study, there are potential gaps that warrant further attention. One notable gap is the limited exploration of deep learning techniques, which have shown promising results in various computer vision tasks. Integrating deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), could potentially enhance the classification performance on the Hockey Fights Dataset.

Moreover, the paper primarily focuses on traditional feature-based methods and lacks exploration of the latest advancements in the field of computer vision, such as attention mechanisms or transformer models. Incorporating these modern approaches could possibly lead to improved accuracy and robustness in classifying violent interactions in hockey videos.

While the paper provides a valuable insight into the performance of different feature descriptors and classifiers for analyzing hockey fight videos, there exist opportunities for future research to delve into incorporating deep learning and cutting-edge techniques to address the identified gaps and further enhance the accuracy and efficiency of violence detection in sporting events.

X Recognizing Human Actions in Surveillance Videos [32].

The paper focuses on developing a robust human action recognition system for real-world surveillance videos. The authors address the importance of local spatio-temporal features around interest points for effective video analysis and motion recognition. They introduce an algorithm named MoSIFT, which goes beyond traditional approaches by explicitly capturing local motion information in addition to local appearance. By detecting distinctive local features and constructing MoSIFT feature descriptors akin to SIFT descriptors, the authors aim to enhance robustness to small deformations through grid aggregation. Moreover, the incorporation of a bigram model to establish correlations between local features signifies an attempt to capture the global structure of actions. The proposed method achieves a notable improvement in accuracy on the KTH dataset, reaching 95.8%. Furthermore, the application of the approach to 100 hours of surveillance data in the TRECVID Event Detection task shows promising results in human action recognition in real-world surveillance scenarios.

While the paper presents significant advancements in human action recognition in surveillance videos, several potential gaps merit consideration for future research. Firstly, a deeper exploration of the scalability and real-time performance of the MoSIFT algorithm could enhance its practical utility in large-scale surveillance systems. Secondly, assessing the algorithm's performance across diverse environmental conditions, such as varying lighting or occlusions, would provide insights into its robustness in challenging real-world scenarios. Additionally, further investigation into the adaptability of MoSIFT to different types of actions and movements could broaden its applicability beyond the current scope.

XI Violence Detection using Oriented Violent Flows[33].

In the realm of violence detection in videos, previous studies have shown a combination of vision and acoustic technologies. While some surveillance systems incorporate both modalities, audio cues are often unavailable, leading to a predominant reliance on visual methods. Early efforts by Datta et al. focused on violence detection through background subtraction. However, limitations arose when violence occurred in crowded environments, indicating potential challenges with this approach.

Furthermore, the presence or absence of blood has been identified as a crucial cue for violence recognition in some studies. Nevertheless, when surveillance cameras only provide grayscale videos, the effectiveness of blood-based approaches may diminish. Recent advancements have seen the use of local interest-point methods by Clarin et al. for detecting fights. Additionally, Nievas et al. introduced a novel descriptor, ViF, tailored for real-time crowd violence detection.

Notably, the creation of benchmark datasets such as Hockey Fights and Violent-Flows has significantly contributed to the evaluation of violence detection algorithms. These databases provide a standardized platform for testing different approaches in violence detection, thus enhancing the reproducibility and comparability of research outcomes. However, there remains a gap in the literature regarding the adaptation and evaluation of these vision-based methods in real-world scenarios outside controlled environments. Future research should focus on the robustness and generalizability of these algorithms in diverse settings to enhance the practical applicability of violence detection technologies.

XII Abnormal Behavior Recognition for Intelligent Video Surveillance Systems: A Review [34].

Previous studies have predominantly focused on human action recognition within the realm of computer vision. Noteworthy research has been conducted in areas like video surveillance, scene modeling, and video content annotation and retrieval. Various surveys have delved into human motion detection,

behavior analysis, and activity recognition, emphasizing the significance of these endeavors across different applications. Surveys by Aggarwal & Cai, Ji & Liu, Pantic et al., and Shian-Ru et al. have laid the foundation for understanding human actions in video contexts. Furthermore, recent reviews by Dawn et al., Hassan et al., and Bux et al. have explored computer vision techniques for recognizing simple activities and phases of human activity recognition. Notably, certain studies by Sarvesh & Anupam and Mishra & Bhagat have concentrated on motion analysis and activity recognition specifically in video surveillance applications.

However, despite the intense focus on human action recognition, there exists a noticeable research gap concerning abnormal behavior detection - a crucial facet of ensuring safety in surveillance settings. With the proliferation of surveillance cameras, the challenge of detecting abnormal events has intensified, prompting the need for automated surveillance systems capable of identifying anomalies and triggering alerts. While some literature reviews have touched on anomaly detection within surveillance systems, such as Valera & Velastin and Oluwatoyin & Kejun, the concentration on abnormal behavior detection remains relatively limited. The scarcity of in-depth exploration in this area presents a promising avenue for further research to enhance the efficacy and accuracy of abnormal event detection in video surveillance setups.

XIII Recognizing Violent Activity Without Decoding Video Streams [35].

In the paper "Recognizing Violent Activity without Decoding Video Streams" by Xie et al., the authors propose a novel method for recognizing violent activities based on motion vectors extracted directly from compressed video data. The approach involves analyzing motion vectors to generate a Region Motion Vectors descriptor (RMV) and utilizing Support Vector Machine (SVM) classification with a radial basis kernel to determine the presence of violent activities in videos with high accuracy (96.1%) and low false probability (5.1%). The method's efficiency also allows for potential integration into embedded systems, showcasing practical applicability.

The work builds upon existing activity recognition methods by focusing on violent activity detection, which is crucial for enhancing video surveillance systems' effectiveness. By leveraging motion vectors from compressed video data, the proposed method bypasses the need for intensive target detection and tracking processes, addressing limitations present in traditional recognition approaches. Furthermore, the study conducts experiment on diverse datasets, showcasing the method's robustness and high performance in identifying violent activities.

While the paper presents a significant advancement in violent activity recognition, there are potential gaps that future research could address. One key area for further exploration could be the enhancement of feature extraction techniques from motion vectors to improve the method's precision and adaptability to a wider range of violent activities. Additionally, investigating the scalability of the proposed method to handle real-time video streams in large-scale surveillance systems could offer valuable insights for practical implementations. Furthermore, exploring the integration of other machine learning algorithms or fusion strategies with SVM for enhanced classification accuracy may provide avenues for further optimization and generalization of the method.

XIV Fight Recognition in Video Using Hough Forests and 2D Convolutional Neural Network [36].

The study presented in this paper significantly contributes to the field of video analysis by introducing a method that achieves notable results on datasets with various frame sizes. By conducting a comprehensive comparison with handcrafted and deep learning methods from the literature, the proposed approach stands out, particularly on the Movies and Behave datasets. Notably, the use of a 2D Convolutional Neural Network trained from scratch with specific parameters demonstrates the method's effectiveness in comparison to existing techniques.

In the literature, the work done includes evaluating handcrafted feature methods such as Violent Flows (ViF), LMP, and MoIWLD, along with the application of

different classifiers like SVM, Adaboost, and Random Forests. Deep feature learning methods like 3D Convolutional Neural Networks (3D-CNN) and C3D are also considered, each showcasing varying levels of success. Additionally, the incorporation of the BRISK descriptor with Hough Forest classifier provides insights into alternative approaches for video analysis.

Despite the advancements presented, some potential gaps warrant further exploration. Firstly, while the proposed method excels on the Movies and Behave datasets, more detailed insights into the method's performance on larger or more diverse datasets could enhance the generalizability of the results. Secondly, a deeper investigation into the robustness of the method across different input data characteristics and scenarios would provide a more comprehensive understanding of its applicability. Future research could focus on optimizing the method's parameters to potentially enhance its efficiency and effectiveness in real-world applications.

The summary of reviewed existing literature is shown in Table 4 below.

Table 4 Summary of existing literature on violence behavior detection

S/No.	Author(s)	Title	Methodology Proposed	Strength(s)	Gap(s)
1	Nova D, Ferreira A, Cortez P	A Machine Learning Approach to Detect Violent Behaviour from Video	The study utilizes Python, OpenPose, and other libraries like Numpy and scikit-learn to experiment with SVM for violent behavior detection via feature extraction from video frames	The methodology offers high true positive and negative rates, showcasing robust classification accuracy using the SVM mode	Limited information on the scalability and real-world applicability of the model, such as in different or more complex environments
2	Omarov B, Narynov S, Zhumanov Z, Gumar A, Khassanova M	State-of-the-art Violence Detection Techniques in Video Surveillance Security Systems: A Systematic Review	A systematic literature review integrating both qualitative and quantitative analysis, utilizing five digital libraries and key conferences, focused on violence detection techniques in video surveillance	Comprehensive review scope, inclusion of high-value conferences, and integration of advanced machine learning and deep learning methods	Exclusion of non-English and non-journal papers may overlook relevant studies; focus on video surveillance limits broader application insights
3	Al-Dhamari A, Sudirman	Transfer Deep Learning Along with Binary	Researchers introduced a framework for anomaly	The methodology effectively identifies unusual behaviors	Limited scalability and potential high false positives in

	R, Mahmood N	Support Vector Machine for Abnormal Behavior Detection	detection in dense scenes using dynamic scene modeling and anomaly localization techniques	in crowded scenes, enhancing security and monitoring systems	extremely diverse crowd behaviors need addressing
4	Zhou P, Ding Q, Luo H, Hou X	Violent Interaction Detection in Video Based on Deep Learning	A ConvNet named FightNet was developed to detect violent interactions by modeling long-term temporal structures, pretrained on the UCF101 dataset and tested on various public datasets like Hockey and Movies	FightNet achieved 100% accuracy on the Movies dataset and showed high adaptability across different datasets with reasonable computational costs	Despite its high performance, the robustness across all tested datasets was not uniform, suggesting a need for further enhancement in feature detection under varied conditions
5	Hassner T, Itcher Y, Kliper-Gross O	Violent Flows: Real-Time Detection of Violent Crowd Behavior	The methodology involves using ViF descriptors with SVMs for violence detection in videos from crowd environments under unconstrained conditions	ViF outperforms other methods with significant margin, especially effective for videos displaying crowd behaviors	Limited by the type and variety of crowd videos available, which may affect generalizability of findings

6	Lin J, Wang W	Weakly-Supervised Violence Detection in Movies with Audi and Video Based Co-training	Audio violence is detected by segmenting audio into clips, extracting low-level features, clustering these into an audio vocabulary, and using probabilistic latent semantic analysis (pLSA) with Expectation Maximization (EM) for classification	Leverages proven speech recognition features and robust clustering via k-means algorithm, enhancing precise categorization in audio analysis	The scalability in varied real-world scenarios and sensitivity to diverse audio contexts are not deeply explored
7	Datta A, Sha M, Da N, Lobo V	Person-on-Person Violence Detection in Video Data	Analyzes scenes for violence using motion history, object handling, and orientation data. Checks skin presence and adjusts for non-visible object elements	Effective in detecting intricate violent and non-violent activities using silhouette-based orientation and object interaction analysis	Challenges remain in identifying objects as potential weapons and in distinguishing smaller objects during handovers
8	Nam J, Alghoniemy M, Tewfik A	Audio-Visual Content-based Violent Scene Characterization	The technique integrates audio and visual tracks to detect violence, using dynamic	Enhances accuracy by combining both audio and visual cues, covering a broader	Potential for false positives due to mixed sounds; differentiation from non-violent

			activity signatures and sound effects analysis	aspect of violent scenes characterization	high-action scenes remains challenging
9	Souza F, Pedrini H	Detection of Violent Events in Video Sequences based on Census Transform Histogram	Introduced a video analysis method using CENTRIST features for detecting violence, involving preprocessing, feature extraction, and classification with machine learning algorithms	Effective on two datasets, competitive accuracy with simple yet robust features, enhanced with preprocessing techniques	Did not surpass top existing methods, limited novelty in the conceptual approach
10	Chen M, Hauptmann A	Recognizing Human Actions in Surveillance Videos	The MoSIFT algorithm captures local spatio-temporal features through interest points detection and encodes local appearance and motion using histograms of gradients and optical flow, integrated with a bigram model for structural correlation	MoSIFT effectively enhances human action recognition in surveillance videos, showing a significant improvement over traditional models with a robust feature descriptor and a bigram approach for global structure analysis	While MoSIFT advances action detection, it struggles with actions that have subtle motions (e.g., CellToEar), and its performance on certain complex actions remains uncertain due to the lack of detailed annotations

11	Gao Y, Liu H, Sun X, Wang C, Liu Y	Violence Detection using Oriented Violent Flows	The methodology employs a Histogram of Oriented Optical Flow (HOOF) designed for violence detection, which concatenates histograms from magnitude and angle measurements of flow vectors into a vector H, followed by binary indicators based on magnitude changes	The approach integrates traditional machine learning algorithms, SVM and AdaBoost, for enhanced classification performance. This combination effectively selects features and improves classifier training, leveraging the strengths of both methodologies	The methodology lacks normalization and specific details on the counting of orientations may limit its adaptability to different scenarios. Additionally, there is no mention of validation across diverse video datasets which might affect generalized performance
12	Ben Mabrouk A, Zagrouba E	Abnormal Behavior Recognition for Intelligent Video Surveillance Systems: A Review	The methodology involves detecting the interest region using low-level features, followed by describing the region with generated primitives to provide semantic information about human actions	The approach effectively combines low-level feature detection with semantic analysis, enhancing understanding and classification of human actions	The methodology faces challenges in feature robustness against transformations, impacting the accurate behavior representation of the interest object

13	Xie J, Yan W, Mu C, Liu T, Li P, Yan S	Recognizing Violent Activity Without Decoding Video Streams	The proposed methodology involves detecting violent behavior in video streams by analyzing motion vectors extracted directly from these streams, streamlining processing for real-time applications	This approach uses direct video stream analysis, ensuring faster processing suitable for real-time systems	The extraction relies on motion vectors which might not capture all aspects of violent behaviors, potentially affecting accuracy
14	Serrano I, Deniz O, Espinosa-Aranda J, Bueno G	Fight Recognition in Video Using Hough Forests and 2D Convolutional Neural Network	A 2D Convolutional Neural Network (CNN) utilizes handcrafted images, highlighting motion details from video frames to detect fight sequences effectively	The method leverages simplified CNN architecture for efficient processing, achieving high accuracy with optimized frame resolution and parameters	Relies heavily on predefined handcrafted features, potentially limiting adaptability to different, unstructured video content. Lacks real-time processing capabilities.

5 SELECTED DEEP LEARNING MODELS FOR VISUAL DATA PROCESSING

In this chapter, I explore four (4) prominent deep learning models used for the experiments in this work: ResNet50, VGG-16, DenseNet-121 and Inception-v3, explaining their architectural differences, contributions and practical applications. The key concept applied here is known as Transfer learning. This allows for pretrained models to be enhanced and fine-tuned on a domain-specific dataset for purpose of achieving optimal performance.

The field of deep learning has undergone a lot of research and new areas of application have come up as a result [37]. Among the different types of deep learning architectures, CNNs have been instrumental in tasks related to image classification, object detection, and semantic segmentation. Most common CNN models in use today are best known for their performance and effectivity.

CNNs play a significant role in image processing which is central to recognising patterns in images and videos alike. CNNs mainly excel in learning hierarchical representations of visual data through convolutional layers, pooling operations and non-linear activations [37]. Various deep learning systems have surpassed human-level performance in image recognition benchmarks, thereby pushing the boundaries of what is possible today with advancements in medical field, self-driving cars, robotics as well as many other intelligent systems [17].

ResNet architecture introduced by He et al. in 2015 addresses the challenges of training deep networks by introducing skip connections, allowing for gradients to be propagated more effectively through the network [38]. This approach makes it possible to train exceptionally deep networks, removing the limitations faced by older models. The vanishing gradient problem is also minimised allowing for convergence and enabling the exploration of deeper architectures.

The VGG-16 model proposed by Simonyan and Zisserman in 2014 is a system of multiple convolutional layers interspersed with max-pooling layers, resulting in several fully connected layers for classification [39]. It is quite simple when compared to other contemporary architectures, but demonstrates a good performance on various image recognition tasks. The ability to learn discriminative features from visual data is something that sets it apart from other models [39].

DenseNet architecture is based on a connectivity pattern known as dense connections. The idea behind dense connections is that each layer of the network receives input from all preceding layers within a dense block [40]. This densely connected architecture makes it possible to reuse features, while facilitating gradient flow and enhancing the model's compactness. DenseNet has a distinctive connectivity pattern that solves the vanishing gradient problem, while maintaining feature propagation and recording a good performance on image classification benchmarks [40].

In 2015, Inception-v3 model architecture was proposed by Szegedy et al. This model makes use of inception modules which uses multiple parallel convolutional pathways with different kernel sizes to capture spatial hierarchies of features at different scales [41]. This multi-scale processing capability allows Inception-v3 to effectively capture both local and global features. Inception-v3 performs well in feature extraction and can discern very intricate patterns from images [41].

In the following chapters, each model is described in details, showing relevant architectural components, training methodologies and general performance indicators.

5.1 RESIDUAL NETWORK (RESNET)

ResNet introduces a novel architecture element called residual block, which changed the way in which deep networks are constructed [38]. While traditional CNNs relied on stacking multiple layers to learn, ResNet uses the concept of residual blocks in order to solve the problem of vanishing gradient commonly associated with models that use multiple layers. ResNet introduces a concept known as 'skip connections' which means that activations of layers can be connected to further layers by skipping some layers in between.

One of the major breakthroughs in the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was the debut of ResNet. ResNet achieved groundbreaking success by tackling the difficulties of training extremely deep neural networks. It surpassed other architectures by a significant margin, winning the image classification task in ILSVRC 2015 with a top 5 error rate of 3.57% and delivering remarkable performance [12].

[12] https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9

Researchers have long discovered that more layers in a CNN implied a greater flexibility to adapt to various datasets due to the expanded parameter space. However, newer studies have shown that beyond a certain depth, performance begins to decline in models with several layers [38]. This limitation is prominent in VGG models which loses its generalization ability for deeper configurations.

As shown in Figure 13, the residual block acts as a key component of the ResNet architecture. It takes the input to the block and adds it to the output of the block creating a residual connection.

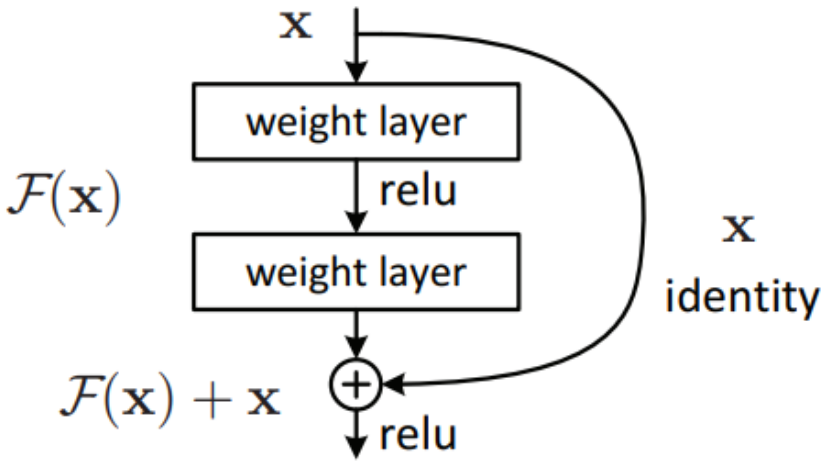

Figure 13 Residual block in a ResNet model architecture [38]

Residual blocks usually are made up of convolutional layers with batch normalization and ReLU activations, followed by a skip connection that adds the input to the output of the second convolutional layer. This design allows the network to retain information from earlier layers without being diluted or transformed excessively by subsequent layers [38].

From the equation (6) below, relationship between the input and output shows that NNs are good function approximators. They are able to solve functions where the output of the function becomes the input itself as shown in equation (6).

$$f(x) = x \tag{6}$$

Having established this logic, if the input to the first layer is bypassed to be the output of the last layer, we should see the model predict the previous function it learnt before the input was passed to it.

This relationship is expressed in equation (7) below.

$$f(x) + x = h(x) \tag{7}$$

ResNet-18 and ResNet-34 represent the earlier iterations of ResNet architectures, characterized by their relatively shallow depths compared to later variants. ResNet18 comprises 18 layers, while ResNet34 extends this to 34 layers. These architectures primarily consist of basic residual blocks, each containing two convolutional layers [38]. They are commonly employed in scenarios where computational resources are limited or for tasks with moderate complexity, such as image classification on small to medium-sized datasets.

ResNet50 introduces a significant departure from its predecessors by leveraging bottleneck residual blocks. With a depth of 50 layers, ResNet50 strikes a balance between model complexity and computational efficiency [38]. These bottleneck blocks utilize 1x1 convolutional layers to reduce and then restore the dimensions of feature maps, effectively reducing computational complexity while maintaining expressive power. As a result, ResNet50 finds widespread use across various computer vision tasks, including image classification, object detection, and semantic segmentation [38].

Building upon the success of ResNet50, ResNet101 further extends the depth of the architecture to 101 layers. By incorporating additional residual blocks, ResNet101 enhances the representational power of the network, making it suitable for more challenging tasks or scenarios where higher accuracy is required [38]. This deeper architecture enables the model to capture increasingly complex patterns and features from the input data, leading to improved performance on a wide range of tasks.

At the pinnacle of the ResNet hierarchy lies ResNet152, the deepest variant among its counterparts. With a staggering depth of 152 layers, ResNet152 offers the highest level of representational capacity within the ResNet family [38]. This architecture is particularly well-suited for tasks demanding very high accuracy, such as fine-grained image classification or medical image analysis, where intricate details and subtle distinctions are crucial for decision-making.

Table 3 below summarises the evolution of ResNet architectures – from ResNet18 to ResNet152, showing the progression in the number of parameters supported.

Table 5 ResNet architectures for ImageNet[13]

Number of Layers	Number of Parameters
ResNet 18	11.174 M
ResNet 34	21.282 M
ResNet 50	23.521 M
ResNet 101	42.513 M
ResNet 152	58.157 M

ResNet architectures offer several benefits with the most significant being the improved training process for deep networks which leads to faster convergence and a simplified model overall. ResNet is able to achieve this without any major performance hits as observed in plain networks.

The incorporation of skip connections allows the model to easily identify identity functions as illustrated in equation (7). This same logic is applied by the model to better generalise data it had not been exposed to as the network is able to skip information not required for decision making. All these highlighted features are what makes ResNet desirable for deep learning tasks such as the experiment conducted in this work.

5.2 VISUAL GEOMETRY GROUP (VGG-16)

VGG-16 is a CNN model widely used in image classification tasks. This model is known for its simplicity and efficient performance in image classification and recognition. The creators of this model – Karen Simonyan and Andrew Zisserman from Visual Geometry Group, in a paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition" evaluated the networks while increasing the depth using an architecture with convolution filters of 3 x 3 [39]. The depth was set to 16-19 weight layers and this allowed up to 138 parameters to be trained.

[13] https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8

$$Y^{(l)} = \sigma(W^{(l)} * Y^{(l-1)} + b^{(l)}) \quad (8)$$

In the equation (8) above, the output of the l-th convolutional layer is represented as a mathematical expression. $Y^{(l)}$ represents the output feature map, $W^{(l)}$ denotes the filter weights, * denotes the convolution operation, $b^{(l)}$ is the bias term, and σ represents the activation function which in most cases is ReLU.

For the max-pooling operation, equation (9) and equation (10) below describe the relationship at the fully connected layers with $W^{(l)}$ denoting the weight matrix and $b^{(l)}$ representing the bias vector.

$$Y^{(l)} = MaxPool(Y^{(l-1)}) \quad (9)$$

$$Y^{(l)} = Softmax(W^{(l)} * Y^{(l-1)} + b^{(l)}) \quad (10)$$

The max-pooling operation above down-samples the feature maps by selecting the maximum value within a defined window. The fully connected layers at the end of the network perform classification using SoftMax activation function.

Training VGG-16 usually involves stochastic gradient descent (SGD) optimization with momentum, cross-entropy loss function, and weight decay regularization. At the beginning of training, preprocessing of dataset is essential before feeding into the model. This process involves tasks such as resizing of images, and normalizing the individual pixel values [39].

As soon as the dataset is prepared, VGG-16 is initialised and assigned random weights. As explained in the model paper [39], there are a number of initialisation techniques that ensure the model starts with reasonable weights which speeds up convergence during training.

During the training process, the choice of loss function is pivotal, particularly for classification tasks. Cross-entropy loss is commonly employed, measuring the disparity between the predicted probability distribution and the actual distribution of labels. This loss function guides the optimization process by quantifying the model's performance and facilitating gradient-based updates to the parameters [42].

Optimization algorithms play a fundamental role in training deep neural networks like VGG-16. Stochastic Gradient Descent (SGD) with momentum is a popular choice due to its ability to efficiently navigate the parameter space while mitigating oscillations [42]. The

momentum term accelerates convergence by incorporating past gradients, enhancing the optimization process.

Regularization techniques are essential for preventing overfitting and improving the generalization of deep learning models. Weight decay regularization, also known as L2 regularization, penalizes large weights in the model to prevent excessive complexity. Dropout, another regularization technique, randomly drops a fraction of neurons during training, forcing the network to learn more robust features.

Data augmentation is a critical aspect of training deep learning models, especially when working with limited datasets [42]. Techniques such as random cropping, flipping, and color jittering introduce variability into the training data, enabling the model to learn invariant features and improve its generalization capability.

The training loop iteratively feeds batches of training data into the model, computes the loss, and updates the model parameters using backpropagation. Throughout training, it's essential to monitor the model's performance on a separate validation set to prevent overfitting and fine-tune hyperparameters accordingly [39].

Finally, the trained model is evaluated on a held-out test set to assess its performance on unseen data and validate its generalization capability. This rigorous evaluation ensures that the model's performance is robust and reliable for real-world applications.

Leveraging pretrained models like VGG-16 significantly enhances the performance of image classification tasks, especially when the training data is sparse. VGG-16 as a pretrained model is extensively trained on diverse datasets such as ImageNet, which include millions of images across thousands of categories. With transfer learning, VGG-16 can effectively transfer the learned features to new classes, bypassing the need to learn from scratch.

VGG-16 remains one of the best vision model architectures for image classification and feature extraction. The analysis chapter of this work delves more into the results I obtained from the VGG-16 model after training it for purpose of identifying potential violent behavior.

5.3 DENSELY CONNECTED CONVOLUTIONARY NETWORKS (DENSENET)

DenseNet started as a significant deep learning model which performs very well with computer vision tasks. Introduced by Huang et al. in their paper titled “Densely Connected

Convolutional Networks" [40], DenseNet moves away from the traditional CNN by introducing the concept of dense connectivity among layers.

In terms of model architecture, DenseNet network layers receive direct inputs from all preceding layers within a dense block [40]. This is achieved by creating paths between the different layers of the network. The growth rate parameter controls the model's capacity and influences its ability to correctly identify complex patterns in visual data.

Figure 14 below shows the full DenseNet architecture with different number of dense layers for each dense block.

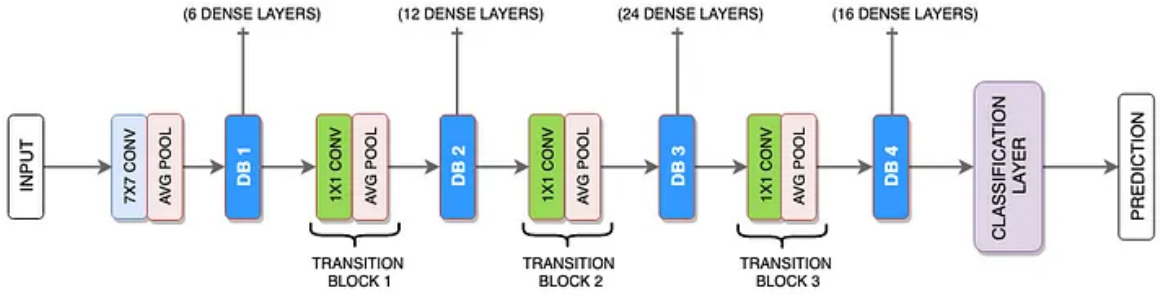

Figure 14 Full DenseNet Architecture[14]

The architectural design above makes it possible to reuse features and for gradient flow throughout the network thereby addressing the vanishing gradient problem common with some traditional CNNs. From Figure 14, the parameter-efficient design can be observed. This design style makes it possible to perform training on smaller datasets while reducing the memory requirements and computational costs involved [40].

Due to the feed-forward nature of DenseNet, every layer is able to receive feature maps from the previous layer. This process of back propagation is done through concatenation, unlike ResNet that performs this through summation [40].

The implication of having dense connections between layers as is done in DenseNet is that the model then does not require as much layers because the feature maps which do not improve the overall results are discarded.

The DenseNet architecture puts emphasis on the importance of dense connections between layers in a CNN. This brings about various improvements in gradient flow, reduced

[14] https://miro.medium.com/v2/resize:fit:720/format:webp/1*CE11_lfEz00aoOjLiw5sdw.png

parameter counts and a much better feature mapping across the network. Overall, DenseNet is computationally efficient and well suited for visual data processing [40].

5.4 INCEPTION-V3

Inception-v3 is a deep learning model developed by researchers at Google for the purpose of performing various computer vision tasks [43]. This model was developed from the earlier Inception models which were built to optimise computational resources without compromising on overall performance. The idea was to scale up networks not just by the addition of layers to make them deeper, but by ensuring that computations are as efficient as possible [43].

Inception-v3 makes use of inception modules in capturing multi-scale features correctly. These modules are made up to parallel convolutional layers with different kernel sizes [43] as represented in Figure 15 below.

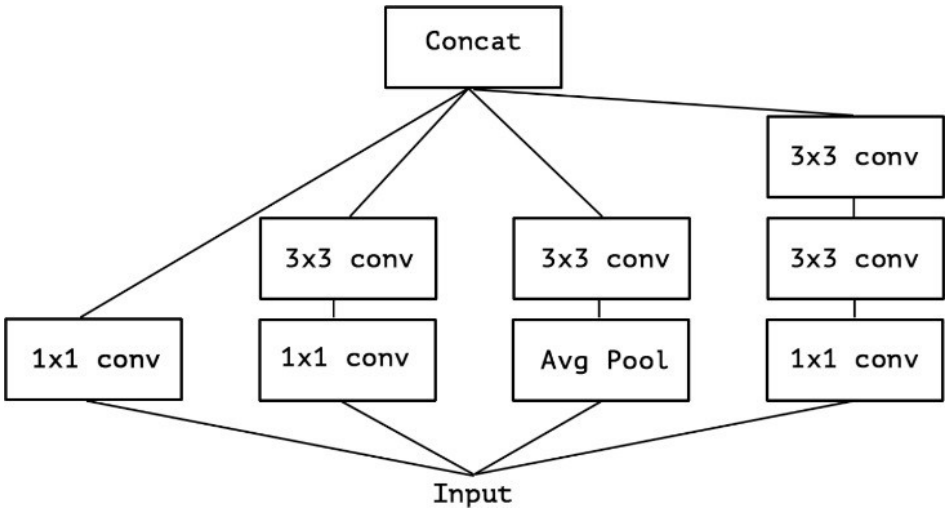

Figure 15 Schematic representation of an Inception module in Inception-v3

Inception-v3 utilises several methods in achieving more performance and these include the introduction of Label smoothing, introduction of factorization into smaller convolution layers, bottleneck layers and an auxiliary classifier in propagating label information in the network[15]

[15] https://medium.com/@sravanthigoalla917/convolutional-neural-network-in-deep-learning-4d8777a5831f

Factorization helps improve computational efficiency without necessarily increasing the number of parameters. The concept of “bottleneck” layers is introduced in Inception-v3 to refer to a case where 1 x 1 convolutions are used to reduce the dimensionality before performing more computationally intensive operations. One benefit of this approach is that it encourages feature reuse and allows for a better generalisation and learning robustness [43].

Auxiliary classifiers play an important role in Inception-v3 architecture as they help to prevent the vanishing gradient problem during the model training. There are usually loss functions at different layers to promote feature propagation and help improve the convergence speed and accuracy of the model.

II ANALYSIS

6 EXPERIMENTAL METHODOLOGY

In this work, a dataset of videos labelled as violent and non-violent have been collected from Kaggle Real Life Violence and Non-Violence dataset. This dataset contained 1,000 violence and 1000 non-violence video clips originating from publicly available online sources. The dataset features real street fight situations under varying conditions and settings, from urban to suburban environments.

Figure 16 Sample image frames extracted from violence and non-violence videos

As shown in Figure 16 above, each video clip went through preprocessing where frames were extracted and resized to 224 x 224 pixels. These pixel values were further normalised to a range between 0 and 1. Labelling was carried out to categorise the frames as either violent or non-violent. This process ensured that the dataset was suitable for the analysis that followed.

Incorporating pretrained models such as ResNet50, VGG16, DenseNet-121 and Inception-v3 was essential due to several reasons. Firstly, these models have been previously trained on the ImageNet dataset containing millions of data points, allowing the models to learn intricate features from different domains. This was instrumental in saving considerable time and computing resources that would have gone into training from scratch.

Using the concept of transfer learning, I effectively utilised knowledge that already existed in the four (4) models, but fine-tuning it to apply to the violence detection domain. Transfer learning allowed me to repurpose the four (4) selected pre-trained models to the specific task of detecting violence in videos. I was able to adapt the learned representations to suit the nuances of the violent detection problem domain, and enhancing the performance of the model in the process.

DenseNet-121, Inception-v3, ResNet50 and VGG-16 are high performant CNNs widely accepted for use in image classification tasks. The highly optimised architectural design, and computational requirements influenced my decision to use them in my experiments.

DenseNet-121 stood out in the area of feature reuse as it incorporates feature propagation throughout the network. This architectural design proved to be helpful in capturing intricate patterns that were useful in differentiating between violent and non-violent scenes.

Inception-v3 through inception modules excelled at efficiently capturing features at multiple scales. The benefit of this architectural design was evidenced in this work as it allowed the model to focus on both fine-grained details and global context within the image frames extracted from violent and non-violent scenes.

ResNet introduced a deep architecture with residual connections helping solve the vanishing gradient problem. In this work, I saw ResNet50's ability to capture hierarchical features from violent actions which contributed to the high accuracy rates obtained from the model.

VGG-16's simple architecture showed strong signs of computational efficiency while also maintaining high accuracy levels in its predictions. The stacked convolutional layers in VGG-16 facilitated the learning of discriminative features relevant to violence detection.

I developed an experiment pipeline to systematically evaluate the performance of all four (4) models selected for the purpose of detecting potential violent behavior. Data preprocessing was first carried out on the Kaggle dataset The dataset was then split into training and validation ration of 75/25.

Each selected model was trained on the preprocessed dataset using the Adam optimizer with a learning rate of 0.001 and binary cross-entropy loss function. Data augmentation techniques such as random rotation, resizing of images and normalisation of pixel values were applied to the image frames to add to the diversity of the training set. The models were trained for seven (7) epochs with a batch size of 32.

Grid search was performed to tune the hyperparameters of each model. The hyperparameters tuned were learning rate, momentum and decay. I ran the hyperparameter tuning process to retrain and evaluate the models with different combinations of hyperparameters to observe the optimal model configuration. My hyperparameter tuning process did not yield better results than the results originally obtained due to the limited number of training data. I also observed that the limited hyperparameter space, high complexity of the selected models combined with a relatively compact dataset used for transfer learning will require a more extensive tuning on larger datasets to see any improvements from hyperparameter tuning.

I performed an evaluation of the four (4) selected deep learning models using metrics such as accuracy, precision, recall and F1-score. Confusion matrices were extracted to visualise each model's performance in differentiating violent and non-violence frames in uploaded video clips. Classification reports were generated for each model and details of the results are discussed in the following chapters. DenseNet-121, Inception-v3, Resnet50 and VGG-16 recorded accuracy values of 98%, 98%, 97% and 96% respectively.

Additionally, I have developed a web application using a Python framework known as Streamlit[16] for purpose of utilising my trained models in performing real-time predictions on videos. The decision to use Streamlit framework came as a result of its robust API documentation, which significantly streamlined the learning process required to deploy my application for testing the models.

The developed application as shown in Figure 17 below provides a user-friendly interface where videos can be uploaded, with a selected trained model running on the backend to provide real-time predictions on uploaded videos. The detailed source code of this application is made available in Appendix A.

[16] https://streamlit.io/

Figure 17 Screenshot from developed violence detection system based on four selected DL models

The core of this project was built in Python using various machine learning and data science libraries. The main dependencies for this work are numpy version 1.26.4, pandas version 2.2.1, and tensorflow version 2.16.1.

Numpy was fundamental for the numerical computations performed in this work. It provided me with the functionality to use large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on my arrays more efficiently.

Pandas is used in this work for data manipulation and analysis. I utilised it for loading my dataset, cleaning and transforming the pixel values from the video frames generated.

Tensorflow as an open-source deep learning framework was instrumental in providing the low-level functionalities I used in this work to build and train my deep learning models.

A more detailed list of dependencies and software versions are provided in Appendix B.

6.1 DENSENET-121 EXPERIMENT

This chapter outlines the steps I took to perform the experiment on DenseNet-121 for the purpose of the detection of violent versus non-violent content within the video dataset. The main objective here was to evaluate the performance of DenseNet-121 under specific configurations, and determining its predictive accuracy and efficacy for the task.

6.1.1 DATA COLLECTION AND PREPROCESSING

The primary dataset used for this experiment for the Real Life Violence Situations Dataset (RLVS) available on Kaggle[17]. The dataset is a collection of videos aimed at training and evaluating machine learning models for the purpose of violence classification.

RLVS contains two thousand (2000) videos equally divided into two (2) categories. The first thousand shows real-life violence situations with emphasis on street fights under varying environmental conditions. The other thousand represents non-violent human actions such as eating, walking, dancing as well as other casual activities. The dataset forms the baseline for transfer learning performed in this work.

The decision to use this dataset was based on the fact that it contains both violent and non-violent video scenes which directly apply to my classification problem. The decision to use a balanced dataset helped me to avoid bias in my classifier.

The preprocessing procedures involved the extraction of frames from the videos, sampling the frames to ensure consistent image size and pixel values.

The result of this procedure is shown in Figure 18 below.

[17] https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset/data

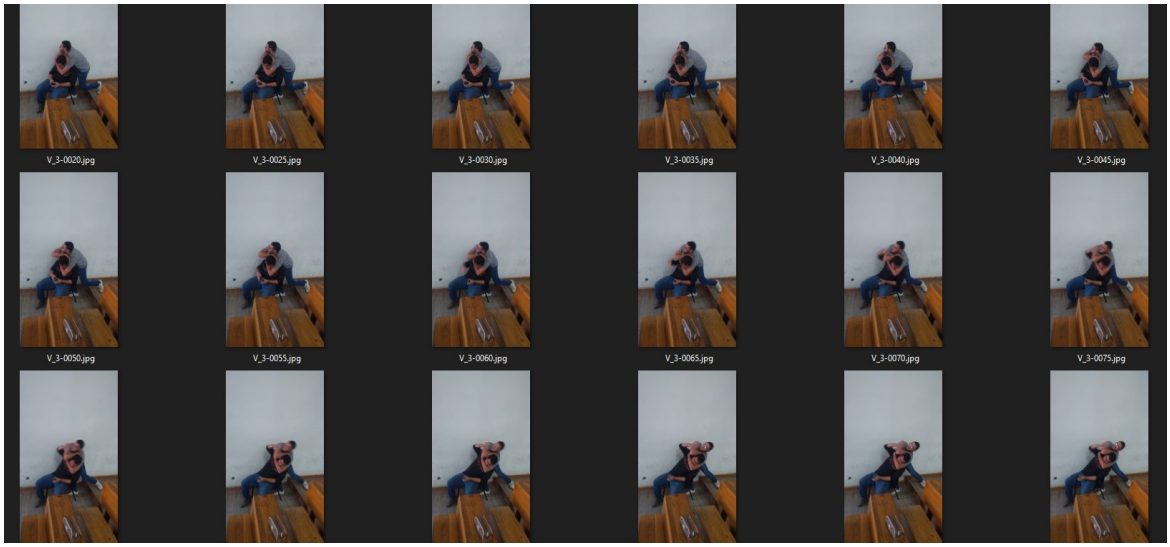

Figure 18 Extracted frames from a sample violent video

In this work, I devised a strategy to capture relevant frames at intervals, and subsequently saving them as individual images. The individual frames were then resized to a uniform dimension with their pixel values normalised to ensure input consistency. These processed images have been stored in dedicated directories for violence and non-violence categories, providing a base for the subsequent analysis that followed.

6.1.2 MODEL SELECTION AND IMPLEMENTATION

The decision to use DenseNet-121 was largely due to its efficient architecture that supports feature reuse. When compared with VGG and ResNet, I found that DenseNet was more parameter-efficient. This is because the architecture of DenseNet allows for each layer to be connected to every other layer in a feed-forward manner, maximising feature reuse and supporting feature propagation across the entire network.

DenseNet consists of densely connected blocks each with multiple convolutional layers. Using Tensorflow, I leveraged pre-built layers such as tf.keras.layers.Conv2D, tf.keras.layers.BatchNormalization, and tf.kera.layers.Dense to construct the DenseNet architecture.

6.1.3 TRAINING AND EVALUATION

The model training was conducted over 7 iterations/epochs with the hyperparameters values of Learning rate set to 0.0001, Momentum: 0.9, and Decay set to 0.0001

I chose these parameters based on their ability to stabilise and optimise the training process. In order to prevent the model from over-correcting, I decided to choose a very small learning

rate. This small step size allowed the model to train with a high level of precision accounting for small adjustments in the training set.

Momentum of 0.9 was chosen to accelerate the SGD in the relevant direction. This high momentum value allows my model to heavily factor in previous update directions leading to a smoother and faster convergence.

The accuracy and loss metrics over epochs for DenseNet-121 showed a predominantly positive trend, with validation accuracy reaching as high as 1.0000 and the stabilising around 0.9725. The validation loss decreased significantly, showing that learning and generalisation was effective over epochs. This model yielded a final accuracy value of 98%.

6.1.4 HYPERPARAMETER TUNING

I performed hyperparameter tuning using DenseNet-121 in an attempt to further improve the experimental outcome. I iterated over learning rates of 0.1, 0.01 and 0.001. Momentum values of 0.5, 0.7, 0.8 and 0.9 were also iterated over as well along with Decay options for 0.00001, 0.1, 0.001 and 0.001. The best performing result obtained from hyperparameter tuning yielded an accuracy value of 82% which is significantly lower than the 98% value obtained from the initial hyperparameters chosen.

The hyperparameter tuning did not yield any improvements mainly because the model was not highly sensitive to the hyperparameters being tuned. It appeared that the initially chosen hyperparameters that yielded the accuracy of 98% were already near-optimal for the dataset. The limited dataset quantity also reduced the overall benefits of hyperparameter tuning.

6.1.5 RESULTS AND ANALYSIS

Results from the DenseNet-121 model shows an accuracy value of 98% as shown in the classification report in Figure 19 below.

```
------------------------------------------------------------
[INFO] Classification Report:
------------------------------------------------------------
              precision    recall  f1-score   support

 NonViolence       0.97      0.98      0.98       750
    Violence       0.98      0.97      0.98       750

    accuracy                           0.98      1500
   macro avg       0.98      0.98      0.98      1500
weighted avg       0.98      0.98      0.98      1500

------------------------------------------------------------
[INFO] Confusion Matrix:
------------------------------------------------------------
[[734  16]
 [ 20 730]]
```

Figure 19 DenseNet-121 Classification report

The precision and recall are both high indicating that the model had a balanced capability in differentiating between violent and non-violent frames without bias.

The F1-score of 0.98 for both violence and non-violence frames show that there was a good balance between precision and recall, suggesting that the model is robust.

The training accuracy and loss for the model is shown in Figure 20 below.

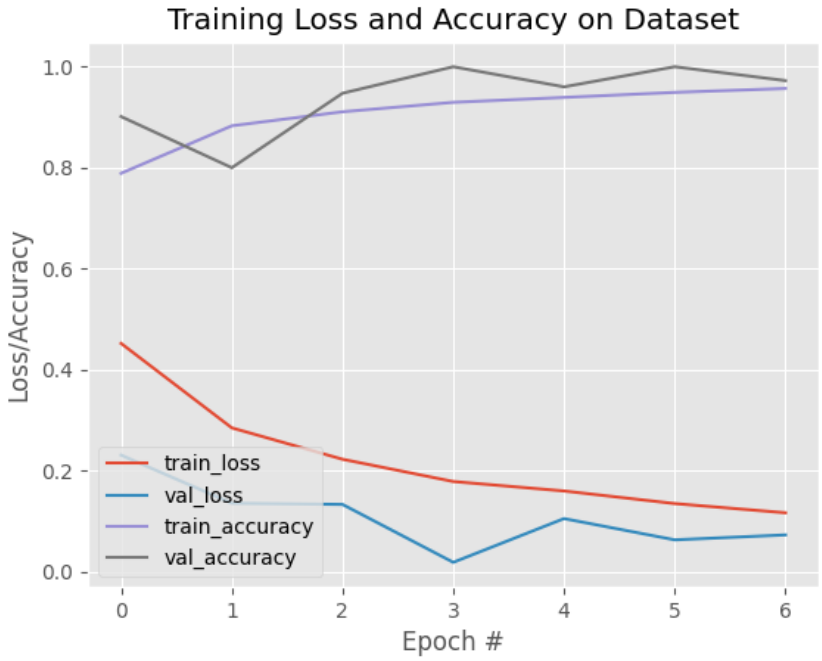

Figure 20 DenseNet-121 Plot of training loss and accuracy

The confusion matrix for DenseNet-121 is shown in Figure 21 below.

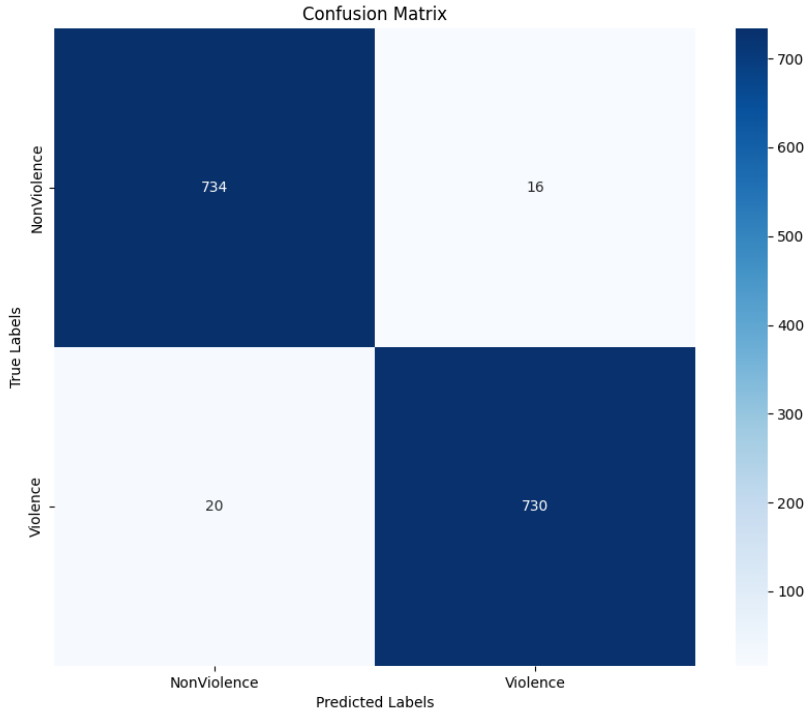

Figure 21 DenseNet-121 Plot of confusion matrix

The DenseNet-121 model correctly predicted 734 instances as positive. 16 instances were predicted as positive when they were actually negative.

The model further correctly predicted 730 instances as negative. 20 instances were wrongly predicted as negative when they were actually positive.

From these metrics, the DenseNet-121 model performs well with high accuracy, precision, recall and F1-score showing a good classification performance.

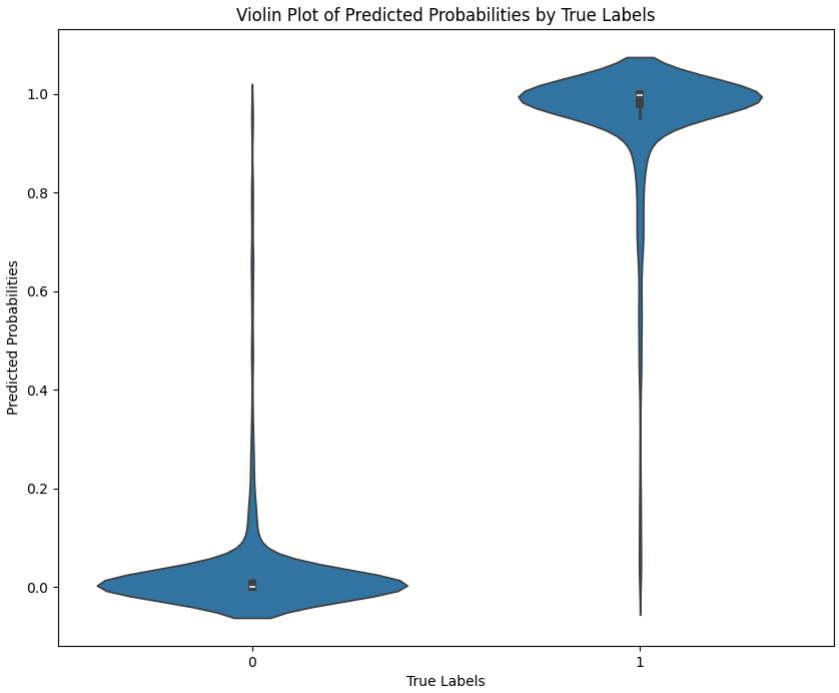

Figure 22 DenseNet-121 Violin plot

The violin plot for DenseNet-121 in Figure 22 above visualises the distribution of the predictions. As expected, the wider sections of the violin plot tend more towards 0 and 1 with not many visible outliers due to the fact that the work is a binary classification problem.

6.2 INCEPTION-V3 EXPERIMENT

In this chapter, I evaluate the performance of Inception-v3 as was done for DenseNet in the previous chapter. The information obtained helped to determine predictive accuracy and efficiency of the Inception-v3 for the specific task of predicting violent behavior.

6.2.1 DATA COLLECTION AND PREPROCESSING

The data collection and preprocessing steps for the Inception-v3 experiment follows the same procedure already described in the experiment for DenseNet-121 in chapter 6.1.1 above.

6.2.2 MODEL SELECTION AND IMPLEMENTATION

The choice to use Inception-v3 in this experiment is largely due to the balance Inception-v3 brings between computational efficiency and performance. Unlike the other selected models – VGG and ResNet, Inception-v3 uses multiple parallel convolutional pathways within each module, which allows it to perform feature selection at different scales.

The main innovation behind Inception-v3 is the integration of different convolutional layers of sizes 1 x 1, 3x 3, and 5 x 5 together with pooling layers to make feature selection more efficient. Inception-vs is able to perform factorisation on smaller convolutions by replacing single 5 x 5 convolutions into multiple 3 x 3 convolutions. This made a huge difference in the area of computational power utility as I was able to get similar performance out of this model with a reduced processing power.

In TensorFlow, I constructed the Inception-v3 architecture using TensorFlow's high level APIs. I made use of pre-built layers such as tf.keras.Conv2D, tf.keras.MaxPooling2D, and tf.keras.layers.Concatenate.

6.2.3 TRAINING AND EVALUATION

Similar to the training approach used in chapter 6.1.3, Inception-v3 was trained using similar hyperparameters. The choice of hyperparameters is also justified in chapter 6.1.3.

The accuracy and loss metrics over epochs for Inception-v3 showed a positive trend, with validation accuracy reaching 1.0000 on 2 epochs and finally stabilising around 0.9738. The validation loss decreased indicating that the model was good at learning the data. Overall, this model yielded an accuracy value of 98%.

6.2.4 HYPERPARAMETER TUNING

For the hyperparameter tuning of Inception-v3, I kept the learning rate, momentum and decay rates consistent with the values used for DenseNet-121 experiment as described in chapter 6.1.4.

I iterated over learning rates of 0.1, 0.01 and 0.001. Momentum values of 0.5, 0.7, 0.8 and 0.9 were also iterated over along with Decay options for 0.00001, 0.1, 0.001 and 0.001. The best performing result obtained from hyperparameter tuning yielded an accuracy value of 74% which is significantly lower than the 98% value obtained from the default hyperparameters.

The hyperparameters did not yield any improvements mainly because the model being pre-trained with millions of data points is not highly sensitive to the hyperparameters being tuned. The hyperparameters that yielded 98% were already near-optimal for the dataset.

6.2.5 RESULTS AND ANALYSIS

Results from the Inception-v3 model shows an accuracy value of 98% as shown in the classification report in Figure 23 below.

```
------------------------------------------------------------
[INFO] Classification Report:
------------------------------------------------------------
              precision    recall  f1-score   support

 NonViolence       0.97      0.99      0.98       750
    Violence       0.99      0.97      0.98       750

    accuracy                           0.98      1500
   macro avg       0.98      0.98      0.98      1500
weighted avg       0.98      0.98      0.98      1500

------------------------------------------------------------
[INFO] Confusion Matrix:
------------------------------------------------------------
[[744   6]
 [ 20 730]]
```

Figure 23 Inception-v3 Classification report

With a precision value of 0.99 and 0.97 in violence and non-violence respectively, the model shows a relatively balanced capability to differentiate between violent and non-violent scenes.

The F1-score of 0.98 for both violence and non-violence frames show that there was a good balance between precision and recall, suggesting that the model is robust.

The training accuracy and loss for the Inception-v3 model is shown in Figure 24 below.

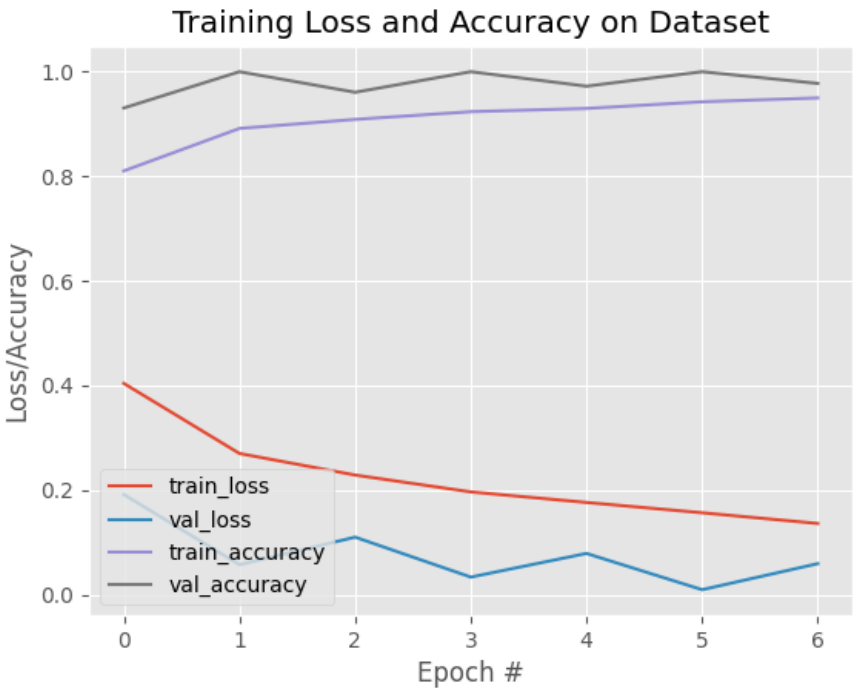

Figure 24 Inception-v3 Plot of training loss and accuracy

The confusion matrix for the Inception-v3 model is shown in Figure 25 below.

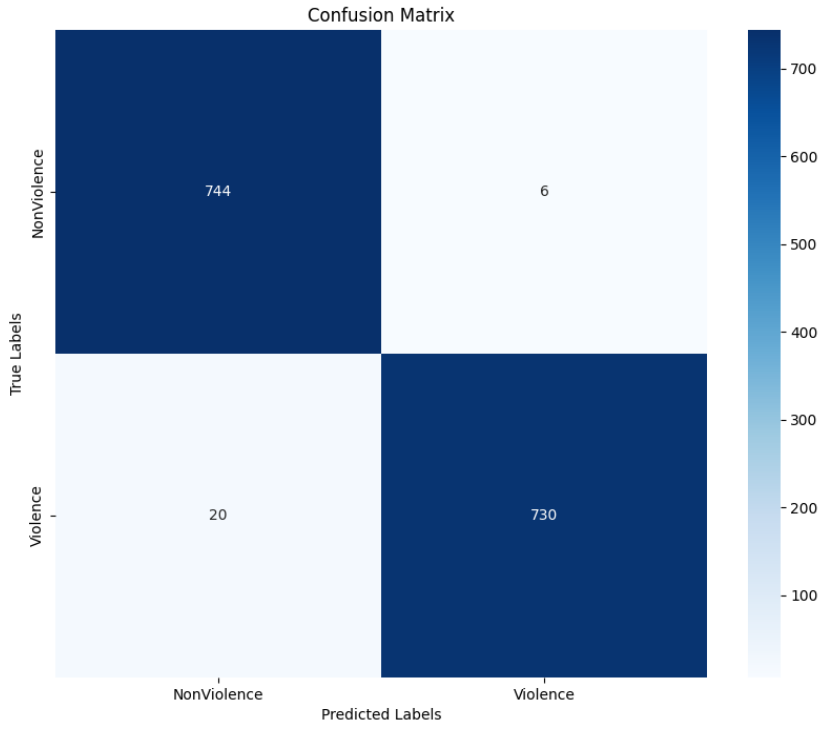

Figure 25 Inception-v3 Plot of confusion matrix

The Inception-v3 model correctly predicted 744 instances as positive. 6 instances were wrongly predicted as positive when they were actually negative.

The model further correctly predicted 730 instances as negative. 20 instances were wrongly predicted as negative when they were actually positive.

From these metrics, the Inception-v3 model performs well with high accuracy, precision, recall and F1-score showing a good classification performance just like the other selected models.

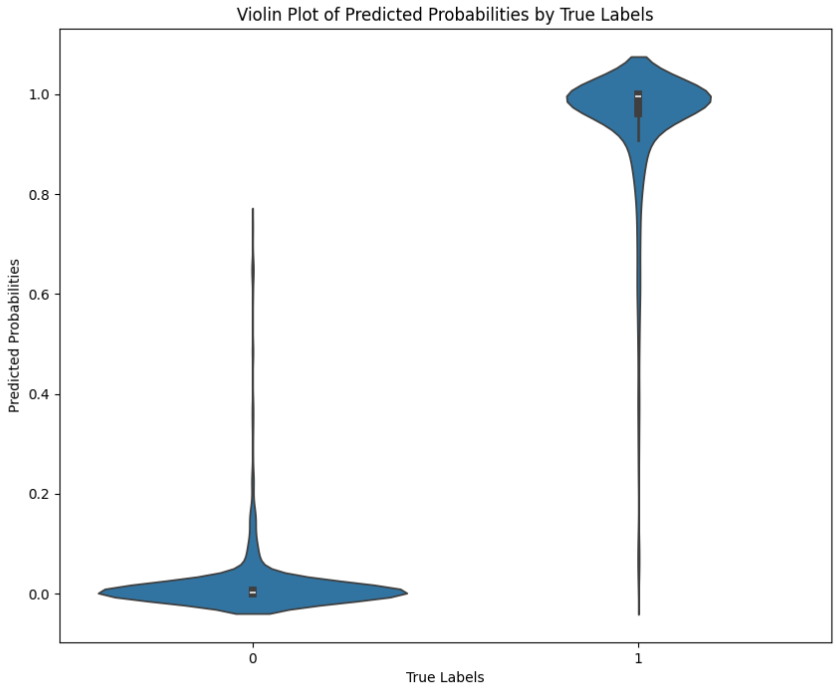

Figure 26 Inception-v3 Violin plot

The violin plot for Inception-v3 in Figure 26 above shows the distribution of the predictions. The shape of the violin plot is as expected with the wider sections tilted more towards values of 0 and 1.

6.3 RESNET50 EXPERIMENT

This chapter details the results I obtained from the experiment conducted on ResNet50 model. The results are discussed with relevant tables and figures analysed to provide insights into the model's performance.

6.3.1 DATA COLLECTION AND PREPROCESSING

The data collection and preprocessing steps for the ResNet50 experiment follows the same procedure already described in the experiment for DenseNet-121 in chapter 6.1.1 above.

6.3.2 MODEL SELECTION AND IMPLEMENTATION

I selected ResNet50 for this experiment due to the efficient nature of the model's architectural design. The ResNet architecture is made of a lot of convolutional layers which allows direct propagation of gradients throughout the network. The vanishing gradient problem observed with the VGG model is not present in ResNet.

In TensorFlow, I implemented the ResNet50 architecture using TensorFlows intuitive API to construct the residual blocks using pre-built layers such as tf.keras.layers.Conv2D and tf.keras.layers.BatchNormalization. Additionally, the tf.keras.layers.Add layer facilitates the merging of the main pathway with the shortcut connection, ensuring seamless information flow throughout the network.

6.3.3 TRAINING AND EVALUATION

In a similar training approach used in chapter 6.1.3, ResNet50 was trained using similar hyperparameters. The choice of hyperparameters was justified in the same chapter.

The accuracy and loss metrics over epochs for ResNet showed a positive trend, with validation accuracy reaching 1.0000 on 3 epochs and finally stabilising around 0.9722. The validation loss decreased indicating that the model was good at learning the data. Overall, this model yielded an accuracy value of 97% which is slightly lower than the values obtained for DenseNet-121 and Inception-v3.

6.3.4 HYPERPARAMETER TUNING

I performed hyperparameter tuning on ResNet50 model. No significant improvement was observed after the hyperparameter tuning process.

I experimented with learning rates of 0.1, 0.01 and 0.001. Momentum values of 0.5, 0.7, 0.8 and 0.9 were also iterated over along with Decay options for 0.00001, 0.1, 0.001 and 0.001. The best performing result obtained from hyperparameter tuning yielded an accuracy value of 65% which is significantly lower than the 97% value obtained from the default hyperparameters.

The hyperparameters did not yield any improvements mainly because the pre-trained model was not highly sensitive to the hyperparameters being tuned. The hyperparameters that yielded 97% were already near-optimal for the dataset. The limited dataset used in the transfer learning could have contributed to the hyperparameter tuning failing to discover a better configuration.

6.3.5 RESULTS AND ANALYSIS

Results from the ResNet50 experiment shows an accuracy of 97% as shown in the classification report in Figure 27 below.

```
------------------------------------------------------------
[INFO] Classification Report:
------------------------------------------------------------
              precision    recall  f1-score   support

 NonViolence       0.96      0.99      0.97       750
    Violence       0.99      0.95      0.97       750

    accuracy                           0.97      1500
   macro avg       0.97      0.97      0.97      1500
weighted avg       0.97      0.97      0.97      1500

------------------------------------------------------------
[INFO] Confusion Matrix:
------------------------------------------------------------
[[740  10]
 [ 34 716]]
```

Figure 27 ResNet50 Classification report

With a precision value of 0.96 for non-violence is relatively lower than the value of 0.99 obtained for violence scenes. F1-score of 0.97 for both violence and non-violence frames show that there is a relatively good balance between precision and recall. The ResNet50 model can be considered robust as a result.

The training accuracy and loss for the ResNet50 model is shown in Figure 28 below.

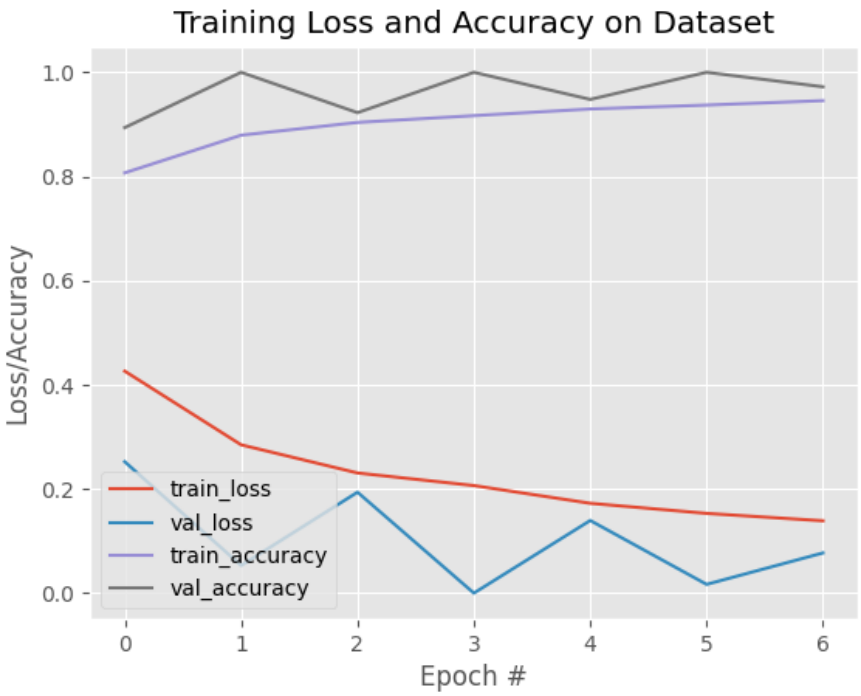

Figure 28 ResNet50 Plot of training loss and accuracy

The confusion matrix for the ResNet50 model is shown in Figure 29 below.

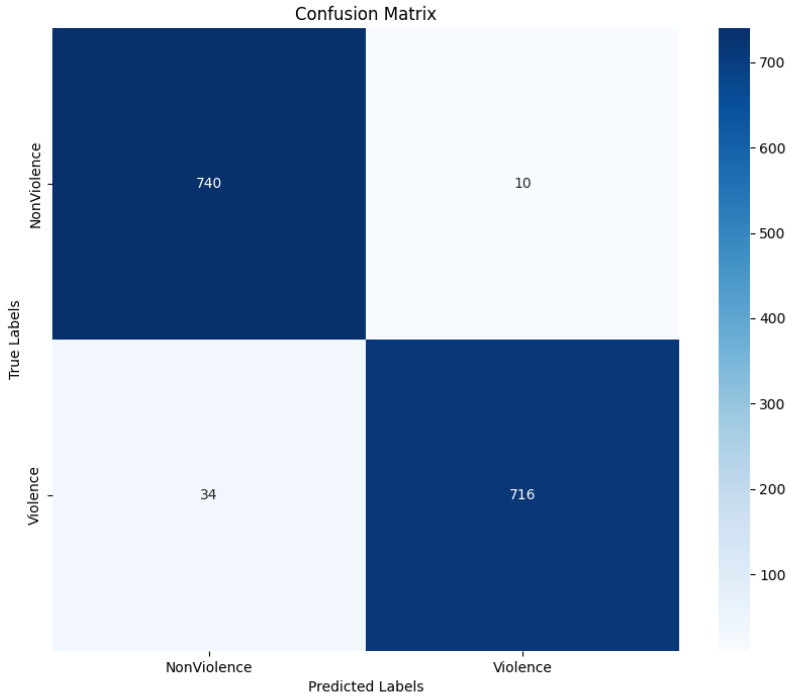

Figure 29 ResNet50 Plot of confusion matrix

The ResNet50 model correctly predicted 740 instances as positive. 10 instances were wrongly predicted as positive when they were actually negative.

The model further correctly predicted 716 instances as negative. 34 instances were wrongly predicted as negative when they were actually positive.

From these metrics, the ResNet50 model, though less accurate than DenseNet-121 and Inception-v3 in this experiment performs relatively well with high accuracy, precision, recall and F1-score showing a good classification performance.

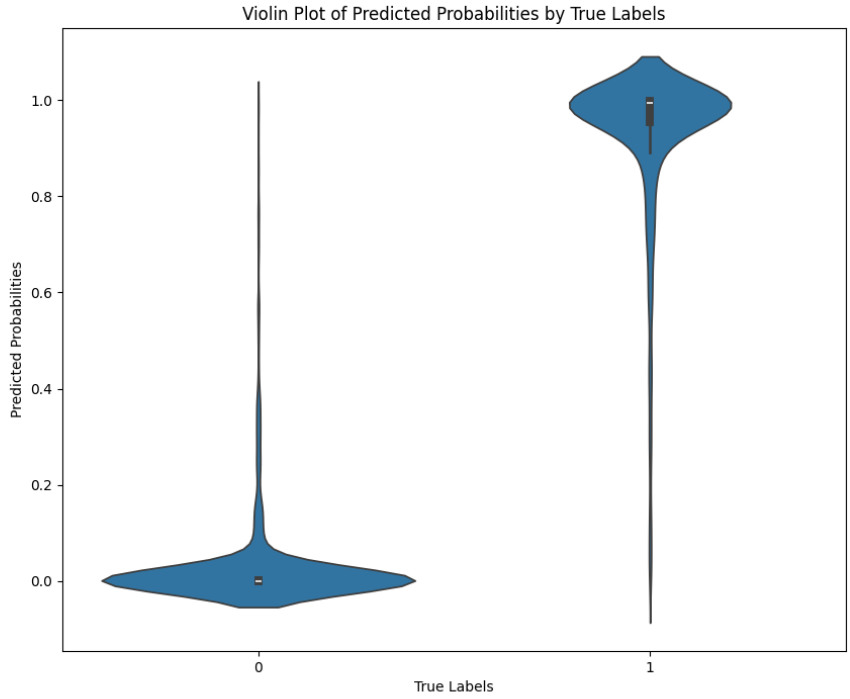

Figure 30 ResNet50 Violin plot

The violin plot for ResNet50 model in Figure 30 above has the wider sections of the violin plot spread more across values of 0 and 1. This is expected as this work deals with a binary classification problem.

6.4 VGG-16 EXPERIMENT

This chapter details the results from the experiment on VGG-16 model. The results obtained provide insights into the model performance on the specific task of predicting violence.

6.4.1 DATA COLLECTION AND PREPROCESSING

The data collection and preprocessing steps for the VGG-16 experiment follows the same procedure already described in the experiment for DenseNet-121 in chapter 6.1.1 above.

6.4.2 MODEL SELECTION AND IMPLEMENTATION

The selection of VGG-16 for this experiment was mainly influenced by the model's simplicity and ease of implementation. VGG-16 has a straight-forward architecture with multiple convolutional layers stacked on top of each other.

This simplicity in the architecture also came at the cost of the vanishing gradient problem which was observed in this model. This model performed slightly worse than the other three (3) models considered in this work with an overall accuracy of 96%.

In TensorFlow, I constructed the VGG-16 architecture by utilizing the tf.keras.layers.Conv2D and tf.keras.layers.MaxPooling2D layers to sequentially stack the convolutional and pooling layers.

6.4.3 TRAINING AND EVALUATION

The training approach used for VGG-16 is similar to the techniques already described in chapter 6.1.3 for the DenseNet-121 model architecture. The choices of hyperparameters were also justified in chapter 6.1.3.

The accuracy and loss metrics over epochs for VGG-16 again showed a positive trend like previous selected models, with validation accuracy reaching 1.0000 on 3 epochs and finally stabilising around 0.9722. The validation loss decreased indicating that the model was good at learning the data. Overall, this model yielded an accuracy value of 96%, this being the worst performing model of all the models considered in this experiment.

6.4.4 HYPERPARAMETER TUNING

I performed hyperparameter tuning on this model in an attempt to further improve the model performance. As observed with previous selected models, hyperparameter tuning of this model did not yield any improvements.

I iterated over learning rates of 0.1, 0.01 and 0.001. Momentum values of 0.5, 0.7, 0.8 and 0.9 were also iterated over along with Decay options for 0.00001, 0.1, 0.001 and 0.001. The

best performing result obtained from hyperparameter tuning yielded a record low accuracy value of 56% which is significantly lower than the 96% value obtained from the default hyperparameters.

It appeared that the pre-trained VGG-16 model was not highly sensitive to the hyperparameters being tuned. The hyperparameters that yielded 96% were already near-optimal for the dataset, and this explains why there was no improvement after the hyperparameter tuning process.

6.4.5 RESULTS AND ANALYSIS

Results from the VGG-16 model shows an accuracy value of 96% as shown in the classification report in Figure 31 below.

```
------------------------------------------------------------
[INFO] Classification Report:
------------------------------------------------------------
              precision    recall  f1-score   support

 NonViolence       0.96      0.96      0.96       750
    Violence       0.96      0.96      0.96       750

    accuracy                           0.96      1500
   macro avg       0.96      0.96      0.96      1500
weighted avg       0.96      0.96      0.96      1500

------------------------------------------------------------
[INFO] Confusion Matrix:
------------------------------------------------------------
[[723  27]
 [ 30 720]]
```

Figure 31 VGG-16 Classification report

With a value of 0.96, the precision values are high and consistent across violence and non-violence scenes. This demonstrates that the model is able to correctly differentiate between violence and non-violence. This result is also consistent with values recorded in previous selected models used in this experiment.

F1-score for both violence and non-violence have both values set at 0.96 indicating a very good balance between precision and recall. The VGG-16 can be considered a robust model as a result.

The graph for training accuracy and loss for the VGG-16 model is as shown in Figure 32 below.

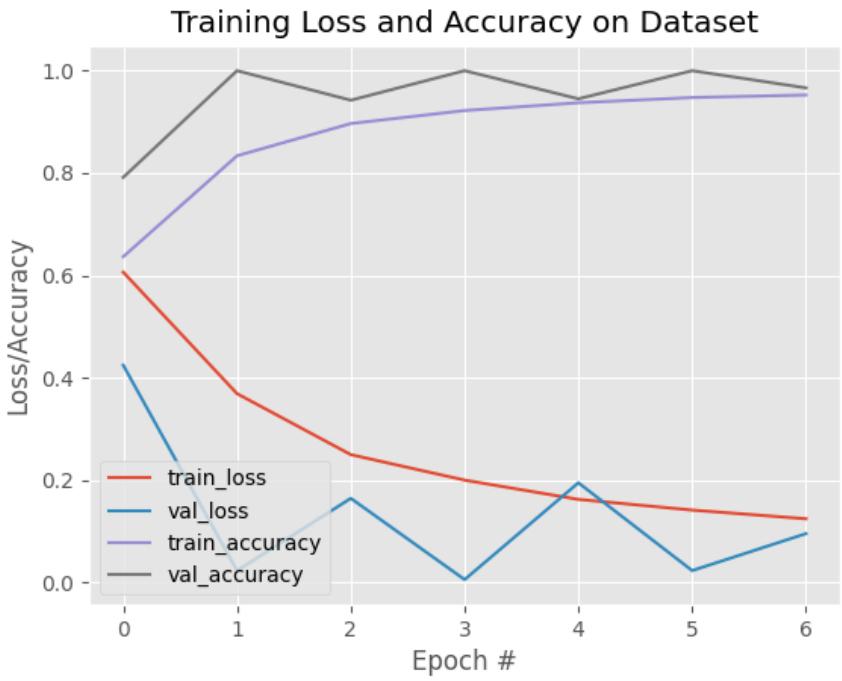

Figure 32 VGG-16 Plot of training loss and accuracy

The confusion matrix for the VGG-16 model is shown in Figure 33 below.

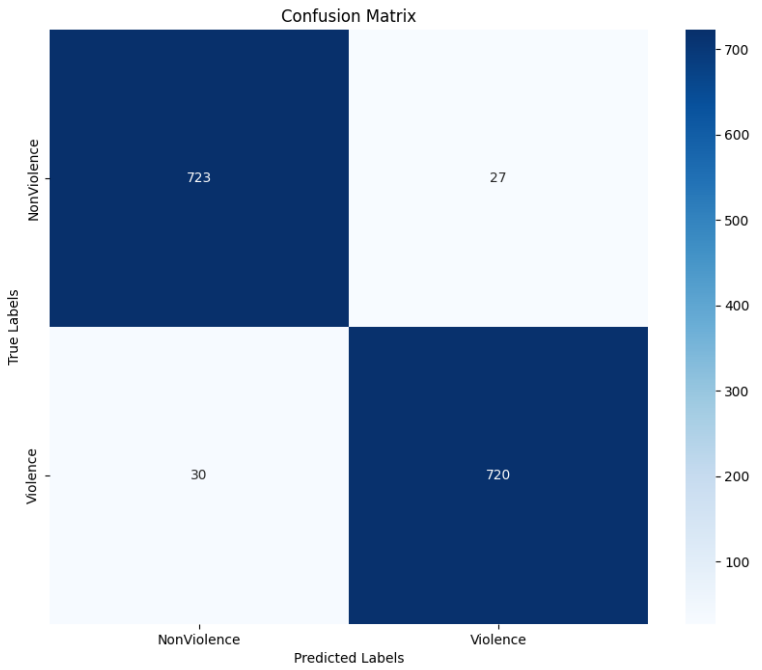

Figure 33 VGG-16 Plot of confusion matrix

The VGG-16 model correctly predicted 723 instances as positive. 27 instances were wrongly predicted as positive when they were actually negative.

The model further correctly predicted 720 instances as negative. 30 instances were wrongly predicted as negative when they were actually positive.

From these metrics, the VGG-16 model performs relatively well with high accuracy, precision, recall and F1-score of 96% showing a good classification performance.

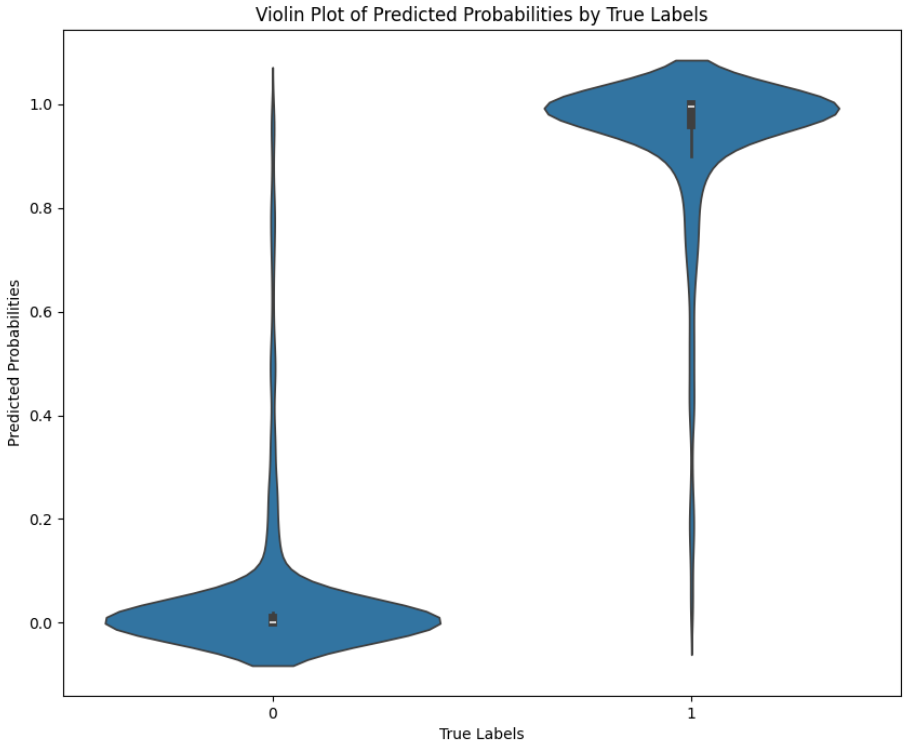

Figure 34 VGG-16 Violin plot

The violin plot for the VGG-16 model as shown in Figure 34 above is consistent with what was seen with previous selected models in this experiment. The violin plot tends more towards 0 and 1 and appear more spread out at those points due to the binary classification task in this experiment.

7 SUMMARY RESULTS AND DISCUSSION

In this chapter, a summary of the results obtained in the earlier chapters for each model is put together with comparisons made to determine how they performed in relation to one another.

7.1.1 COMPARATIVE ANALYSIS OF ACCURACY, CONFUSION MATRICES, PRECISION, RECALL, AND F1-SCORE

In terms of model accuracy, DenseNet-121 and Inception-v3 models performed best across all metrics as summarised in Table 6 below.

Table 6 Comparative analysis of accuracy, precision, recall and F1-score

Model	Accuracy	Confusion Matrix	Precision	Recall	F1-score
DensetNet-121	0.98	[[734 16] [20 730]]	0.98	0.98	0.98
Inception-v3	0.98	[[744 6] [20 730]]	0.98	0.98	0.98
ResNet50	0.97	[[740 10] [34 716]]	0.97	0.97	0.97
VGG-16	0.96	[[723 27] [30 720]]	0.96	0.96	0.96

ResNet50 followed closely with slightly lower scores in the area of accuracy, precision, recall and f1-scores. VGG-16 recorded the least values in accuracy, precision, recall and f1-scores.

The confusion matrix revealed an interesting pattern. DenseNet-121 and Inception-v3 had very identical confusion matrices showing a consistent performance in correctly classifying data points. ResNet50 also performs well, but with a slightly higher number of false negatives when compared to DenseNet-121 and Inception-v3. VGG-16 recorded the highest number of false positives and false negatives among the models considered in this experiment.

7.2 INTERPRETATION OF RESULTS

This chapter provides the analysis of the performance of the selected models – DenseNet-121, Inception-v3, ResNet50, and VGG-16. The strengths and weaknesses observed in each model is discussed to provide insights into model's overall applicability.

7.2.1 STRENGTHS AND WEAKNESSES OF SELECTED MODELS

DenseNet-121 and Inception-v3 had the best performance based on the performance metrics in Table 6. The identical confusion matrices obtained for both models further show their ability to make accurate predication on data points. Both models were the best at maintaining high accuracy and precision while reducing the number of false positives and false negatives when making predictions.

DenseNet-121 and Inception-v3 despite the impressive performance in this experiment may not be well suited for use on complex datasets where computational resources are limited. These two (2) models have very deep and complex architectural structures that demands significant computational resources.

ResNet50 showed a strong performance as well, although with slightly lower values than DenseNet-121 and Inception-v3. It evidently maintains high levels of accuracy, precision, recall and f1-score proving to be suitable for the task of violent behavior detection. ResNet50 when compared to DenseNet-121 and Inception-v3 has a shallower model architecture which makes it more computationally efficient.

ResNet50 demonstrated some weaknesses in the area of predicting a high number of false negatives when put side-by-side with DenseNet-121 and Inception-v3. This slight weakness can however be corrected by further fine-tuning to improve the model's sensitivity to the data points.

VGG-16 also demonstrated decent performance across most metric. Considering the fact that the VGG-16 model has the simplest architecture when compared with other models in this experiment, the performance is impressive with a relatively high accuracy, precision, recall and f1-scores.

VGG-16 likely suffered slightly from the vanishing gradient problem as it falls short when compared to DenseNet-121, Inception-v3, and even ResNet50. It recorded the highest number of false positives and false negatives suggesting that the model might require further hyperparameter tuning with a different combination of hyperparameters.

7.2.2 INSIGHTS INTO MODEL PERFORMANCE

The comparative analysis done presents a lot insights into the performance of selected models.

DenseNet-121 and Inception-v3 emerged as the best performing models for the violent behavior detection task in this experiment. Both models had very consistent performance across the different matrices presented in Table 6 in the earlier chapter.

ResNet50 performance is not very far off when compared with the leading models in this experiment. The shallow architecture of ResNet50 provided computational advantage which was missing from the top performing models.

VGG-16 given the simple architecture performed well, but with significantly higher number of false positives and false negatives than other models considered in this experiment.

8 CONCLUSION

Within the thesis, four (4) deep learning models; DenseNet-121, Inception-v3, RestNet50 and VGG-16 were implemented for purpose of detecting potential violent behavior using transfer learning principles.

A comprehensive review of existing literature in the field of human violence detection using deep learning techniques was conducted. The models evaluated in this thesis were selected based on suitability for this work. Experimental results for different model configurations were presented.

A comprehensive comparative analysis of the four (4) selected CNN models was presented along with an evaluation of their performance using metrics such as accuracy, precision, recall and f1-score.

8.1 SUMMARY OF FINDINGS

The analysis showed that DenseNet-121 and Inception-v3 were the most suited for violence behavior detection task in this work. Both models were the most reliable as shown in their identical confusion matrices. ResNet50 and VGG-16 performed well also, but with more noticeable weaknesses.

8.2 CONTRIBUTIONS AND IMPLICATIONS

This work serves as a contribution to the existing literature in the field of human violent behavior detection. The discoveries made in this work will be particularly helpful for researchers in the area of computer vision and robotics. The analysis of the performance of the models offers great insights into the applicability of the models and will help researchers make informed decisions when choosing models for their specific applications.

8.3 FUTURE DIRECTIONS FOR RESEARCH

While this work offers insights into the performance of the four (4) selected CNN models, there are still key areas future research works can explore.

Future research can be targeted at exploring how to generalise the findings of this work across larger datasets and adapting it to wider domains to better understand how the models perform under different conditions. More optimization and hyperparameter tuning can be done with different configurations with a goal to further to improve model performance.

BIBLIOGRAPHY

[1] D. Saha, “A Brief Introduction to Artificial Intelligence.”

[2] “Deep learning.” Accessed: Apr. 09, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Deep_learning

[3] N. Buduma and N. Locascio, “Deep Learning DESIGNING NEXT-GENERATION MACHINE INTELLIGENCE ALGORITHMS Nikhil Buduma with contributions by Nicholas Locascio.”

[4] I. Alžběta Turečková, “Využití metod hlubokého učení v počítačovém vidění.”

[5] D. Nova, A. Ferreira, and P. Cortez, “A Machine Learning Approach to Detect Violent Behaviour from Video.”

[6] B. Omarov, S. Narynov, Z. Zhumanov, A. Gumar, and M. Khassanova, “State-of-the-art violence detection techniques in video surveillance security systems: A systematic review,” *PeerJ Comput Sci*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.920.

[7] Y. Li, R. Xia, Q. Huang, W. Xie, and X. Li, “Survey of Spatio-Temporal Interest Point Detection Algorithms in Video,” *IEEE Access*, vol. 5, pp. 10323–10331, 2017, doi: 10.1109/ACCESS.2017.2712789.

[8] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” 2005. [Online]. Available: http://lear.inrialpes.fr

[9] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.2661

[10] W. S. Mcculloch and W. Pitts, “A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY,” 1943.

[11] F. Rosenblatt, “The Perceptron - A perceiving and recognizing automaton,” 1957.

[12] D. Rumelhart, G. Hinton, and R. Williams, “Learning representations by back-propagating errors,” vol. 323, 1986.

[13] L. Liu *et al.*, “Deep Learning for Generic Object Detection: A Survey,” Sep. 2018, [Online]. Available: http://arxiv.org/abs/1809.02165

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks." [Online]. Available: http://code.google.com/p/cuda-convnet/

[16] Y. Lecun, L. Eon Bottou, Y. Bengio, and P. H. Abstract|, "Gradient-Based Learning Applied to Document Recognition."

[17] V. Sze and T.-J. Yang, "Efficient Image Processing with Deep Neural Networks."

[18] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[19] C. Szegedy *et al.*, "Intriguing properties of neural networks," Dec. 2013, [Online]. Available: http://arxiv.org/abs/1312.6199

[20] S. Hochreiter and J. ¨ Urgen Schmidhuber, "Long Short-Term Memory."

[21] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?"

[22] A. Vaswani *et al.*, "Attention Is All You Need."

[23] P. Quentin, S. Swan, W. Hugo, R. Léo, H. Siba, and Y. Antoun, "Balancing Accuracy and Training Time in Federated Learning for Violence Detection in Surveillance Videos: A Study of Neural Network Architectures," Jun. 2023, [Online]. Available: http://arxiv.org/abs/2308.05106

[24] A. Lambebo Tonja, M. Arif, O. Kolesnikova, A. Gelbukh, and G. Sidorov, "Detection of Aggressive and Violent Incidents from Social Media in Spanish using Pre-trained Language Model," 2022. [Online]. Available: http://ceur-ws.org

[25] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent Flows: Real-Time Detection of Violent Crowd Behavior *." [Online]. Available: www.openu.ac.il/home/hassner/

[26] P. Zhou, Q. Ding, H. Luo, and X. Hou, "Violent Interaction Detection in Video Based on Deep Learning," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2017. doi: 10.1088/1742-6596/844/1/012044.

[27] A. Al-Dhamari, R. Sudirman, and N. H. Mahmood, "Transfer Deep Learning along with Binary Support Vector Machine for Abnormal Behavior Detection," *IEEE Access*, vol. 8, pp. 61085–61095, 2020, doi: 10.1109/ACCESS.2020.2982906.

[28] J. Lin and W. Wang, "LNCS 5879 - Weakly-Supervised Violence Detection in Movies with Audio and Video Based Co-training," 2009.

[29] A. Datta, M. Shah, N. Da, and V. Lobo, "Person-on-Person Violence Detection in Video Data."

[30] J. Nam, M. Alghoniemy, and A. H. Tewfik, "AUDIO-VISUAL CONTENT-BASED VIOLENT SCENE CHARACTERIZATION."

[31] F. De Souza and H. Pedrini, "Detection of Violent Events in Video Sequences Based on Census Transform Histogram," in *Proceedings - 30th Conference on Graphics, Patterns and Images, SIBGRAPI 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 323–329. doi: 10.1109/SIBGRAPI.2017.49.

[32] M.-Y. Chen and A. Hauptmann, "MoSIFT: Recognizing Human Actions in Surveillance Videos," 2009.

[33] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu, "Violence detection using Oriented VIolent Flows," *Image and Vision Computing*, vol. 48–49. Elsevier Ltd, pp. 37–41, Apr. 01, 2016. doi: 10.1016/j.imavis.2016.01.006.

[34] A. Ben Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Expert Systems with Applications*, vol. 91. Elsevier Ltd, pp. 480–491, Jan. 01, 2018. doi: 10.1016/j.eswa.2017.09.029.

[35] J. Xie, W. Yan, C. Mu, T. Liu, P. Li, and S. Yan, "Recognizing violent activity without decoding video streams," *Optik (Stuttg)*, vol. 127, no. 2, pp. 795–801, Jan. 2016, doi: 10.1016/j.ijleo.2015.10.165.

[36] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno, "Fight Recognition in Video Using Hough Forests and 2D Convolutional Neural Network," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4787–4797, Oct. 2018, doi: 10.1109/TIP.2018.2845742.

[37] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[39] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Dec. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[42] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, p. p9420, Oct. 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[43] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "Rethinking the Inception Architecture for Computer Vision."

LIST OF ABBREVIATIONS

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
CENTRIST	Census Transform Histogram
CNN	Convolutional Neural Network
CNN	Convolutional Neural Network
CNN-LSTM	Combination of Convolutional and LSTM networks
DenseNet	Densely Connected Convolutional Network
DL	Deep Learning
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
FFNN	Feed-Forward Neural Network
FPR	False Positive Rate
GAN	Generative Adversarial Network
GPT	Generative Pre-trained Transformer
GPU	Graphical Processing Unit
HOG	Histograms of Oriented Gradient
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LSTM	Long Short-Term Memory
LSVRC	Large Scale Visual Recognition Challenge
MAC	Multiply-Accumulate Operations
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error

NN	Neural Network
R^2	R-squared Score
ReLU	Restricted Linear Unit
ResNet	Residual Network
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RSI	Relative Strength Index
SGD	Stochastic Gradient Descent
STIP	Spatio-Temporal Interest Point
Tanh	Hyperbolic tangent
TPR	True Positive Rate
TPU	Tensor Processing Unit
VAE	Variable Autoencoder
VGG	Visual Geometry Group
VS Code	Visual Studio Code

LIST OF FIGURES

LIST OF TABLES

APPENDICES

Appendix A: Web application code written in Streamlit Python to test trained model

```python
import streamlit as st
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os

# Import a pre-trained model for use
model = load_model('model/vgg_violence_model.h5')
show_st_image = 0

# Set the initial value for image mean used in mean subtraction
mean = np.array([123.68, 116.779, 103.939][::1], dtype="float32")

# Create a queue to store predictions and initialise it as an empty array
Q = []

# Data augmentation for preprocessing
data_augmentation = ImageDataGenerator(
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest"
)

def preprocess_frame(frame):
    # Apply data augmentation to enhance diversity and size
    augmented_image = data_augmentation.random_transform(frame)

    # Resize frame to match model input shape
    resized_frame = cv2.resize(augmented_image, (224, 224))

    # Convert BGR to RGB
    rgb_frame = cv2.cvtColor(resized_frame, cv2.COLOR_BGR2RGB)

    # Perform mean subtraction
    frame = rgb_frame.astype("float32")
    frame -= mean

    return frame
```

1/2

```python
def main():
    st.title("Violent Behavior Detection")

    if show_st_image:
        st.image("img/image.png", use_column_width=True)

    # Fetch uploaded video file
    uploaded_file = st.sidebar.file_uploader("Upload a video file", type=["mp4", "mov"])

    if uploaded_file:
        with open("temp_video.mp4", "wb") as f:
            f.write(uploaded_file.read())

        # Display uploaded video
        st.video("temp_video.mp4")

        # Add a button to trigger violence detection
        if st.sidebar.button("Detect Violence"):
            # Open video file
            cap = cv2.VideoCapture("temp_video.mp4")

            # Get dimensions of the video frames
            ret, frame = cap.read()
            if ret:
                H, W, _ = frame.shape

                # Loop over frames from the video file stream
                while True:
                    ret, frame = cap.read()
                    if not ret:
                        break

                    # Preprocess frame
                    preprocessed_frame = preprocess_frame(frame)

                    # Make predictions on the frame
                    preds = model.predict(np.expand_dims(preprocessed_frame, axis=0))[0]
                    Q.append(preds)

                    # Perform prediction averaging
                    results = np.array(Q).mean(axis=0)
                    violence_prob = results[1]

                    # Print intermediate results
                    print("Violence probability:", violence_prob)

                # Decide label based on violence probability
                overall_label = 'Violence Detected' if violence_prob > 0.50 else 'No Violence Detected'

                # Print the result
                st.markdown(overall_label, unsafe_allow_html=True)

            # Release the file pointers
            cap.release()
            cv2.destroyAllWindows()

            # Remove temporary file
            os.remove("temp_video.mp4")

if __name__ == "__main__":
    main()
```

2/2

Appendix B: Software dependencies for this work is outlined in Table 7 below

Table 7 Software dependencies and versions

Library name	Version
absl-py	2.1.0
asttokens	2.4.1
astunparse	1.6.3
certifi	2024.2.2
charset-normalizer	3.3.2
colorama	0.4.6
comm	0.2.1
contourpy	1.2.0
cycler	0.12.1
debugpy	1.8.1
decorator	5.1.1
dm-tree	0.1.8
executing	2.0.1
flatbuffers	24.3.7
fonttools	4.49.0
gast	0.5.4
google-pasta	0.2.0
grpcio	1.62.1
h5py	3.10.0
idna	3.6
imutils	0.5.4
ipykernel	6.29.3

ipython	8.22.2
jedi	0.19.1
joblib	1.3.2
jupyter_client	8.6.0
jupyter_core	5.7.1
keras	3.0.5
kiwisolver	1.4.5
libclang	16.0.6
Markdown	3.5.2
markdown-it-py	3.0.0
MarkupSafe	2.1.5
matplotlib	3.8.3
matplotlib-inline	0.1.6
mdurl	0.1.2
ml-dtypes	0.3.2
namex	0.0.7
nest-asyncio	1.6.0
numpy	1.26.4
opencv-python	4.9.0.80
opt-einsum	3.3.0
packaging	23.2
pandas	2.2.1
parso	0.8.3
pillow	10.2.0
platformdirs	4.2.0

prompt-toolkit	3.0.43
protobuf	4.25.3
psutil	5.9.8
pure-eval	0.2.2
Pygments	2.17.2
pyparsing	3.1.2
python-dateutil	2.9.0.post0
pytz	2024.1
pywin32	306
pyzmq	25.1.2
requests	2.31.0
rich	13.7.1
scikit-learn	1.4.1.post1
scipy	1.12.0
six	1.16.0
stack-data	0.6.3
tensorboard	2.16.2
tensorboard-data-server	0.7.2
tensorflow	2.16.1
tensorflow-intel	2.16.1
tensorflow-io-gcs-filesystem	0.31.0
termcolor	2.4.0
threadpoolctl	3.3.0
tornado	6.4
tqdm	4.66.2

traitlets	5.14.1
typing_extensions	4.10.0
tzdata	2024.1
urllib3	2.2.1
wcwidth	0.2.13
Werkzeug	3.0.1
wrapt	1.16.0