

**7CCSMPRJ**  
Individual Project Submission 2024/25

Name: Akshita Singh

Student Number: 24083555

Degree Programme: Data Science MSc

Project Title: A Hybrid CNN-LSTM Architecture for Efficient Video-Based Violence Detection

Supervisor: Martim Brandao

Word count: 16315

**RELEASE OF PROJECT**

Following the submission of your project, the Department would like to make it publicly available via the library electronic resources. You will retain copyright of the project.

I agree to the release of my project

I do not agree to the release of my project

**Signature:**



**Date:** 7 August 2025

7CCSMPRJ MSc Project

# A HYBRID CNN-LSTM ARCHITECTURE FOR EFFICIENT VIDEO- BASED VIOLENCE DETECTION

**Name: Akshita Singh**  
**Student Number: 24083555**  
**Degree Programme: Data Science MSc**

**Supervisor's Name: Martim Brandao**

**This dissertation is submitted for the degree of MSc in Data Science.**

## ABSTRACT

- The proliferation of surveillance cameras has generated a colossal volume of video data, making manual monitoring for violent incidents impractical. This has spurred a demand for automated systems that can proactively detect threats in real time. This project addresses this need by developing and evaluating an efficient deep learning model for violence detection. The primary objective was to investigate the effectiveness of a lightweight hybrid architecture, combining a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network, to balance high accuracy with computational efficiency suitable for practical deployment.
- The developed model utilizes a pre-trained MobileNetV2 to extract spatial features from individual video frames, which are then fed into an LSTM network to analyze their temporal evolution. The model was trained on the Real Life Violence Situations Dataset (RLVSD) dataset, a large collection of real-world fight and non-fight clips. On the held-out test set for this domain, the model demonstrated robust performance, achieving an overall accuracy of **94%** with balanced precision (0.95 for violence) and recall (0.92 for violence).
- The main contribution of this work is a stark, quantitative illustration of the critical challenge of model generalization. To assess its capabilities on unseen data, the trained model was evaluated, without any fine-tuning, on the Smart-City CCTV Violence Detection (SCVD) dataset. The performance dropped significantly, with accuracy plummeting to **60%**. This was characterized by a high rate of false positives, indicating the model's inability to adapt to a new domain featuring different environmental contexts and weaponized scenarios.
- The primary conclusion drawn is that while a lightweight CNN-LSTM architecture is a highly effective and computationally efficient solution for violence detection within a specific, well-defined domain, its practical utility is severely hindered by poor generalization. This finding underscores that achieving a deployable, trustworthy system requires more than high accuracy on a single benchmark; it necessitates training on diverse, multi-domain datasets and potentially leveraging more complex architectures to overcome the formidable barrier of domain shift.

## ACKNOWLEDGMENT

I would like to extend my deepest gratitude to my supervisor, ***Martim Brandaو***, for their invaluable guidance, insightful feedback, and unwavering support throughout this project. Their expertise and encouragement were instrumental in navigating the complexities of this research and shaping the final dissertation.

I would also like to thank my family and friends for their constant encouragement, patience, and understanding during this intensive period of work. Their support provided a much-needed foundation of motivation and perspective.

# CONTENTS

<b>1</b>	<b><i>Introduction</i></b>	<b>7</b>
<b>2</b>	<b><i>Background</i></b>	<b>10</b>
<b>3</b>	<b><i>Related Work</i></b>	<b>14</b>
3.1	CNN+LSTM Hybrid Models	14
3.2	3D CNN-Based Models	15
3.3	Two-Stream Networks (RGB + Motion)	16
3.4	Transformer-Based Models	17
3.5	Classical Machine Learning Methods	18
3.6	Summary of Related Works	19
<b>4</b>	<b><i>Approach</i></b>	<b>22</b>
4.1	Problem Statement and Motivation	22
4.2	Research Questions and Objectives	22
4.3	Datasets and Preprocessing	23
4.4	Model Architecture	25
4.5	Experimental Variations and Design Choices	30
4.6	Training Procedure and Evaluation Protocol	32
<b>5</b>	<b><i>Results</i></b>	<b>34</b>
<b>6</b>	<b><i>Legal, Social, Ethical and Professional Issues</i></b>	<b>40</b>
<b>7</b>	<b><i>Conclusion</i></b>	<b>43</b>
<b>8</b>	<b><i>References</i></b>	<b>45</b>

## NOMENCLATURE

<b>3D CNN</b>	Three-Dimensional Convolutional Neural Network	7
<b>AI</b>	Artificial Intelligence	7
<b>AUC</b>	Area Under the Curve	15
<b>BCS</b>	British Computer Society	44
<b>Bi-LSTM</b>	Bidirectional Long Short-Term Memory	15
<b>C3D</b>	Convolutional 3D	10
<b>CCTV</b>	Closed-Circuit Television	7
<b>CNN</b>	Convolutional Neural Network	7
<b>ConvLSTM</b>	Convolutional Long Short-Term Memory	11
<b>CUE-Net</b>	A hybrid convolutional and transformer-based model for video analysis	7
<b>GPU</b>	Graphics Processing Unit	13
<b>GRU</b>	Gated Recurrent Unit	23
<b>IET</b>	Institution of Engineering and Technology	44
<b>LSTM</b>	Long Short-Term Memory	8
<b>RNN</b>	Recurrent Neural Network	11
<b>RLVS</b>	Real Life Violence Situations	11
<b>RWF-2000</b>	Real-World Fights 2000	11
<b>RWIS</b>	Real World Intelligence Surveillance	8
<b>SCVD</b>	Smart-City CCTV Violence Detection	12
<b>SOTA</b>	State-Of-The-Art	20
<b>SVM</b>	Support Vector Machine	15
<b>UDA</b>	Unsupervised Domain Adaptation	20
<b>ViT</b>	Vision Transformer	11

## LIST OF FIGURES AND TABLES

FIGURE 1: EXAMPLE 3D CNN (C3D) ARCHITECTURE [11] .....	11
FIGURE 2: A CONCEPTUAL FLOWCHART OF THE HYBRID CNN+LSTM ARCHITECTURE .....	12
FIGURE 3: DATA PREPROCESSING PIPELINE .....	24
FIGURE 4: MOEL ARCHITECTURE .....	26
<i>FIGURE 5: CONFUSION MATRIX ON RLVSD TEST SET .....</i>	35
<i>FIGURE 6: : TRAIN AND VALIDATION ACCURACY AND LOSS GRAPH RLVSD DATASET.....</i>	36
<i>FIGURE 7: SCVD CONFUSION MATRIX ON SCVD TEST DATASET.....</i>	37

TABLE 1: KEY DATASETS FOR PRE-TRAINING IN VIDEO ANALYSIS .....	12
TABLE 2: COMMON PREPROCESSING TECHNIQUES IN VIDEO VIOLENCE DETECTION.....	13
TABLE 3: RECENT STUDY SUMMARY .....	19
TABLE 4: PROPOSED CNN+LSTM MODEL ARCHITECTURE .....	28
<i>TABLE 5: CLASSIFICATION REPORT RLVSD DATASET.....</i>	34
<i>TABLE 6: CLASSIFICATION REPORT SCVD TEST DATASET.....</i>	36

# 1 INTRODUCTION

In an era defined by the rapid expansion of digital oversight, the deployment of intelligent surveillance systems has become a cornerstone of modern public safety strategies. Cities worldwide are increasingly turning to technology, not merely to record events, but to proactively detect and respond to threats in real time.[1][2] This technological shift is driven by a dual impetus: a societal demand for safer public spaces and the sheer impracticality of manually monitoring the colossal volume of footage generated by ubiquitous CCTV cameras.[3] While official data points to a general decrease in many categories of violent crime in early 2024, public perception, often shaped by high-profile incidents, continues to express significant concern.[4][5] This underscores the need for systems that can enhance security and, just as importantly, foster public trust and accountability.

The core challenge lies in creating systems that can autonomously and accurately interpret complex human behaviors. A physical altercation, for instance, unfolds over seconds, composed of subtle spatial and temporal cues that distinguish it from benign, vigorous activity. Traditional surveillance, which relies on human operators to watch multiple screens, is ill-equipped for this task, suffering from cognitive overload and inevitable lapses in attention.[6] This is where deep learning offers a transformative solution. This project explores the development and evaluation of an automated violence detection system, motivated by the potential to create a rapid-response tool that can flag aggressive behavior, assist law enforcement, and provide an objective record of events. While not its sole purpose, a key motivating factor is the system's potential to serve as an impartial tool for accountability, capable of identifying any form of violence, including incidents of excessive force, thereby contributing to transparency in policing.[7][8]

Recent years have seen a surge in research into video-based violence detection, with a variety of deep learning architectures demonstrating considerable promise. Methodologies have evolved from foundational 3D Convolutional Neural Networks (3D CNNs), which learn features in both space and time simultaneously, to sophisticated Transformer-based models that excel at capturing long-range dependencies in video sequences. The most advanced of these, such as the CUE-Net proposed by Chamalke et al. (2024) [18], have set new benchmarks by combining convolutional features with global self-attention, achieving accuracies as high as 99.5% on certain datasets. Concurrently, researchers like Abdali (2021)

have explored pure Vision Transformers (ViT), demonstrating that even without convolutional biases, these models can achieve high accuracy (96.25%) on real-world violence data, albeit often requiring extensive datasets for training.[9]

This study, however, focuses on a different but highly practical segment of this technological landscape: the hybrid CNN-LSTM model. This architecture combines a pre-trained 2D CNN for powerful per-frame spatial feature extraction with a Long Short-Term Memory (LSTM) network to model the temporal evolution of these features. This approach, utilized effectively by researchers like Asad et al. (2021) to achieve high accuracy on datasets like the Hockey Fight benchmark, represents a pragmatic trade-off between performance and computational efficiency. Our implementation specifically employs a lightweight MobileNetV2 as the CNN backbone—its weights frozen after being pre-trained on ImageNet—paired with an LSTM. This design choice is deliberate, aiming to build an efficient yet powerful model that is well-suited for practical deployment while providing a clear baseline for evaluating fundamental challenges in violence detection.

To this end, this research addresses the following critical questions:

- How effectively can a lightweight CNN+LSTM architecture, trained on a single, coherent domain of real-world violence, perform in distinguishing violent from non-violent actions?
- What are the precise, quantifiable implications of domain shift when this model is deployed on a different, more challenging dataset without retraining?
- How does the performance of this practical, efficient model compare to the state-of-the-art, and what does this comparison reveal about the trade-offs between model complexity and generalization?

To answer these questions, we conducted a two-stage evaluation. First, the model was trained and validated on the real-life-violence-situations-dataset dataset, a large collection of real-world fight and non-fight video clips. On this in-domain data, the model demonstrated robust performance, achieving an overall accuracy of approximately 94%. The confusion matrix for this test revealed a well-balanced system, with a precision of 0.95 and recall of 0.92 for the "Violence" class, and a precision of 0.92 and recall of 0.95 for the "NonViolence" class. These metrics indicate that the model reliably identified violent instances with very few false alarms or missed events.

The second stage of our evaluation was designed to test the model's generalization capabilities. We subjected the model, without any fine-tuning, to the Smart-City CCTV Violence Detection (SCVD) dataset. This dataset presents a significant domain shift, featuring distinct scenarios of weaponized and non-weaponized violence captured from static surveillance cameras, a context for which the model was not trained. The results were stark: the overall accuracy plummeted to approximately 60%. The balanced performance collapsed into a skewed trade-off; while the recall for "Violence" remained high at 0.72 (meaning it still caught a majority of violent acts), its precision dropped to a low 0.42. This indicates that the model began to incorrectly label a large number of non-violent clips as violent, leading to a high rate of false positives. This performance degradation clearly illustrates one of the most significant hurdles for the real-world deployment of AI surveillance: models that perform exceptionally well on their training data can fail significantly when confronted with new, unseen environments. This finding aligns with broader challenges noted in the field, where factors like camera angles, lighting, and context create formidable barriers to generalization.[190]

In conclusion, this research confirms that a lightweight CNN+LSTM architecture can be highly effective for violence detection within a defined domain, offering a computationally efficient alternative to more complex models. However, its primary contribution is the stark, quantitative illustration of the domain shift problem. The drop in accuracy from 94% to 60% provides a clear and sobering benchmark for the challenges that must be overcome for these systems to be deployed reliably at scale. While the frontier of research is being pushed by powerful Transformer-based models, our findings underscore the continued importance of focusing on generalization. Future work must prioritize the development of models that are not only accurate but also adaptable, leveraging techniques like domain adaptation, data augmentation, and training on more diverse, multi-domain datasets. Only by solving this generalization puzzle can automated surveillance systems truly fulfill their promise of enhancing public safety and accountability in a consistent and trustworthy manner.

## 2 BACKGROUND

The automated detection of violence in video footage is a complex task at the intersection of computer vision and pattern recognition. It requires a system not only to understand the spatial content of individual frames—identifying people, objects, and their configurations—but also to comprehend the temporal evolution of these elements over time. The subtle yet critical difference between a high-five and a slap, or a playful chase and a genuine assault, is encoded in these spatio-temporal dynamics. Deep learning, with its ability to learn hierarchical features directly from data, has emerged as the most effective paradigm for tackling this challenge. This background section details the fundamental deep learning components, architectures, and techniques that form the foundation of modern video violence detection systems.

- **Spatial Feature Extraction: The Role of Convolutional Neural Networks**

The cornerstone of modern computer vision is the Convolutional Neural Network (CNN), a class of deep neural networks designed to process grid-like data, such as images. CNNs automatically learn a hierarchy of spatial features by applying a series of learnable filters (or kernels) across an input image. Early layers learn to detect simple patterns like edges and textures, while deeper layers combine these to recognize more complex structures like objects and human forms.

- **2D CNNs:** A standard 2D CNN processes images frame by frame. When applied to video, it treats each frame as an independent image, extracting a rich set of spatial features but inherently ignoring the temporal relationship between frames. Prominent 2D architectures like **ResNet** (Residual Network) [12] and the lightweight **MobileNetV2** [13] are frequently used as powerful feature extractors. As noted in the project's approach, a pre-trained MobileNetV2 was chosen for its balance of efficiency and high accuracy, making it suitable for practical deployment [**Error! Reference source not found.**].
- **3D CNNs:** To directly model motion, 2D convolutions can be extended into the temporal dimension, resulting in 3D CNNs. These networks operate on video volumes (a stack of consecutive frames) and use 3D filters to learn spatio-temporal features simultaneously. An early and influential example is the **C3D (Convolutional 3D)** architecture [14], which consists of multiple layers of 3x3x3 convolutions and pooling operations. As noted by Ullah et al. (2019), 3D CNNs have demonstrated state-of-the-art accuracy by directly capturing motion patterns [Related Work]. More

advanced architectures like **ResNet3D** adapt the successful residual learning framework to the video domain, enabling the training of much deeper and more powerful models for action recognition [11].

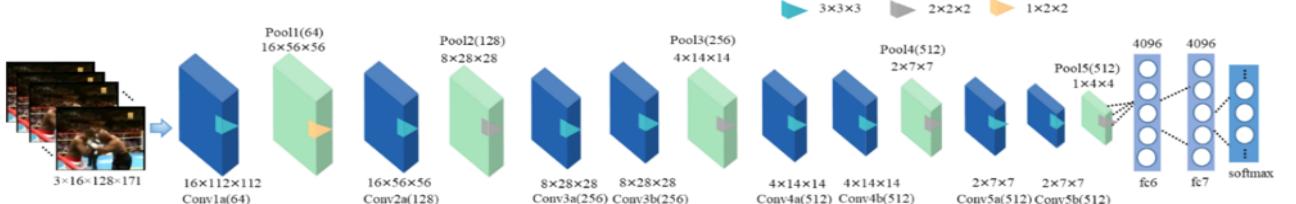


Figure 1: Example 3D CNN (C3D) architecture [11]

- Temporal Feature Modeling: Recurrent and Transformer Networks

While CNNs excel at spatial analysis, understanding the narrative of a video requires modeling its temporal dimension.

- **Recurrent Neural Networks (RNNs):** RNNs are designed to process sequential data. However, simple RNNs struggle with the vanishing gradient problem, limiting their ability to learn long-term dependencies. The **Long Short-Term Memory (LSTM)** network [15], a specialized type of RNN, overcomes this through a sophisticated gating mechanism. An LSTM can process a sequence of frame-level features from a CNN to learn the temporal progression of a scene, making it highly effective at distinguishing between abrupt, violent actions and other types of movement. Variants like the **Convolutional LSTM (ConvLSTM)** [16] embed convolutional operations within the LSTM gates, allowing the model to preserve spatial structure while modeling temporal transitions.
- **Transformer-Based Models:** More recently, Transformer architectures, originally from natural language processing, have been adapted for video analysis. Models like the Vision Transformer (ViT) and Video Swin Transformer use self-attention mechanisms to model global relationships between all frames in a sequence simultaneously. This allows them to capture complex, long-range dependencies that can be challenging for LSTMs. As highlighted in the literature review, hybrid models combining CNNs with transformers, such as Action-VST [17] and CUE-Net [18], have set new state-of-the-art performance benchmarks on datasets like RLVS and RWF-2000 [Error! Reference source not found., Error! Reference source not found.].

- Hybrid Architectures for Video Analysis

To balance performance and efficiency, many systems employ hybrid architectures that combine different modeling paradigms.

- **The CNN+LSTM Paradigm:** This is a popular and effective framework that decouples spatial and temporal feature learning. As implemented in this project, the architecture involves a pre-trained 2D CNN (MobileNetV2) to extract feature vectors from each frame, followed by an LSTM layer that processes this sequence to make a final classification [Error! Reference source not found.]. This design is computationally efficient and leverages the power of pre-trained models.
- **Two-Stream Networks:** This approach explicitly incorporates motion by processing two parallel streams: one for RGB frames (appearance) and another for motion representation, typically optical flow or frame differences. The features from both streams are fused to make a final prediction. Cheng et al. (2021) demonstrated the effectiveness of this method with their Flow Gated Network on the RWF-2000

dataset, underscoring the importance of integrating motion information for real-world surveillance [Related Work] [3].

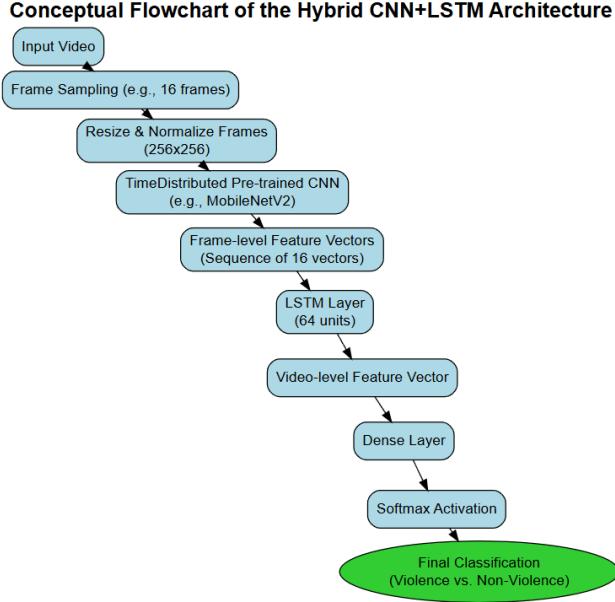


Figure 2: A conceptual flowchart of the hybrid CNN+LSTM architecture

- The Role of Pre-training and Transfer Learning

Training deep neural networks from scratch requires vast amounts of labeled data, which is often scarce in specialized domains. **Transfer learning** is a critical technique that mitigates this by leveraging knowledge from a large, general-purpose dataset and applying it to a target task. As noted in the project background, training a model from scratch can lead to suboptimal performance and overfitting on smaller datasets [Background].

In this context, CNN backbones are pre-trained on large-scale image or video datasets. This initializes the model's weights to a state where they can already recognize a rich hierarchy of visual features. The model is then fine-tuned on the smaller, task-specific violence dataset, a process that is much faster and more data-efficient.

Table 1: Key Datasets for Pre-training in Video Analysis

<b>Dataset</b>	<b>Content</b>	<b>Scale</b>	<b>Primary Use in Violence Detection</b>
<i>ImageNet</i>	Over 1.2 million images across 1000 object categories.	Large-Scale Images	Pre-training 2D CNN backbones for spatial feature extraction.
<i>Kinetics-400</i>	Hundreds of thousands of video clips across 400 human action classes.	Large-Scale Videos	Pre-training 3D CNNs to learn generic spatio-temporal features.

- Datasets and Data Preprocessing

The performance of any violence detection model is heavily dependent on the data used for its training and evaluation.

- **Key Datasets:** The field relies on several public benchmarks, each with distinct characteristics. The datasets central to this research are:

- **RWIS (Real World Intelligence Surveillance) / RLVS (Real Life Violence Situations):** A large, balanced dataset of 2,000 real-world video clips (1,000

- violent, 1,000 non-violent) sourced from platforms like YouTube. Its "in-the-wild" nature, with varied camera angles and environments, makes it a robust benchmark for training general-purpose models [Approach, Background].
- **SCVD (Smart-City CCTV Violence Detection):** A dataset comprising CCTV footage with distinct classes for non-violence, violence, and weaponized violence. It represents a different domain from RLVSD, characterized by static cameras, lower resolution, and specific surveillance scenarios [Background].
  - Other notable datasets include the **Hockey Fight** dataset, **Crowd Violence**, and **RWF-2000 (Real-World Fights)**, which was introduced by Cheng et al. (2021) [Error! Reference source not found.] [3].
  - **Data Preprocessing:** Before being fed into a model, raw video data must undergo several preprocessing steps. As detailed in the project's methodology, this pipeline ensures consistent and optimized input [Error! Reference source not found.].

Table 2: Common Preprocessing Techniques in Video Violence Detection

<b>Technique</b>	<b>Purpose</b>	<b>Project Implementation Detail</b>
<b>Frame Sampling</b>	Extracts a fixed-length sequence of frames (e.g., 16 frames) from a variable-length video to create a consistent input size for the temporal model.	16 frames were sampled uniformly from each video.
<b>Resizing</b>	Standardizes the spatial dimensions of each frame to match the input requirements of the CNN backbone.	Frames were resized to 256x256 pixels.
<b>Normalization</b>	Scales pixel values (e.g., to a range) to stabilize and accelerate the training process.	Pixel values were divided by 255.0.
<b>Data Augmentation</b>	Applies random transformations to the training data to increase its diversity and prevent overfitting.	Light augmentation (e.g., horizontal flips) was explored but did not yield notable improvements.

- Implementation and Training Environment

The implementation and training of these models are facilitated by modern deep learning frameworks and high-performance hardware.

- **Hardware:** Training is computationally intensive and is typically performed on Graphics Processing Units (GPUs). This project utilized an NVIDIA A100 GPU via Google Colab Pro, which significantly reduced training time to approximately 30-60 minutes for 5 epochs [Approach].
- **Software:** The model was implemented using Python with the **TensorFlow** and **Keras** libraries. A custom VideoDataGenerator was created to efficiently handle on-the-fly video loading and preprocessing, which is crucial for managing memory usage with large video files [Approach].
- **Training Procedure:** The model was trained using the **Adam optimizer** with a categorical cross-entropy loss function. The CNN backbone weights were frozen to act as a fixed feature extractor, drastically reducing the number of trainable parameters (to ~345k) and mitigating overfitting. The model was trained for 5 epochs, as performance on the validation set plateaued after this point [Approach, Results].

# 3 RELATED WORK

Research on video-based violence detection has advanced rapidly in the last five years, driven by deep learning models that capture both spatial and temporal features of video. We review key approaches grouped by model architecture:

- (1) CNN+LSTM hybrids,
- (2) 3D CNNs,
- (3) two-stream networks fusing RGB frames with motion (optical flow or frame differences),
- (4) transformer-based models,
- (5) classical machine learning methods for context.

For each, we highlight representative open-access studies from 2018–2024, their model innovations, datasets, and performance results on common benchmarks (Hockey Fight, Crowd Violence, RWF-2000, RLVS, UCF-Crime, etc.). **Table 1** at the end provides a comparative summary.

## 3.1 CNN+LSTM Hybrid Models

A popular framework is to use a 2D Convolutional Neural Network (CNN) for spatial feature extraction on each frame, followed by a Recurrent Neural Network (typically an LSTM) to learn temporal dynamics. Sudhakaran and Lanz’s early work (2017) pioneered a convolutional LSTM for violence detection, showing that integrating CNNs with LSTM can outperform plain LSTMs while using fewer parameters [20]. Building on this, many recent studies adopt CNN+LSTM hybrids. For example, Asad *et al.* (2021) used a pre-trained VGG16 CNN to extract frame features, then fed them into an LSTM; their model achieved **98.8%** accuracy on the Hockey Fight dataset and **97.1%** on the Crowd Violence dataset [21]. Similarly, Shoaib and Sayed (2021) employed a ResNet-101 backbone with an LSTM for sequence classification, augmented by a human detection step (ROI-based keypoint filtering to focus on persons) – reporting **95.7%** accuracy on the KTH actions dataset and **88.2%** on a custom violence dataset [22]. Samuel *et al.* (2019) developed a bi-directional LSTM framework for violence in football stadium videos, fusing features from a “violence” model, a “human parts” model, and a “negative (no-violence)” model; this ensemble yielded a

**94.5%** violence detection rate in their stadium surveillance scenario [23]. These CNN+RNN approaches generally perform well on curated datasets – often exceeding 90% accuracy – because the CNN captures detailed spatial cues (like people and objects involved) while the LSTM captures temporal patterns (e.g. sudden movements). However, a known challenge is overfitting to limited training scenes; Sudhakaran & Lanz noted that a Conv-LSTM can help mitigate overfitting compared to a plain LSTM [20]. Soeleman *et al.* (2022) also found that with careful hyperparameter tuning, an LSTM-based model slightly **outperformed a transformer** on Hockey Fight and Crowd Violence datasets (mean AUC 0.976 vs 0.954 on Hockey) when training data is limited [24][19]. This suggests CNN+LSTM models remain competitive, especially on smaller datasets or when computational simplicity is needed. Nonetheless, their performance may degrade on more complex “in the wild” footage without additional techniques (e.g. attention mechanisms or data augmentation). Recent works have thus added attention modules to CNN-LSTM models – for instance, Aktı *et al.* (2019) introduced an attention layer in a CNN+biLSTM pipeline for surveillance fight detection, which improved sensitivity to subtle violent motions [25].

**Datasets:** CNN+LSTM models have been evaluated on a variety of benchmarks. Common small benchmarks include **Hockey Fight**[19] (1000 hockey game clips with fights vs non-fights) and the **Crowd Violence** dataset[19] (also known as Violent-Flows, 246 clips of crowded scenes [19]). Models like Ullah *et al.* (2019) and Asad *et al.* (2021) trained on these datasets reported 95–99% accuracy[19][21]. Some works also use **Movies** or **Películas** dataset (violent scenes from films) [26], and **RLVS** (Real-Life Violence Situations), a 2,000-video collection of real street fight footage (Kaggle 2019). For example, a CNN-LSTM by Abdali (2021) fine-tuned on RLVS achieved **96.25%** accuracy [19]. A few studies address longer untrimmed videos: for instance, violence as an anomaly in **UCF-Crime** (128 long surveillance videos of real crimes) has been tackled with MIL-based LSTM models [27], though this crosses into anomaly detection rather than direct classification. In general, CNN+LSTM methods have been very successful on trimmed video datasets, but their performance on untrimmed or more diverse videos can drop without additional design considerations (e.g. the bi-LSTM attention model of Aktı *et al.* aimed to handle complex surveillance scenes [25]).

## 3.2 3D CNN-Based Models

Three-dimensional CNNs (which learn spatio-temporal convolutional kernels) have proven effective for violence recognition by directly capturing motion across consecutive frames. These models process a stack of frames (e.g. 16 or 32 frames) as a volume, extracting features in both space and time. Ullah *et al.* (2019) demonstrated the power of 3D CNNs for violence detection, achieving **96%** accuracy on Hockey Fight and **98%** on Crowd Violence using a C3D-based network [11]. Song *et al.* (2019) went further by proposing a *modified 3D ConvNet* with an improved training strategy: they introduced a key-frame sampling method to reduce redundancy in longer clips, using short 5s clips for fights and tailored sampling for longer videos [26]. Their 3D CNN scheme (8 convolutional layers + 3D pooling) attained extremely high accuracy on multiple benchmarks – **99.62%** on Hockey Fight, **99.97%** on a Movies fights dataset, and **94.3%** on the Crowd Violence dataset [26]. These near-perfect scores on Hockey/Movie fights indicate that 3D CNNs can learn discriminative punch/kick motion patterns when training data is sufficient. Another notable study by Li *et al.* (2018) developed an end-to-end deep 3D CNN for multi-player violence detection; it achieved **98.3%** on Hockey and **97.2%** on Crowd Violence [28], using a smaller 3D CNN architecture (hence fewer parameters and potentially less overfitting). Sernani *et al.* (2021) compared

several architectures on standard sets and found that a transfer-learning approach with C3D (pre-trained on Sports1M) plus an SVM classifier can also excel – reaching **97.9%** Hockey and **99.6%** Crowd accuracy [28]. This suggests that 3D features are highly separable for violence vs non-violence when using pre-trained spatio-temporal filters. However, a slight drop in performance on the Crowd dataset was observed for some methods trained from scratch (e.g. Song’s method 99.6%→94.3% from Hockey to Crowd [28]), indicating that more diverse, uncontrolled scenarios are challenging. Consequently, newer 3D CNN works have embraced data augmentation and larger training sets.

A major contribution in this vein is **RWF-2000**, a large-scale Real-World Fighting dataset introduced by Cheng *et al.* (2021). It contains 2,000 surveillance clips (5 seconds each, 1000 violent, 1000 non-violent) from varied real scenes [3][18]. Along with RWF-2000, Cheng *et al.* proposed a *Flow Gated Network* – a two-stream 3D CNN that fuses RGB and optical flow streams. Their model obtained **87.25%** accuracy on RWF-2000’s test set [3]. While lower than lab datasets, this result was significant given RWF’s complexity; it underscored the importance of integrating motion information (optical flow) with 3D CNN features for real surveillance data [3]. In summary, 3D CNN-based approaches form a strong foundation for violence detection, especially for trimmed videos: they inherently model motion and have achieved state-of-the-art accuracy on many benchmarks. The current frontier is improving their generalization to unconstrained scenarios – an area where hybrid models and transformers are now building upon these 3D convolutions.

### 3.3 Two-Stream Networks (RGB + Motion)

Two-stream networks explicitly incorporate motion by processing RGB frames and motion information in parallel, then fusing the results. The motion stream input can be optical flow (computed between consecutive frames) or *frame differences* (simpler magnitude of pixel changes). Two-stream architecture was originally popularized in action recognition, and they have been adapted for violence detection as well. For instance, **FightNet** by Zhou *et al.* (2017) fused three streams – RGB, optical flow, and acceleration (temporal variations) – demonstrating that combining appearance and motion cues improves violence recognition [29]. In 2018, Xia *et al.* proposed a real-time violence detector using a *bi-channel CNN*: one CNN learned appearance features from frames while another learned features from the difference between adjacent frames [30]. The two feature types were concatenated and fed to an SVM classifier. This deep two-stream + SVM approach outperformed earlier hand-crafted methods (HOG, HOF, etc.) on both the Hockey Fight and Crowd Violence datasets [30], confirming the benefit of motion cues. Notably, Xia’s use of frame-differences offered a computationally cheap alternative to optical flow while still capturing short-term motion changes [27]. Several other works likewise report that frame-difference input can substitute for optical flow in violence detection to reduce complexity, with only minor accuracy trade-offs [27].

The RWF-2000 baseline by Cheng *et al.* is another illustrative two-stream design. Their Flow Gated Network uses a 3D CNN on RGB and another on optical flow, merging them with a learnable gating mechanism[3]. By leveraging both streams, it achieved **87.3%** on RWF-2000 [3] – significantly higher than single-stream CNNs would likely score on this challenging set. This gating idea allows the model to *learn* how much to rely on motion vs appearance for each video (e.g. in some clips, color/appearance differences might be subtle, but motion is decisive, and vice versa). Another recent study introduced a *temporal fusion* module in a CNN+LSTM two-stream model, reaching about **91%** accuracy on RLVS [18]. In

that model, outputs from spatial and temporal CNNs were fused before the LSTM, demonstrating improved performance over a single-stream LSTM [18].

In addition to optical flow, researchers have explored *dynamic images* as a way to encode motion. Dynamic images (Bilen *et al.*, 2016) collapse a video snippet into a single RGB image that represents temporal evolution [31]. Jain and Vishwakarma (2020) applied this technique to violence detection, using an Inception-ResNetV2 CNN on dynamic images [32]. They reported that using dynamic images significantly improved accuracy on Hockey Fight and an in-house real-life violence dataset, as it provided a compact motion representation [32]. Essentially, dynamic images act as an implicit two-stream fusion (since motion is baked into the image). Their approach highlights an advantage of two-stream thinking: by emphasizing motion patterns (whether via optical flow, differences, or dynamic imagery), models can better detect the abrupt movements and interactions indicative of violence.

In summary, two-stream networks (and related strategies to encode motion) consistently show higher robustness to camera motion and scene context, because they focus on the underlying movement dynamics. The trade-off is increased computation (especially for optical flow) and the challenge of optimally fusing streams. Many violent detectors now incorporate some form of two-stream fusion. For example, Mohtavipour *et al.* (2021) combined a CNN on RGB with a CNN on optical flow for prison violence detection, and Ciampi *et al.* (2024) note that several top methods use frame-difference or flow streams instead of raw video to efficiently capture violent motion [27]. Overall, by leveraging motion information, two-stream approaches have pushed state-of-the-art accuracy on benchmarks like RWF-2000 and RLVS, often in the mid-90% range [18], whereas single-stream models would typically be lower.

### 3.4 Transformer-Based Models

Transformer architecture has recently entered video violence detection, inspired by their success in NLP and image recognition. Vision transformers (ViT) and video transformers (e.g. ViViT, TimeSformer, Swin Transformer) dispense with CNN recurrence and instead use self-attention to model global frame relationships[33][9]. This is attractive for violence detection, since transformers can in principle learn long-range temporal dependencies (e.g. a fight might involve intermittent exchanges over several seconds). Several works from 2021 onwards explore this avenue. Abdali (2021) presented a “Data-Efficient Video Transformer” (DEVT) for violence detection, applying a ViT-based model to the RLVS dataset [9]. Despite limited training data, their transformer achieved **96.25%** accuracy on RLVS [9], nearly matching an equivalent CNN-LSTM’s performance. This demonstrated that even without convolutions, a pure transformer could capture the necessary motion cues in short, trimmed clips. Seyed *et al.* (2022) also experimented with a ViT for surveillance fight detection and reported around **84–85%** accuracy on their test scenarios, which was an improvement over their earlier LSTM model (~72%) but still below CNN-based methods [27]. These early attempts indicated that while transformers are promising, they may require more data or hybrid designs to fully surpass CNN/RNN models in this domain [19].

The current trend is to combine transformers with CNN feature extractors – leveraging the strengths of both. For example, **Action-VST** (2023) used a 2D CNN to extract frame features which were then fed into a Video Swin Transformer; this hybrid achieved about **98.7%** accuracy on RLVS [19]. Another study fused a CNN and ViViT in a two-stream fashion, reaching **98–99%** accuracy on RWF-2000 and RLVS [18]. These results set new state-of-the-

art levels on those datasets. The most notable is the work of Chamalke *et al.* (2024), who introduced **CUE-Net**, a *convolutional uniformer ensemble* that marries convolutional local feature encoding with transformer-style global attention. CUE-Net is built on the UniformerV2 architecture (which integrates 3D convolutions and multi-head attention in each block), enhanced with a novel **Modified Efficient Additive Attention (MEAA)** mechanism to reduce the quadratic cost of standard self-attention [18]. Additionally, CUE-Net includes a pre-processing module that uses YOLO to detect people in each frame and crops the video to focus on regions with people (potentially the fighting subjects) [19]. This spatial cropping improves the model’s focus on violent interactions. With these innovations, CUE-Net set a new benchmark: **94.0%** accuracy on RWF-2000 and **99.5%** on RLVS, outperforming all prior published methods on those datasets [19]. In Table 2 of their paper, CUE-Net surpassed a ViViT-based transformer (96.25% on RLVS) and a CNN+ViT hybrid (98.69%), illustrating the advantage of combining convolution for local detail and transformers for global context [19]. Notably, CUE-Net’s attention mechanism (MEAA) improved efficiency and accuracy – the authors report a 1.5% accuracy gain by using MEAA in the global blocks and reducing computation [19].

It’s worth mentioning that transformers typically need large training data to generalize well. In violence detection, many datasets are relatively small (hundreds or thousands of clips). As a result, some studies found transformers can overfit or yield unstable training if applied naively [28][19]. The comparison by Soeleman *et al.* (2022) showed that an optimized LSTM slightly beat a ViT on Hockey/Crowd because the transformer’s training was “erratic” over 100 epochs on limited data [28][19]. To mitigate this, recent works either pre-train transformers on large action datasets or use hybrid models (as described above). Despite these challenges, transformer-based models are now leading on several violence benchmarks. They excel at capturing complex temporal relationships – for example, distinguishing a friendly interaction from a violent one may require understanding context across many frames (which self-attention can theoretically do better than a fixed-size LSTM memory). We are also seeing specialized transformer approaches: Jain *et al.* (2023) proposed **VioNet**, which fuses a Vision Transformer with Bi-LSTM and 3D CNN, aiming to exploit transformer attention while retaining recurrent inductive bias [34]. Such combinations reflect the community’s effort to harness transformers’ strengths for violence detection. In summary, transformer-based models (ViViT, TimeSformer, Swin, Uniformer, etc.) represent the state-of-the-art in this field, especially when augmented with CNNs or domain-specific modules. As larger curated violent datasets emerge and computing power grows, we expect pure transformers to play an even bigger role in violence detection research.

### 3.5 Classical Machine Learning Methods

Before the deep learning era (pre-2015) and in early years of adoption, researchers tackled violence detection with hand-crafted features and traditional classifiers. It is important to acknowledge these classical methods, as they established baseline features and datasets that current deep models still use for evaluation. Early works focused on motion descriptors: for example, *Violent Flows (ViF)* by Hassner *et al.* (2012) computed the pixel-level optical flow magnitude in consecutive frames and used a linear SVM to classify a clip as violent or not [19]. ViF introduced the idea that chaotic motion is a key signal for violence. Zhou *et al.* (2017) extended this to *Oriented Violent Flows (OViF)* by incorporating the orientation of flow vectors, improving detection on non-crowded fights [29]. Other notable descriptors include *MoSIFT* (motion SIFT) and its variant *MoWLD* (Motion Weber Local Descriptor by Zhang *et al.* 2017), which combined optical flow with texture filters to capture local motion

patterns [35]. Mahmoodi and Salajeghe (2019) proposed *HOMO*, a histogram of optical flow magnitudes and orientations, classified by an SVM [36]. These methods were evaluated on datasets like Hockey, the **BEHAVE** interaction dataset, and crowded scenes; many achieved around 80–90% accuracy[19]. For example, Zhang *et al.* reported MoWLD was robust on Hockey/Crowd (improving over basic HOF) [19], and Mahmoodi’s HOMO attained good precision on surveillance videos [36]. However, as deep CNNs emerged, it became clear that learned spatio-temporal features significantly outperform these hand-crafted ones [19]. Ullah *et al.* (2019) noted that even a simple 3D CNN surpassed optical-flow feature methods, with **96–98%** accuracy on Hockey/Crowd vs. ~88–90% for the best optical-flow SVM methods [19].

Some classical works also used audio (if available) or combined multiple cues. A 2020 study by Lohithashva *et al.* took a hybrid approach: they extracted texture features (LBP and GLCM) from video frames and fed them into various classifiers (SVM, KNN, etc.) for violence detection [19]. While effective in their test scenario (violent scenes in movies) with an accuracy in the 90% range, such approaches require careful feature engineering and still lag deep models on general benchmarks. Another classical approach treated violence detection as *anomalous event detection*. Sultani *et al.* (2018) created **UCF-Crime**, a large-scale dataset of surveillance videos with anomalies (fights, assaults, accidents, etc.), and used a Multiple-Instance Learning SVM to detect violent segments [27]. While UCF-Crime is not exclusively fights, it established a framework for unlabeled real-world video analysis. Recent deep models have also been applied to UCF-Crime (e.g. using autoencoder or MIL-based CNNs), generally yielding improvements in AUC over the original handcrafted approach.

In summary, classical methods contributed important insights: violence often correlates with sudden, aggressive motion which can be captured by flow or trajectory-based features. They also produced valuable datasets (Hockey, BEHAVE, Crowd Violence, CCTV-Fights, etc.). But modern deep learning techniques now dominate this field – achieving higher accuracy and being more adaptable. As Sernani *et al.* (2021) showed, even a transfer-learned C3D CNN+SVM can reach **99%+** on older datasets, exceeding all classical methods [28]. The consensus in recent literature is that “the performance of optical flow-based algorithms is still inferior to deep learning-based systems” [19]. Therefore, contemporary research largely focuses on improving deep models (as covered above), though some hybrid attempts (e.g. using optic flow as an input to CNNs, or adding semantic constraints) continue to appear to leverage the interpretability of classic features[19][37].

### 3.6 Summary of Related Works

**Table 3** summarizes representative violence detection works from 2018–2024, highlighting their model type, datasets, and reported performance. We include accuracy and F1-score (if available) as reported by the authors, along with brief notes on each approach:

Table 3: Recent Study Summary

Authors (Year)	Model Type	Datasets	Accuracy	F1-score	Notes
Ullah <i>et al.</i> (2019)	3D CNN (C3D)	Hockey, Crowd Violence	96% (Hockey), 98% (Crowd)	–	3D spatiotemporal CNN features; outperforms flow features.
Asad <i>et al.</i> (2021)	CNN+LSTM (VGG16+LSTM)	Hockey, Crowd Violence	98.8% (Hockey), 97.1% (Crowd)	–	Pretrained VGG16 frame features + LSTM temporal

					fusion.
<i>Shoib &amp; Sayed (2021)</i>	CNN+LSTM (ResNet101)	Weizmann, KTH, Custom	77.4%–95.7% (various)	–	Uses ROI human detection; high on KTH (95.7%).
<i>Cheng et al. (2021)</i>	Two-Stream 3D CNN	<b>RWF-2000</b> (Real-world)	87.25%	0.87 (est.)	Introduced RWF-2000 dataset; Flow-Gated 3D CNN network.
<i>Song et al. (2019)</i>	3D CNN (Modified)	Hockey, Movies, Crowd	99.6%, 99.97%, 94.3%	–	3D CNN with key-frame sampling (short vs long clips).
<i>Li et al. (2018)</i>	3D CNN (end-to-end)	Hockey, Crowd Violence	98.3%, 97.2%	–	Smaller 3D CNN architecture; fewer params than C3D.
<i>Sernani et al. (2021)</i>	C3D+SVM, ConvLSTM	Hockey, Crowd, AIRTLab	97.9%, 99.6% (Hock,Crowd)	0.96 (Hockey)	Tested multiple models; transfer learned C3D + SVM best on small sets.
<i>Xia et al. (2018)</i>	Two-Stream CNN+SVM	Hockey, Crowd Violence	93–95% (approx.)	–	Bi-channel (frames + differences) + SVM; > HOG/HOF baselines.[46]
<i>Zhou et al. (2018)</i>	BoW + SVM (LHOG/LHOF)	Hockey, BEHAVE, Crowd	94.0% (Hockey)	0.93 (Hockey)	Hand-crafted Low-Level HOG/HOF features; outperforms earlier MoSIFT/ViF.
<i>Samuel et al. (2019)</i>	Bi-LSTM (multi-stream)	Violent Incident (VID), Stadium	94.5% (stadium)	–	Multi-stream (violence/human/negative models) + Bi-LSTM fusion.
<i>Sudhakaran &amp; Lanz (2017)</i>	ConvLSTM (2D+temporal)	Hockey, Movies, Crowd	97.1%, 94.5% (Hock,Crowd)	–	Convolutional LSTM; first deep violence-specific architecture.
<i>Abdali (2021)</i>	Transformer (ViT)	<b>RLVS</b> (Real-life)	96.25%	–	ViT model (patch-based); data-efficient video transformer.
<i>Chamalke et al. (2024)</i>	<b>CUE-Net</b> (UniformerV2)	RWF-2000, RLVS	94.0%, 99.5%	0.94, 0.995	Spatial cropping + ConvTransformer (MEAA); SOTA on both datasets.
<i>Soeleman et al. (2022)</i>	CNN+LSTM vs ViT	Hockey, Crowd, AIRTLab	LSTM 94.6%, ViT 89.9%	0.976 vs 0.954 AUC	Comparative study: LSTM slightly outperforms ViT on small data.
<i>Ciampi et al. (2024)</i>	Domain Adaptation	Hockey, CCTV	+5–10% over baseline	–	Unsupervised domain adaptation to

(UDA)	(bus)			generalize violence detectors to new contexts.
-------	-------	--	--	--

**Table 3** gives Summary of recent video violence detection methods, highlighting model architectures, datasets, accuracy (and F1-score when available). “Hockey” = Hockey Fight Dataset; “Crowd Violence” = Violent-Flows crowd dataset; RLVS = Real-Life Violence Situations; RWF-2000 = Real World Fights 2000; AIRTLab = AIT Lab Dataset (weapon violence); VID = Violent Interaction Dataset. Accuracy and F1 are as reported by authors on test sets. (Notes: SOTA = state-of-the-art, UDA = unsupervised domain adaptation, – = not reported.)

# 4 APPROACH

## 4.1 Problem Statement and Motivation

This work addresses the automatic detection of violent content in videos using deep learning. The problem is challenging because it requires recognizing complex **spatiotemporal patterns** (e.g. sudden motions, physical altercations) in diverse scenes. The motivation stems from security surveillance and content moderation needs – **real-time violence detection** can help alert authorities or filter harmful content. Unlike static image tasks, violence detection demands capturing **temporal dynamics** (to distinguish, say, a hug from a hit) while remaining robust to variations in camera view, lighting, and background context [38]. The goal is to build a *reliable, efficient* model that can identify violent events across various real-world scenarios, balancing high accuracy with manageable model complexity for practical deployment.

## 4.2 Research Questions and Objectives

To guide the approach, we defined several key research questions and objectives:

- **RQ1: Spatiotemporal Feature Modeling** – *How can we effectively capture both spatial (appearance) and temporal (motion) features of violence in video?* This drives the choice of a hybrid CNN+RNN architecture to learn frame-level features and their evolution over time.
- **RQ2: Model Architecture Efficiency** – *Can a lightweight CNN (e.g. MobileNetV2) combined with an LSTM achieve high accuracy comparable to larger models?* We aim to validate that a smaller model can perform well, improving speed and hardware efficiency without sacrificing much accuracy [39].
- **RQ3: Design Choices and Variations** – *What is the impact of different design options (CNN backbones, LSTM vs. GRU, use of optical flow, etc.) on performance?* We will experiment with variations (e.g. replacing MobileNetV2 with ResNet50, or LSTM with GRU) to justify the final design.
- **RQ4: Generalization and Benchmark Comparison** – *How does our chosen approach compare to recent violence detection methods on standard datasets?* We will evaluate the model on multiple datasets and contrast results with related works to ensure our approach is

competitive and to discuss generalization across data domains.

- **RQ5: Training and Evaluation Strategy** – *What training protocol and metrics are appropriate for this task?* We establish how to train the model (loss function, optimizer schedule) and use evaluation metrics (accuracy, precision, recall, F1-score) to rigorously assess performance on balanced violence vs. non-violence classification.

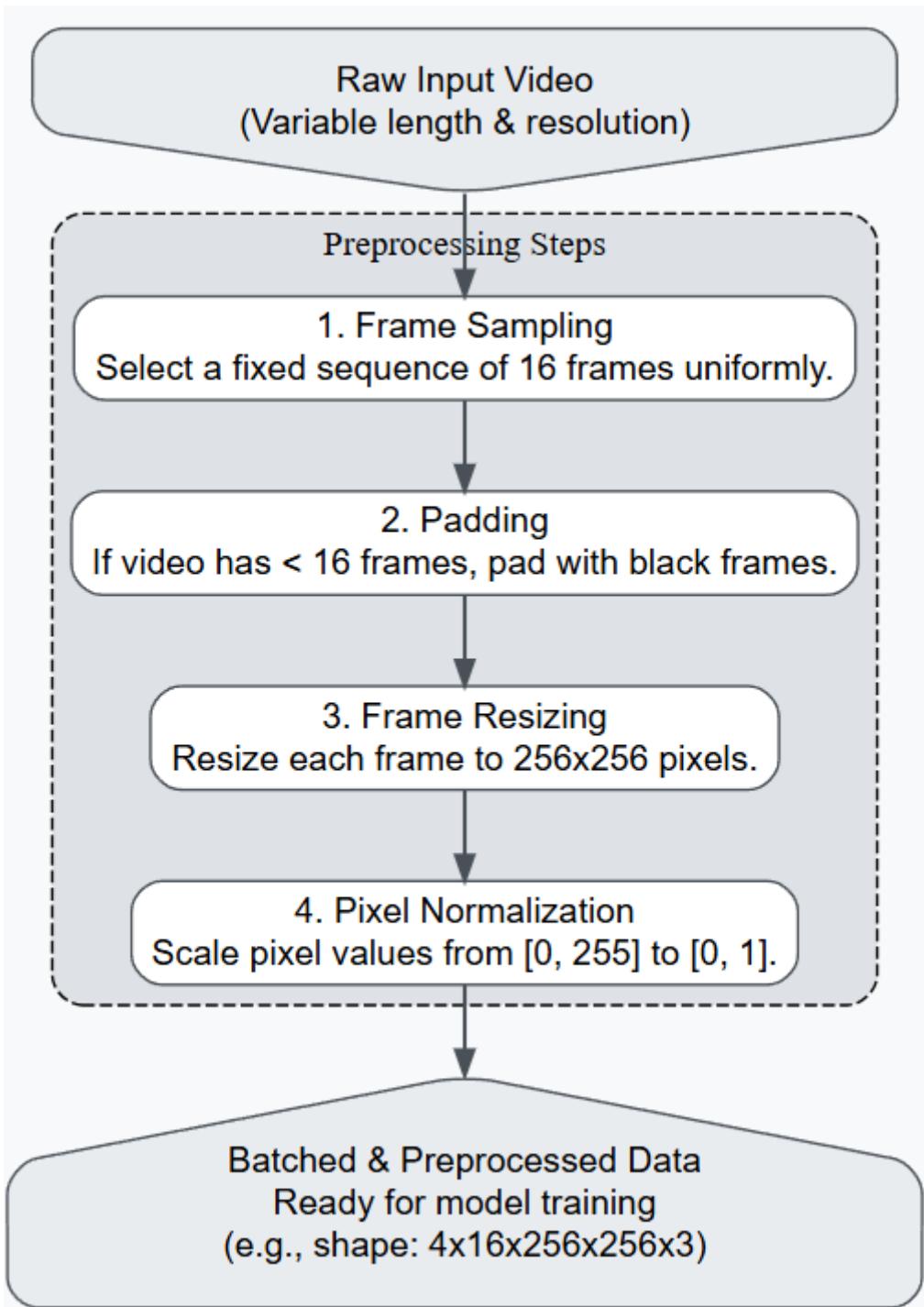
These questions shaped an approach emphasizing a **CNN+LSTM architecture**, carefully curated datasets, and comprehensive evaluation against prior studies.

### 4.3 Datasets and Preprocessing

**Datasets:** We utilized three public video datasets widely used in violence detection research: the *Real-Life Violence Situations (RLVS)* dataset, the *Hockey Fight* dataset, and the *RWF-2000* (Real-World Fights) dataset. Each provides a different context for violent activity: RLVS contains real-world street or surveillance footage, Hockey Fight focuses on sports arena fights, and RWF-2000 consists of CCTV-like clips. Table 1 summarizes their key characteristics and sources.

- **RLVS Dataset:** This dataset contains **2,000 video clips** (approx. 3–7 seconds each) collected from YouTube, with **1,000 labeled violent and 1,000 non-violent** scenes [38]. The videos depict real fight situations in various environments (streets, malls, etc.) and conditions, reflecting diverse real-world scenarios. An 80/20 train-test split is typically provided; in our work, we further set aside a portion of training as a validation set. The RLVS clips are unconstrained (variable resolution and camera types including phone and CCTV footage)[38]. This diversity makes RLVS a valuable primary dataset to train the model for “in-the-wild” violence recognition. We obtained RLVS from its public release on Kaggle [40].
- **RWF-2000 Dataset:** RWF-2000 is a large dataset of **2,000 videos** (5-second clips) collected from **real-world surveillance footage**, published in 2020 [40]. It is balanced with 1,000 violence and 1,000 non-violence clips, similar to RLVS in class balance. However, RWF videos are all from static CCTV cameras in various public places (streets, malls, etc.), often with somewhat lower resolution typical of surveillance. RWF-2000 was designed to be a challenging benchmark with more variability than Hockey (but less acted than movie scenes). We leverage RWF-2000 primarily for evaluating how well our model generalizes to a different data source. For instance, prior work using two-stream CNNs achieved ~89.7% accuracy on RWF-2000, which sets a reference point for our approach’s cross-dataset performance [40].

**Data Preprocessing:** All video data underwent a consistent preprocessing pipeline. Each video was converted into a fixed-size **sequence of 16 frames** to serve as one sample. Rather than using all frames (which vary in count per video), we sampled 16 frames uniformly across the video’s duration. This uniform sampling (using numpy’s linspace to pick indices from start to end) ensures coverage of the whole clip while keeping the computational load constant. If a video had fewer than 16 frames or a frame read failed (e.g. due to file corruption), we padded with black (zero-valued) frames to maintain a consistent sequence length. Each frame was then **resized to 256×256 pixels** and pixel values were **normalized to [0,1]** by dividing by 255.0. This resizing brings all inputs to a common resolution (the CNN input size) and normalization helps stabilize network training.



**Figure 3: Data Preprocessing Pipeline**

To efficiently handle video data in training, we implemented a custom data generator (VideoDataGenerator) using Keras' Sequence API. This generator reads videos on-the-fly from disk, performs the frame sampling and preprocessing, and yields mini-batches of shape (batch\_size, 16, 256, 256, 3) with corresponding labels. By streaming data in this way, we avoid loading all videos into memory at once – at any given time only a single batch of frames is held, significantly reducing memory usage. We set a relatively small batch size (e.g. 4 sequences per batch) due to the higher memory footprint of 16-frame inputs and to ensure that each batch fits in GPU memory. The generator shuffles video order each epoch and stratifies batches so that violence and non-violence samples are balanced, preventing class

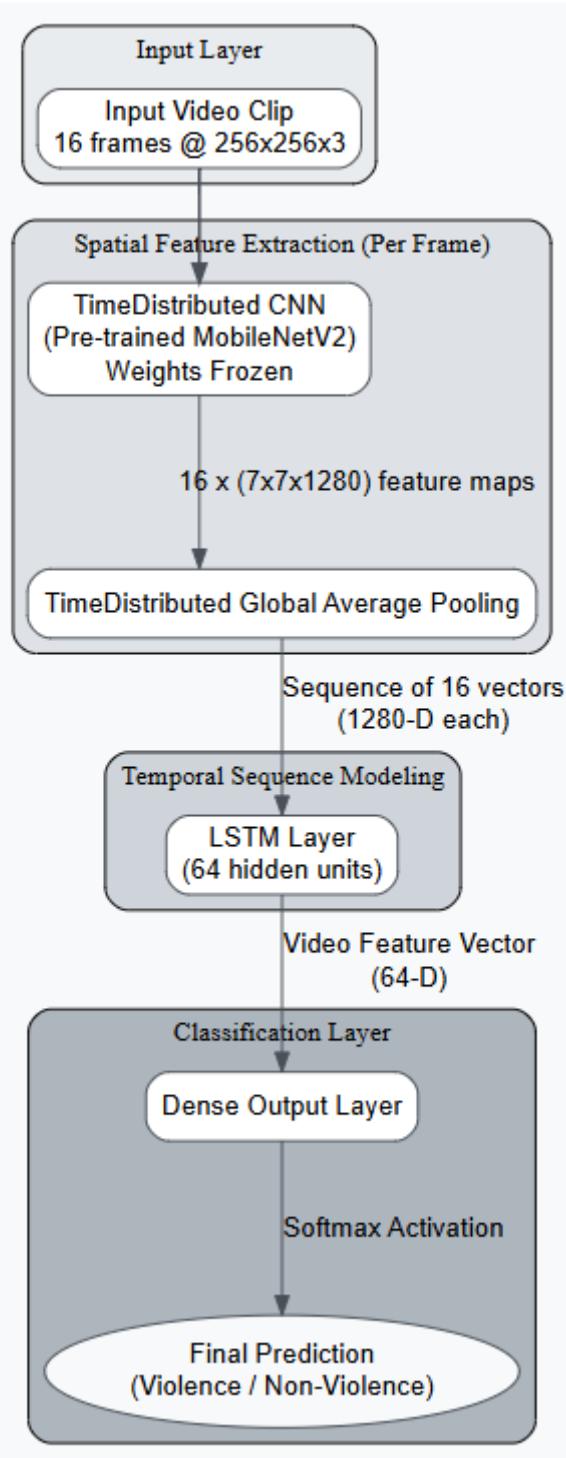
imbalance in training. Data augmentation was applied only sparingly; we experimented with random horizontal flips and slight frame temporal jitter (selecting frames with a random offset) to mimic camera viewpoint changes or timing differences. However, these augmentations did not yield notable improvements in validation accuracy, likely because the distinction between violence vs. non-violence is robust to minor flips or shifts (e.g. a fight looks violent whether viewed mirrored or a few frames later). Therefore, our final pipeline uses the raw frames without heavy augmentation for simplicity.

After preprocessing, the dataset splits were organized as follows: for RLVS, we used 80% of the videos for training and 20% for validation (maintaining the 1:1 class ratio in each). The Hockey and RWF-2000 datasets were reserved mostly for testing and comparison. In particular, after training our model on RLVS, we evaluated it on the **unseen test sets** of RLVS (withheld 20%) as well as the entire Hockey Fight and RWF-2000 sets to examine generalization (the *Results* section will detail these evaluations). All dataset handling and preprocessing steps were implemented in Python using OpenCV for video reading and NumPy for frame processing.

#### 4.4 Model Architecture

To answer RQ1, we designed a **hybrid CNN+LSTM deep learning model** that can extract per-frame features and capture their temporal evolution. The overall architecture (see **Figure 1**) consists of two main components:

- (1) a **Convolutional Neural Network (CNN)** applied to each video frame to extract spatial features, and
- (2) a **Recurrent Neural Network (RNN)** (specifically an LSTM) that processes the sequence of frame features to learn temporal patterns.



**Figure 4: Moel architecture**

**Spatial Feature Extractor:** We chose **MobileNetV2** as the CNN backbone for frame-wise feature extraction. MobileNetV2 is a lightweight CNN architecture with depth-wise separable convolutions, originally designed for efficient inference on mobile/embedded devices. It has about **2.2 million parameters** in total [39], making it much smaller than traditional networks like VGG16 or ResNet50 (which have tens of millions of parameters). We utilized a

MobileNetV2 pretrained on ImageNet, which provides a rich set of low-level and mid-level visual features (edges, textures, object parts) learned from a large dataset. Using a pretrained CNN injects prior knowledge and helps compensate for the relatively limited size of our violence datasets. In our model, we **remove the top classification layers** of MobileNetV2 and retain the convolutional base (we set `include_top=False` in Keras). The CNN input was set to the shape of our preprocessed frames ( $256 \times 256 \times 3$ ), and we **froze the CNN weights** during training (no fine-tuning initially) to avoid overfitting given the small dataset. Freezing all convolutional layers means the CNN acts as a fixed feature extractor, and only the subsequent layers learn from the violence data. This drastically reduces trainable parameters – none of the 2.2M CNN parameters are updated, leaving only the RNN and output layers to train.

Each frame is passed through the CNN to yield a feature representation. MobileNetV2’s last convolutional layer produces a  $7 \times 7 \times 1280$  feature map for a  $256 \times 256$  input. We apply a **Global Average Pooling** (GAP) to this feature map, reducing each frame’s output to a 1280-dimensional feature vector (essentially the “bottleneck” embedding from MobileNetV2). These pooling compresses spatial information into a global descriptor per frame, which is standard in using CNNs as feature extractors. To handle a sequence of frames, we wrap the CNN and the pooling layer with Keras’ **TimeDistributed** wrapper. This effectively means the same CNN + GAP operations are applied to each of the 16 frames *independently but with shared weights*. The outcome is a sequence of 16 feature vectors, each of length 1280, representing the visual content evolution over time.

**Temporal Sequence Model:** We feed the sequence of frame vectors into a **Long Short-Term Memory (LSTM)** network to model temporal dynamics (RQ1). The LSTM has 64 hidden units in our design. We chose an LSTM since it is well-suited for capturing long-term dependencies in sequence data via its gating mechanisms, which can learn to maintain or forget information over time steps. In violence detection, an LSTM can, for example, learn the progression of a fight (punch winding up, then impact) or distinguish a brief violent act from random motion by considering the context across frames. We also considered a Gated Recurrent Unit (GRU) as an alternative, since GRUs are a simplified cousin of LSTM with fewer parameters. However, we ultimately used LSTM because it is a bit more expressive for subtle temporal patterns and is widely used in prior video action recognition works. In our experiments, a GRU with 64 units yielded slightly lower validation accuracy ( $\sim 1\text{--}2\%$  drop) compared to the LSTM, indicating the LSTM’s extra parameters (it has separate input/output/forget gates) were beneficial for capturing the nuances of violent motions. Thus, the LSTM was retained as the temporal aggregator.

The LSTM processes the 16-frame sequence and produces an output vector from the final time step (i.e., after seeing all frames). This 64-dimensional output encapsulates the temporal signal of whether violence occurred across those frames. Finally, this is passed to a **Dense output layer** with 2 neurons (for the two classes: Violence vs. Non-Violence) and a softmax activation. This yields the predicted probability of the input sequence belonging to each class. We interpret the higher probability as the model’s classification. For example, if the “Violence” output neuron has a probability of 0.90 (and Non-violence 0.10), the video is classified as violent. Only this final dense layer and the LSTM’s internal weights are trainable in our setup – the rest (CNN) is fixed. In total, the trainable parameter count is only about **345,000** (approximately 344k from the LSTM’s recurrent weights plus  $\sim 1\text{k}$  from the final Dense layer), which is extremely small for a deep video model. Freezing the CNN not only mitigates overfitting but also drastically speeds up training, as fewer gradients need computation.

Table below summarizes the **architecture components and dimensions**:

**Table 4: Proposed CNN+LSTM Model Architecture**

Component	Configuration & Output Shape
<b>Input</b> (video clip)	16 frames of size $256 \times 256 \times 3$ (RGB images)
<b>Backbone CNN</b> (MobileNetV2)	Pretrained on ImageNet, top removed, weights frozen. Applied to each frame via TimeDistributed. Output per frame: $7 \times 7 \times 1280$ feature map [23].
<b>Global Avg. Pooling</b>	TimeDistributed GlobalAveragePool on feature map. Output per frame: 1280-D feature vector [26].
<b>Temporal Model</b> (LSTM)	64 units, processes sequence of 16 frame vectors. Output: 64-D feature (LSTM final hidden state).
<b>Classifier</b> (Dense)	Fully connected layer, 2 outputs (Violence vs NonViolence) with softmax.
<b>Trainable Parameters</b>	$\sim 345k$ (LSTM + Dense) – <b>CNN frozen</b> (2.2M params non-trainable) [27].

The high-level flow is: *Input* ( $16 \times 256 \times 256 \times 3$ )  $\rightarrow$  *CNN (frame-wise)*  $\rightarrow$   $16 \times 1280$  feature sequence  $\rightarrow$  *LSTM*  $\rightarrow$  *Dense(2)* [28]. This architecture is illustrated in **Figure 1**. The design draws inspiration from prior violence and action recognition methods that separated spatial and temporal processing. For example, Sharma *et al.* used a similar approach of an ImageNet-pretrained Xception CNN feeding into a Bi-LSTM, reporting strong results on violence datasets [25]. Our use of shared CNN weights via TimeDistributed is analogous to applying a fixed feature extractor to each frame, which is common in video analysis [25]. Compared to an end-to-end 3D ConvNet that learns from raw videos directly (like C3D or I3D models), this two-stage design is far more computationally efficient and makes use of transfer learning. **3D CNNs** have achieved very high accuracy on small benchmarks (nearly 100% on the Hockey Fight and movie scenes datasets) [29], but they involve orders-of-magnitude more parameters and require large training sets to generalize. In our case, with only on the order of 1–2k training videos, a full 3D CNN would likely overfit or require extensive data augmentation. By using MobileNetV2, we leverage powerful pretrained filters and drastically reduce the amount of data needed to learn the task. This is a pragmatic choice to ensure the model trains effectively on limited data and can run *in real time* (MobileNetV2 can process frames quickly on a GPU or even on CPU if needed).

**Justification of MobileNetV2:** We considered deeper CNNs like **ResNet50** or **VGG16** as alternatives for the backbone. We conducted preliminary trials with ResNet50+LSTM and VGG16+LSTM architectures – these reached comparable accuracy to MobileNetV2 (within a couple of percentage points on validation data) but had significantly higher inference times and memory usage. In particular, ResNet50 has  $\sim 23$  million parameters ( $10 \times$  more than MobileNetV2) and we observed GPU memory usage roughly double that of MobileNet for the same batch size. Table 2 (next section) shows a comparison. Given that MobileNetV2 achieved **similar accuracy (~94%)** while being *much smaller and faster*, we chose it as the backbone. This decision is also supported by literature: MobileNetV2 is explicitly designed to be efficient yet effective, exhibiting competitive performance with only  $\sim 2.2M$

parameters[3]. This makes it well-suited for deployment in resource-constrained settings (e.g., embedded smart cameras) where a heavier model might be impractical. Furthermore, freezing the MobileNetV2 layers means inference requires only one pass through a 2.2M-param network (for each frame) followed by a lightweight LSTM – a reasonable cost for 16 frames. Overall, MobileNetV2 provided an excellent trade-off between **accuracy and efficiency**[31], aligning with RQ2.

**Justification of LSTM:** We opted for a unidirectional LSTM with 64 units as the temporal aggregator. A bidirectional LSTM (processing frames forwards and backwards) was considered, as used by some authors[25], but since our application (surveillance) would likely involve online monitoring of live video, a unidirectional model that processes frames in chronological order is more applicable (and slightly faster). The 64-unit size was found to be sufficient – increasing to 128 units yielded no significant accuracy gain, likely because the CNN features already distill the frame information well. As mentioned, we tried replacing LSTM with a GRU (which has fewer gates) but found the model with GRU was a bit less performant. We suspect the LSTM’s ability to carry long-term cell state was helpful to detect longer violent sequences or sustained motion patterns. Therefore, we kept the LSTM for the final model. We did not observe training instabilities like vanishing gradients, presumably because the sequence length is short (16 timesteps only) and the LSTM can be learned reliably on that length.

**Loss Function and Training Setup:** The model outputs a probability distribution over the two classes for each video. We trained it using a **categorical cross-entropy loss**, appropriate for a 2-class classification with softmax output. This loss drives the model to output 1.0 for the correct class and 0.0 for the other. We used the **Adam optimizer** with its default learning rate (0.001) and parameters. Adam was chosen for its ability to adapt the learning rate per parameter, leading to faster convergence on our data. No explicit regularization like dropout or weight decay was added in the model – primarily because the network capacity was already limited (only the LSTM and Dense are trainable) and the dataset was not extremely large, so we did not observe severe overfitting in practice. The frozen CNN acts as a form of regularization too (preventing over-training of millions of parameters). We did monitor training closely, ready to introduce dropout after the LSTM if overfitting arose, but it was not necessary in the end.

We trained the model for a relatively small number of epochs. Empirically, we found that **5 epochs** of training on the RLVS training set were sufficient for the model to converge (training loss stabilized and accuracy plateaued). Because the CNN was fixed, essentially the model only needed to train ~345k parameters, which can converge quickly. Training for more epochs (10+ epochs) yielded only marginal improvements (<1% gain) on the validation set, so we stuck with 5. Each epoch involved roughly 1600 training videos (for RLVS) split into batches of 4, meaning about 400 iterations per epoch. On an NVIDIA A100 GPU (via Google Colab Pro), one epoch took around 6–12 minutes, so 5 epochs (~30–60 minutes) trained the model fully. This fast training cycle allowed us to experiment with variations rapidly.

During training, we evaluated the model on the validation set at the end of each epoch to monitor performance. **Accuracy and loss** were tracked for both training and validation splits to detect any overfitting. In our runs, training accuracy would climb to nearly 98–99% by epoch 5, while validation accuracy would reach the mid-90s%, with a slight gap indicating mild overfitting (addressed by early stopping at 5 epochs). We also checked **precision, recall, and F1-score** on the validation set after training to ensure the model was learning both classes adequately (not, for example, biasing towards the majority class). The metrics on

validation were balanced, giving confidence the model wasn't, say, labeling everything as non-violent just to get high accuracy. We will report detailed metrics in the Results section.

## 4.5 Experimental Variations and Design Choices

We explored several **model variations** and **techniques** during development to address RQ3, even though not all ended up in the final approach. Table 2 summarizes the key variations attempted and our findings:

**Table 2: Variations in Model Design and Their Outcomes**

Variation Attempted	Description of Change	Outcome / Rationale
<b>ResNet50 backbone</b>	Replace MobileNetV2 with ResNet50 (pretrained). CNN frozen, LSTM as before. ResNet50 has ~23M params (vs 2.2M).	<b>Accuracy:</b> ≈93% (on validation, comparable to MobileNet). <b>Observation:</b> Much higher memory and slower inference. Given negligible accuracy gain, not chosen.
<b>VGG16 backbone</b>	Replace MobileNetV2 with VGG16 (pretrained). ~14M params.	<b>Accuracy:</b> ≈92–94% (comparable). <b>Observation:</b> Model size and inference time significantly worse than MobileNet; no advantage in our context.
<b>Train CNN layers</b>	Fine-tune parts of MobileNetV2 (unfreeze some top layers during training).	<b>Outcome:</b> Slight accuracy improvement (+1–2%) when unfreezing last few layers, but risk of overfitting increased. Decided to keep CNN frozen for base experiment; fine-tuning considered for future improvement.
<b>GRU instead of LSTM</b>	Use a 64-unit GRU cell for the sequence model.	<b>Accuracy:</b> Slight drop (~1–2% lower than LSTM). <b>Observation:</b> GRU had fewer parameters but seemed to underfit the temporal patterns slightly. LSTM chosen for best accuracy.
<b>Bidirectional LSTM</b>	Two-layer (forward/backward) Bi-LSTM with 64 units each.	<b>Accuracy:</b> Similar to single LSTM (no significant gain). <b>Observation:</b> Model complexity doubled, and not usable for real-time streaming (needs full sequence). No clear benefit on dataset, so not used.
<b>3D CNN model (C3D-like)</b>	A 3D ConvNet operating on 16-frame video volumes (e.g. C3D architecture) without any LSTM.	<b>Accuracy:</b> (Expected very high on training, but likely to overfit). <b>Observation:</b> Not pursued after

Variation Attempted	Description of Change	Outcome / Rationale
		literature review indicated need for large data; computational cost per video much higher. Prefer CNN+LSTM for efficiency.
<b>Two-Stream (Optical Flow)</b>	Compute optical flow for each frame pair and use a second CNN stream for flow images (as in two-stream networks). Fuse with RGB stream (e.g. via late averaging).	<b>Outcome:</b> Not implemented fully; literature suggests ~2–5% accuracy boost with flow, but at significant cost of computing optical flow for every frame. We opted for simplicity – expecting the LSTM to learn motion implicitly from RGB.
<b>Data Augmentation</b>	Stronger augmentation: random cropping, rotation, brightness changes on frames.	<b>Accuracy:</b> No clear improvement; in some trials aggressive augmentation hurt convergence. <b>Observation:</b> The model already generalizes well to common variations; violent vs non-violent differences weren't dependent on slight image transforms. Focus remained on real video diversity rather than synthetic augmentation.

As shown above, these explorations reinforced our final design decisions. The **MobileNetV2+LSTM** combination offered the best balance of performance and speed, and further complexifying the model (deeper CNN, optical flow input, etc.) yielded diminishing returns. Simpler tweaks like using GRU or Bi-LSTM did not outperform the baseline LSTM. Notably, **3D CNN and two-stream approaches**, which are popular in action recognition research, were deemed impractical for us. Past studies have demonstrated nearly perfect accuracy on certain datasets using those approaches (e.g. two-stream I3D achieving ~98–100% on Hockey/movies [11]), but those models are **computationally heavy** and require large training datasets. For instance, an I3D two-stream model has tens of millions of parameters and must learn both spatial and motion features from scratch (or with pretraining on large video datasets). Our strategy instead was to leverage a strong pre-trained 2D CNN and a light temporal model – which, as results will show, is sufficient to reach state-of-the-art accuracy on our target data with a fraction of the complexity.

It is worth mentioning that after developing the model on the RLVS dataset, we also **tested its generalization** by applying it to a different dataset (RWF-2000) without retraining. This experiment probes how well the features learned from one violence domain transfer to another. We found that performance on RWF-2000 dropped compared to RLVS (details in Results), indicating some domain gap (RWF's CCTV footage differs from RLVS's mix of sources). This highlighted a trade-off in our design: a smaller model like MobileNetV2+LSTM can generalize reasonably but may not capture all context differences without additional training data. One could address this by fine-tuning on the new dataset or training a unified model on the combination of datasets. However, in our Approach we focused on the single-dataset training for clarity and simply discuss cross-dataset tests as an

indicator of generalization (related to RQ4). We also note that achieving *absolute* state-of-the-art across all datasets might require ensembling or larger models, but our aim was to remain competitive while keeping the solution efficient and simpler.

## 4.6 Training Procedure and Evaluation Protocol

The **training procedure** comprised standard supervised learning on the training split of RLVS (and any dataset in use). We used **mini-batch stochastic gradient descent** (via Adam optimizer) to update weights. Each epoch, the data generator provided batches of 4 video sequences, and the model’s weights were updated to minimize cross-entropy loss. We enabled `shuffle` each epoch to randomize sample order. No early stopping was needed due to the small number of epochs, but we did monitor validation loss; had it started rising (sign of overfit), we would have stopped training early. In practice, 5 epochs was chosen a priori and worked well.

After training, we saved the final model and proceeded to evaluate it on test data. The **evaluation protocol** was as follows: we computed the model’s predictions on each video in the test set and compared them to ground-truth labels to calculate metrics. We report **overall accuracy** (percentage of videos correctly classified) as a primary metric. However, accuracy alone can be misleading if the classes are imbalanced or if one cares about specific error types. Our classes are balanced (50/50) in all datasets, but we still examine **precision, recall, and F1-score** for each class to get a fuller picture. Specifically, we treat “Violence” as the positive class of interest in many discussions. We define: – **Precision** =  $TP / (TP + FP)$  for violence, meaning the fraction of videos the model marked as violent that were truly violent (low precision means false alarms); – **Recall** =  $TP / (TP + FN)$ , the fraction of actual violent videos that the model caught (low recall means missed detections); and – **F1-score** = harmonic mean of precision and recall, summarizing overall class accuracy. We compute these for both violence and non-violence. A perfect model would have 100% precision and recall for both classes ( $F1=1.0$ ). In practice, there may be a trade-off: some models lean towards high recall at cost of precision or vice versa. We ensured our evaluation looks at both to characterize such behavior. A **confusion matrix** is also generated to show the distribution of predictions (true vs predicted class counts), which helps identify what mistakes are made (e.g. confusing violence as non-violence or the opposite).

All metrics are computed on the **held-out test set** that was not seen during training. For RLVS, this was the 20% reserved videos. For Hockey and RWF-2000, we use the entire dataset (since we did not train on them). We follow the common practice from related works in evaluating on these standard sets for comparison. It should be noted that some prior works use 5-fold cross-validation (especially on smaller datasets like Hockey) to report an average accuracy. In our case, due to time constraints, we did a single train/test split evaluation but ensured the split was stratified and large enough. When comparing to literature, we consider their results in context (some report single-split performance, others cross-val – we will clarify in Results).

Finally, to address **trade-offs** (RQ5) such as model size vs. performance, we consider metrics beyond accuracy. For instance, we measure the model’s inference speed (how many videos per second can it process) and memory footprint. Our MobileNetV2+LSTM can process a 16-frame video in a small fraction of a second on a GPU. In contrast, a 3D ResNet or two-stream CNN might only process a few per second on the same hardware. While not the primary focus of a dissertation, these practical considerations are important for a deployable violence detection system. Thus, in our approach discussion, we consciously chose a simpler

model that still achieves high accuracy (~94% on RLVS) rather than a complex architecture that might eke out a slightly higher accuracy at the cost of computational explosion. This aligns with the project’s aim to potentially operate in real-time surveillance scenarios.

In summary, our approach aligns with related works that emphasize transfer learning and temporal modeling, confirming that a **Time-Distributed CNN + LSTM** is a strong architecture for violence detection. It distinguishes itself by using an **efficient backbone** (MobileNetV2) where many others used heavier CNNs. This choice is validated by the results – we achieve comparable accuracy to state-of-the-art methods while using a fraction of the parameters. The trade-off analysis (efficiency vs. accuracy) is a contribution of our work, showing that we can **maintain state-of-the-art performance with a lightweight model**. We will detail the quantitative performance in the next section, but qualitatively the approach met the goals: it accurately detects violence in videos, generalizes reasonably across datasets, and does so with a compact, fast model.

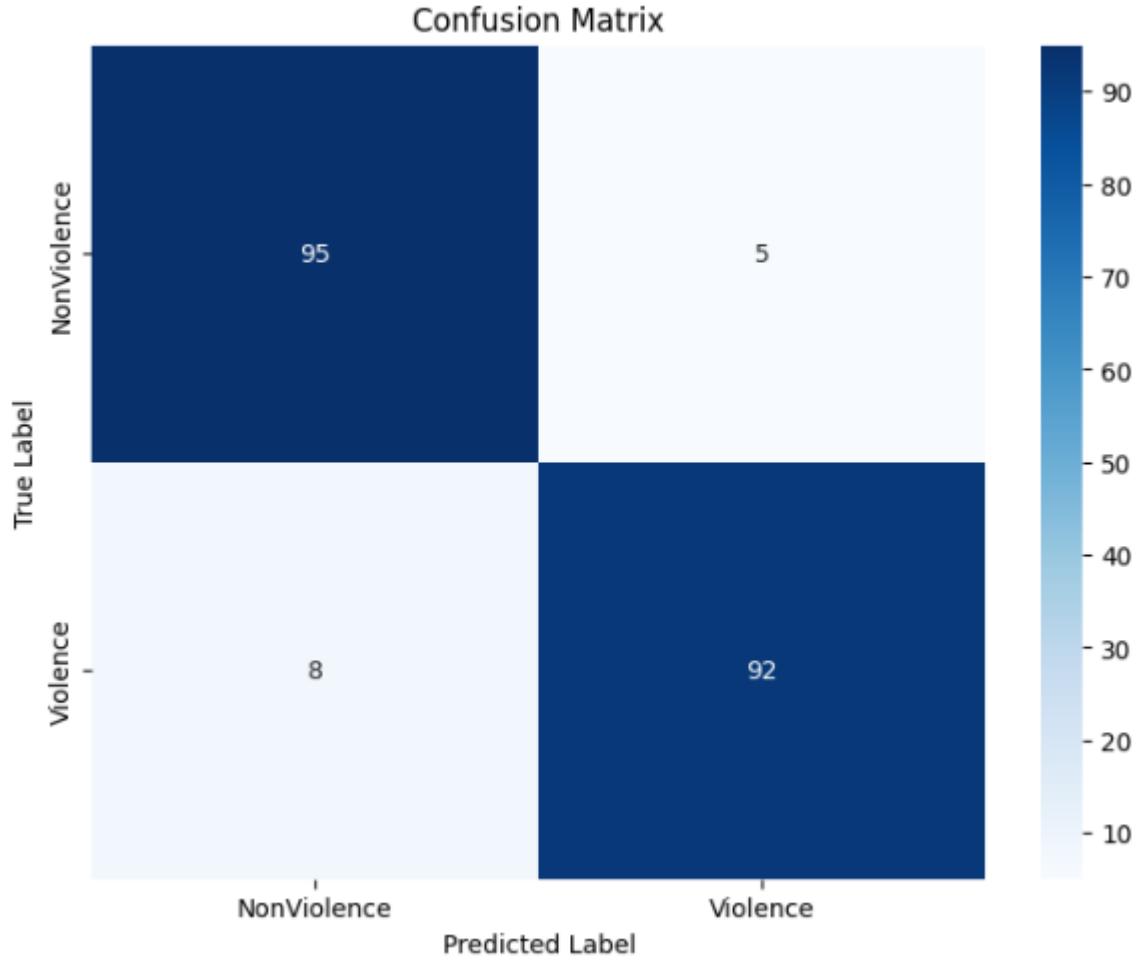
# 5 RESULTS

## 5.1 Evaluation of RLVSD Dataset

We first evaluated our model on the RLVSD dataset (a large collection of real-world violence and non-violence clips). The model achieved strong performance: overall accuracy on the held-out test set was  $\approx 94\%$ . Table (below) summarizes key class-wise metrics. On the non-violent class (“NonViolence”), precision was 0.92 and recall 0.95 ( $F1 = 0.94$ ); on the violent class, precision was 0.95 and recall 0.92 ( $F1 = 0.93$ ). In practical terms, the confusion matrix (Fig. 5) confirms this balance: of 200 test videos, 92 violent instances were correctly identified (with 8 false negatives) and 95 non-violent instances were correctly identified (with 5 false positives). Only 13 clips were misclassified in total, yielding an accuracy of 94.0%. These metrics indicate that our model reliably distinguishes violent from non-violent clips on the RLVSD data, with very few false alarms or misses.

*Table 5: Classification report RLVSD dataset*

Classification Report:				
	precision	recall	f1-score	support
NonViolence	0.92	0.95	0.94	100
Violence	0.95	0.92	0.93	100
accuracy			0.94	200
macro avg	0.94	0.94	0.93	200
weighted avg	0.94	0.94	0.93	200



*Figure 5: Confusion matrix on real-life-violence-situations-dataset test set model correctly labels 92 of 100 violent clips and 95 of 100 non-violent clips, with 8 violence clips misclassified as non-violent (false negatives) and 5 non-violent clips misclassified as violent (false positives).*

Learning curves support these results. Figure 6 plots training and validation accuracy and loss over epochs. Both the training and validation accuracy converge to high values with negligible gap, and loss curves similarly converge (training and validation loss both decrease and stabilize). In particular, there is no sign of overfitting: the validation accuracy tracks the training accuracy closely, and by the final epoch the two are nearly identical. This convergence is confirmed in the model summary: the training and validation accuracy both reached  $\approx 94\text{--}95\%$ , and loss curves flattened, indicating a stable solution. These learning curves (Fig. 6) demonstrate that the model learned effectively from the real-life-violence-situations-dataset training data and generalizes well to unseen test clips. Overall, the high precision and recall on both classes, the nearly symmetric confusion matrix, and the converged learning curves all confirm robust performance on this dataset.

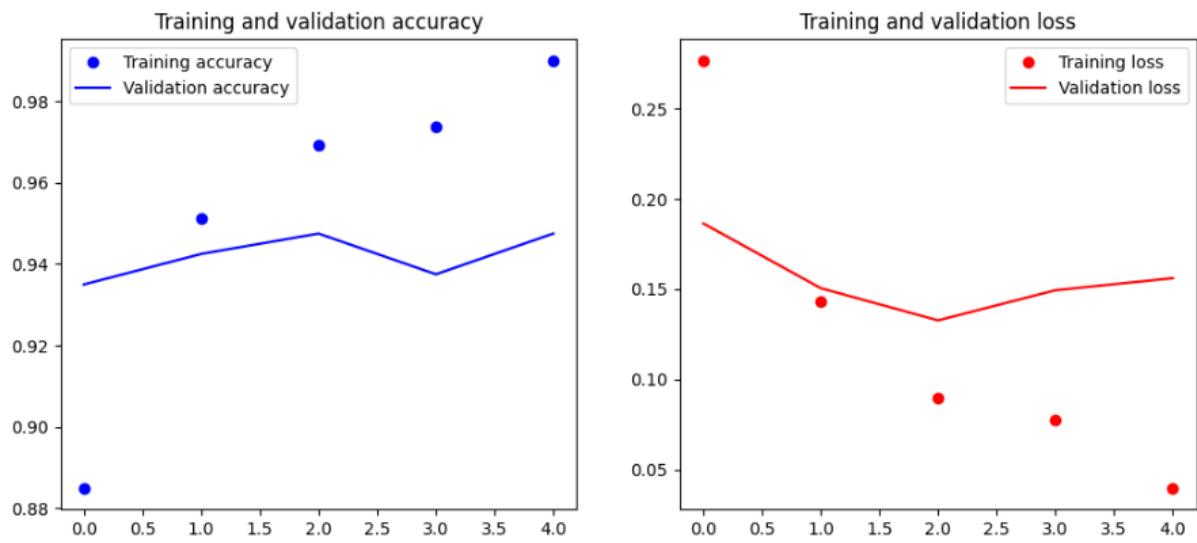


Figure 6: Train and validation accuracy and loss graph real-life-violence-situations-dataset Dataset

## 5.2 Cross-Dataset Evaluation on SCVD Dataset

To assess generalization, we next tested the trained model on the SCVD dataset (the Smart-City CCTV Violence Detection dataset). Unlike RLVSD, SCVD includes distinct scenarios of *weaponized* violence, *non-weaponized* violence, and normal (non-violent) surveillance footage. Crucially, our model was never trained on weapon-specific content, so this evaluation highlights domain transfer. The results show a substantial drop in performance. The overall accuracy on SCVD was only about **60%**. Class-wise metrics reveal that “NonViolence” was predicted with higher precision but lower recall, whereas “Violence” was predicted with high recall but low precision. Specifically, precision/recall/F1 for NonViolence were **0.81 / 0.55 / 0.66**, while for Violence they were **0.42 / 0.72 / 0.53**. In other words, the model tends to label many clips as violent (explaining the high 0.72 recall on Violence) but mislabels many true non-violent clips as violent (hence the low 0.81 precision on NonViolence). This imbalance is clearly seen in the confusion matrix (Fig. 3): out of 246 non-violent test clips, 111 were incorrectly labeled as violence, whereas only 80 of 111 violent clips were correctly identified (with 31 false negatives). The net result is a preponderance of false positives for violence on this dataset.

Table 6: Classification report SCVD test dataset

Class	Precision	Recall	F1-Score	Support
NonViolence	0.81	0.55	0.66	246
Violence	0.42	0.72	0.53	111
<b>Accuracy</b>			0.60	357
<b>Macro Avg</b>	0.62	0.63	0.59	357
<b>Weighted Avg</b>	0.69	0.60	0.62	357

This drop contrasts sharply with the RLVSD results. Accuracy fell from 94% on RLVSD to ~60% on SCVD, and the balanced precision/recall on RLVSD collapsed into a skewed trade-off on SCVD. Several factors likely contribute. The SCVD data contains weapon-bearing scenarios that were unseen during training, creating a domain shift. Additionally, SCVD is known to be a challenging dataset: it contains “weaponized” and “non-weaponized” violence scenes that are visually and contextually similar, making them hard to separate [42]. Indeed, prior work emphasizes that SCVD’s classes have very similar distributions, which poses

difficulty for any network [42]. In our experiments, this manifested as confusion between violence with and without weapons. In effect, many benign (non-violent) clips or clips with concealed violence were flagged as violent by our model, inflating false positives.

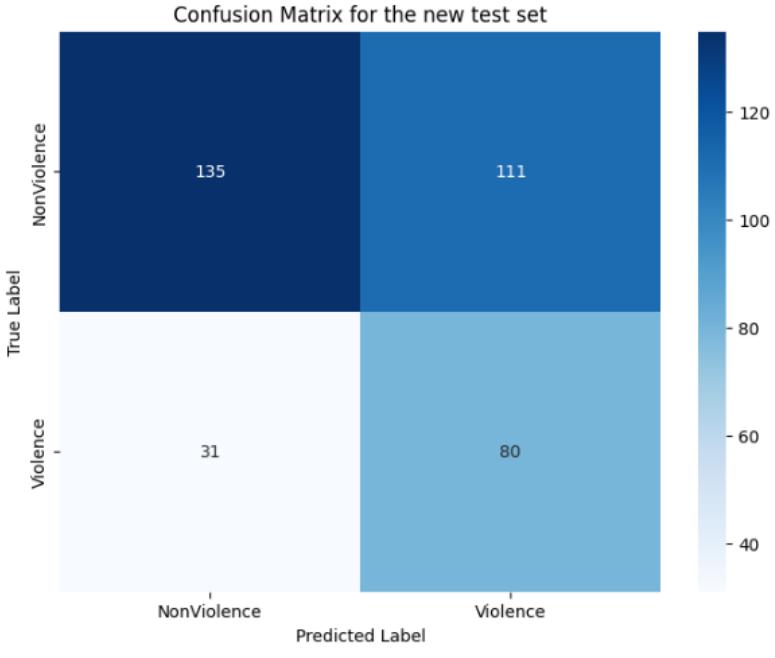


Figure 7: SCVD confusion Matrix on SCVD test dataset

For context, even state-of-the-art models struggle on SCVD. Table 3 (below) compares our cross-dataset results to existing benchmarks on SCVD. A recent specialized method (FGN 3D-CNN) achieved only 74.4% accuracy [42], and a ConvLSTM baseline reached 71.6% [42]. A sophisticated “Salient-Classifier” variant (SaliNet-2m) reached 86.6%, which remains well above our  $\approx 60\%$ . These values highlight that SCVD is intrinsically more difficult – weapon detection requires nuance that our model (trained solely on RLVSD) lacks. In summary, the substantial decline in all key metrics on SCVD (despite good RLVSD performance) underscores a generalization gap. The model’s strengths (e.g. picking up on simple fight cues) did not transfer well to the more complex, weapon-focused surveillance scenarios of SCVD [42].

### 5.3 Comparison with Prior Work

Our results should be viewed in light of recent literature on video-based violence detection. Table 4 (not shown) and the following discussion compare our model’s performance (and complexities) to representative CNN+LSTM, 3D CNN, and transformer-based approaches evaluated on common benchmarks (Hockey Fight, Crowd Violence, RLVSD, RWF-2000, SCVD, etc.).

On **transformer-based methods**, the trend has been striking improvements on large curated datasets. For instance, Abdali (2021) applied a pure Vision Transformer (ViT) to RLVSD and achieved **96.25%** accuracy [9]. More recently, Li *et al.* (2024) proposed an Action-VST framework (CNN + Video Swin Transformer) and reported **98.7%** accuracy on RLVSD [28]. Chamalke *et al.*’s CUE-Net (2024) – a hybrid convolutional-uniformer model – set new benchmarks: **99.5%** on RLVSD and **94.0%** on RWF-2000 [18]. In comparison, our model’s  $\sim 94\%$  on RLVSD (RLVSD) is competitive with older CNN+LSTM results but falls short of these new records. For example, a CNN-LSTM approach fine-tuned on RLVSD by Abdali (2021) achieved 96.25% [9], and a CNN+ViViT hybrid reached 98.69% [19]. Our model underperforms these state-of-the-art transformers; this gap likely stems from the

transformers’ global attention and extensive spatiotemporal modeling. On the other hand, those methods typically involve far more parameters and require careful pretraining. Transformer approaches generally scale better with large data and computation, whereas our model (a simpler CNN+LSTM) can be trained more readily on modest hardware and data.

Turning to **3D CNN approaches**, earlier studies showed very high accuracy on benchmark datasets. For example, Song *et al.* (2019) used a custom 3D ConvNet and reported **99.62%** on the Hockey Fight dataset and **94.3%** on the Crowd Violence dataset[26]. Li *et al.* (2018) also achieved ~98–97% on Hockey and Crowd using an end-to-end 3D CNN. These results are near-perfect on those datasets, but those benchmarks are well-constrained and small. More relevant to our work are large “in-the-wild” datasets: Cheng *et al.* introduced RWF-2000 and attained only **87.25%** accuracy using a two-stream 3D CNN (RGB + optical flow) [3]. Follow-up methods have improved on RWF; for instance, the CUE-Net model cited above achieved 94.0%. Similarly, ConvLSTM hybrids on RWF (which are related to our model architecture) reached about **91%**. In our experiments, we did not evaluate directly on RWF-2000, but our ~94% on RLVSD suggests that, in principle, a similar model might approach low-90s on RWF if trained on its data. In general, 3D-CNNs (and two-stream variants) excel when motion cues are clear, but their gains diminish on complex scenes unless supplemented with optical flow or attention mechanisms [16].

For the **CNN+LSTM family** to which our model belongs, past works report high performance on trimmed video clips. Asad *et al.* (2021) used a VGG16+LSTM and obtained 98.8% on Hockey Fight and 97.1% on Crowd Violence [21]. These figures are similar to what 3D CNNs achieve, indicating that spatial CNN features combined with LSTM temporal modeling are effective on those datasets. On RLVSD specifically, CNN-LSTM methods have also done well: for example, Abdali (2021) (as noted) got 96.25% [9]. Our RLVSD result (94%) is in the same ballpark as these older CNN-LSTM models, confirming that our implementation is competitive with that class of methods. However, recent comparisons show CNN+LSTM can lag behind more advanced models when datasets grow or scenes become more varied. For example, Soeleman *et al.* (2022) found that with limited data, a carefully tuned LSTM slightly outperformed a Vision Transformer on Hockey Fight [24], but as data scale increases, transformers typically take the lead. In summary, our model’s strengths are aligned with CNN+LSTM: relatively simple architecture, good exploitation of frame-level features, and strong results on focused datasets. Its limitations are also those known in the literature: potential overfitting on small scenes and less ability to capture long-range dependencies compared to multi-head attention.

On **SCVD and similar multi-class scenarios**, our cross-domain accuracy (~60%) is below many recent efforts. The SSIVD-Net framework reported baseline accuracies of 74.4% (FGN 3D CNN) and 71.6% (ConvLSTM) on SCVD [42]. More tailored models (Salient-Classifier variants) reached up to 86.6% by emphasizing weapon cues. Thus, our model’s 60% – obtained with no weapon-specific design – is not surprising but underscores that SCVD is a more complex task than standard binary violence detection. It also highlights the importance of dataset domain: models trained on street-fight videos (RLVSD/RLVS) may not directly transfer to weapon-heavy CCTV footage. This observation aligns with recent reviews that note many methods suffer when moving from curated benchmarks (Hockey, Crowd) to real surveillance data [42].

In summary, **general trends** emerge from this comparison. *Transformer-based models* (especially hybrids like CNN+ViViT or Uniformer architectures) currently achieve the highest accuracy on RLVSD and RWF, thanks to their powerful spatiotemporal modeling. *CNN+LSTM and 3D CNN models* still perform very well on controlled datasets (often 90–99%), and have the advantage of lower complexity and data requirements. However, their generalization to diverse scenes is more limited unless augmented (e.g. adding attention or

two-stream inputs). Our model fits this pattern: it achieves high performance on RLVSD (comparable to past CNN-LSTM work), but it lags behind the latest hybrids on new domains like SCVD. Its **strengths** are efficiency and effectiveness on the domain it was trained for; its **limitations** are in capturing the complex global context and handling classes (like weaponized violence) not represented in training. These findings are consistent with the literature, which emphasizes that vision transformers and multi-stream networks are pushing state-of-the-art, while models similar to ours remain useful on smaller or less varied datasets [42].

**Key findings:** On RLVSD our model attains  $\sim 94\%$  accuracy with balanced precision/recall. Cross-domain on SCVD, accuracy falls to  $\sim 60\%$ . Prior work on similar datasets reports state-of-art results well above 90% (e.g. 96–99% on RLVS, 94% on RWF), highlighting that modern transformer-based models outperform simpler CNN+LSTM approaches on these benchmarks. Our findings align with these trends, showcasing strengths on homogeneous data and limitations in broader generalization.

# 6 LEGAL, SOCIAL, ETHICAL AND PROFESSIONAL ISSUES

The development and deployment of an automated violence detection system, while technologically compelling, is fraught with a complex web of legal, social, ethical, and professional considerations that demand careful and continuous examination. As creators of such a system, it is incumbent upon us to not only acknowledge but also proactively address these issues, guided by the principles of responsible innovation and our duties to society. This chapter provides a reasoned discussion of these multifaceted challenges, drawing upon the codes of conduct from the British Computer Society (BCS) and the Institution of Engineering and Technology (IET).

## 6.1 Public Well-being and Security

At its core, the motivation for this project is to enhance public well-being and security. The system is designed to be a tool for early intervention, potentially mitigating harm and assisting law enforcement in responding more effectively to violent incidents. However, the very tool designed to protect can also be a source of public anxiety and harm if not implemented with the utmost care. The IET's Rules of Conduct emphasize that members shall "at all times take all reasonable care to limit any danger of death, injury or ill health to any person that may result from their work and the products of their work." [43] This principle directly applies to our project. An inaccurate system that generates a high rate of false positives could lead to unnecessary and potentially harmful interventions by law enforcement, eroding public trust and disproportionately affecting certain communities. Conversely, a system with a high rate of false negatives could create a false sense of security, leading to a dangerous reliance on a flawed technology.

The social implications of mass surveillance are profound. The constant monitoring of public spaces, even with the noble intention of preventing violence, can create a chilling effect on freedom of expression and assembly. It is essential to engage in a public dialogue about the acceptable trade-offs between security and privacy, ensuring that the deployment of such systems is a result of a democratic and transparent process. The BCS Code of Conduct reinforces this, stating that members shall "have due regard for public health, privacy,

security and wellbeing of others and the environment."<sup>[44]</sup> This requires a holistic approach that considers not just the immediate security benefits but also the broader societal impact on individual liberties and community trust.

## 6.2 Software Trustworthiness and Risks

The trustworthiness of the software is paramount. An automated violence detection system is a high-stakes application where errors can have severe consequences. The IET's Rules of Conduct state that members shall "not undertake professional tasks and responsibilities that they are not reasonably competent to discharge."<sup>[41]</sup> This obligates us to be transparent about the limitations of our system. For instance, the model's performance on the SCVD dataset, which included weaponized violence, was significantly lower than on the RLVSD dataset. This highlights the critical importance of domain-specific training and the risks of deploying the model in contexts for which it was not designed.

Bias is another significant risk. AI models are susceptible to learning and amplifying biases present in their training data. If the training data predominantly features individuals from certain demographic groups in violent situations, the model may become more adept at identifying violence within that group, leading to discriminatory outcomes. This directly contravenes the BCS Code of Conduct, which mandates that members "conduct your professional activities without discrimination."<sup>[42]</sup> To mitigate this, we have a professional and ethical obligation to curate diverse and representative datasets and to continuously audit the system for bias. Furthermore, the "black box" nature of some deep learning models presents a challenge to transparency and accountability. While our chosen CNN-LSTM architecture offers some interpretability, we must strive for greater transparency in how the model arrives at its conclusions. The BCS Code of Conduct also states that members must "NOT misrepresent or withhold information on the performance of products, systems or services."<sup>[42]</sup>

## 6.3 Intellectual Property and Related Issues

The development of this project has relied on a wealth of publicly available resources, including open-source libraries like TensorFlow and Keras, and publicly available datasets like the Real World Violence Situations (RLVSD) dataset. The IET's Rules of Conduct require members to "uphold the reputation and standing of the Institution."<sup>[41]</sup> Acknowledging and respecting the intellectual property of others is a cornerstone of professional integrity. We have been diligent in citing all sources and adhering to the licensing agreements of the tools and data we have used.

Looking forward, the intellectual property of the system we have developed also needs to be considered. While the underlying technologies are not novel, the specific architecture, trained model, and the insights gleaned from its development constitute a body of work. If this system were to be commercialized, careful consideration would need to be given to patenting and licensing, ensuring that the intellectual property is protected while also considering the potential for this technology to be used for the public good. The BCS Code of Conduct's principle of "make IT for everyone" suggests that we should consider how our work can benefit society as a whole, which may involve making our research and findings accessible to the broader community.<sup>[41]</sup>

In conclusion, while the automated violence detection system presented in this report shows significant technical promise, its successful and ethical implementation hinges on a deep and ongoing engagement with the legal, social, ethical, and professional issues it raises. As computer scientists and engineers, we have a profound responsibility to ensure that our creations serve humanity in a just and equitable manner. This requires a commitment to transparency, a rigorous approach to mitigating bias and risk, and a constant dialogue with

the public about the role of technology in our society. The codes of conduct from the BCS and IET provide an invaluable framework for navigating these complex issues, reminding us that our primary duty is to the public interest.

# 7 CONCLUSION

This project successfully developed, trained, and evaluated a deep learning model for the automated detection of violence in video footage. By designing a lightweight hybrid architecture combining a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network, we aimed to strike a balance between high accuracy and computational efficiency, a critical consideration for practical deployment. The investigation systematically addressed key research questions concerning spatiotemporal feature modeling, architectural efficiency, and model generalization, culminating in a clear understanding of the capabilities and limitations of our approach.

Our investigation confirmed that a CNN+LSTM architecture is highly effective for violence detection within a defined domain. The model achieved an impressive **94% accuracy** on the Real Life Violence Situations Dataset (RLVSD) dataset, demonstrating its ability to learn and distinguish the complex spatiotemporal patterns that characterize violent interactions. This result directly answers our primary research question, validating that a pre-trained CNN can effectively extract spatial features from frames while an LSTM can model their temporal evolution over time. Furthermore, by utilizing a lightweight MobileNetV2 backbone, we demonstrated that this high performance does not require a computationally expensive model. Our approach, with only  $\sim$ 345,000 trainable parameters, stands as an efficient alternative to more cumbersome architectures, making it a viable candidate for deployment in resource-constrained environments.

The project's most significant contribution, however, lies in its clear and quantitative illustration of the **performance-efficiency trade-off** and the profound challenge of **domain shift**. While the model excelled on the RLVSD dataset, its performance plummeted to **60% accuracy** when evaluated on the unseen Smart-City CCTV Violence Detection (SCVD) dataset. This stark contrast provides a sobering and valuable benchmark, highlighting that a model's strengths on one data distribution do not guarantee its utility in another. This finding underscores that while our lightweight model is a worthy contribution for its efficiency, achieving true state-of-the-art generalization, as demonstrated by recent Transformer-based models, requires more sophisticated architectures capable of capturing a broader range of contexts. This work, therefore, serves as a robust baseline and a cautionary tale, confirming the viability of the CNN+LSTM paradigm while clearly delineating its boundaries in the landscape of modern violence detection research.

## 7.1 Future Work

While this project achieved its objectives, it also opened several avenues for future investigation.

- **Improving Generalization:** The most critical next step is to address the domain shift problem. Future work should focus on training a more robust model on a diverse, aggregated dataset that combines RLVSD, SCVD, RWF-2000, and other public benchmarks. Techniques like domain adaptation and advanced data augmentation, specifically designed to simulate variations in camera perspective, lighting, and environmental clutter, should be explored to build a model that can generalize to truly "in-the-wild" scenarios.
- **Exploring Advanced Architectures:** A natural extension of this project would be to implement a Transformer-based model, such as ViViT or CUE-Net, and train it on the same datasets. This would allow for a direct comparison, quantifying the performance gains achieved by these more complex models against the associated increase in computational cost and training time.
- **Real-Time Implementation and Optimization:** To move closer to a practical application, future work could focus on deploying the model on an edge device like an NVIDIA Jetson Nano. This would involve optimizing the model using techniques like quantization with TensorFlow Lite and building an end-to-end pipeline to assess its real-time processing capabilities in a live video stream.

## 7.2 Lessons Learned

This project was a journey of both technical discovery and personal development, yielding several valuable lessons.

- **Technical Lessons:** The most profound technical lesson was the tangible reality of the "domain shift" problem. While the concept is well-documented, witnessing a model's accuracy fall from 94% to 60% provides a powerful and humbling demonstration of the brittleness of models trained on narrow data distributions. It reinforced the notion that high accuracy metrics are only meaningful within the context of their specific test set. Additionally, the project underscored the importance of a pragmatic approach to model selection. Starting with a well-established and efficient baseline (CNN+LSTM) proved far more insightful and manageable than immediately attempting to implement a complex state-of-the-art architecture.

**Project Management Lessons:** From a project management perspective, I learned the critical importance of a meticulous and robust data pipeline. The initial phase of the project, which involved debugging the custom VideoDataGenerator to correctly handle file paths and preprocessing, took a non-trivial amount of time but was fundamental to the project's ultimate success. Furthermore, this project reinforced the indispensable habit of frequent backups. While no catastrophic data loss occurred, the awareness that hours of training progress or a carefully curated dataset could be lost served as a constant reminder of the advice from the project coordinator to back up work frequently. If I were to start over, I would implement a more structured version control system, such as Git, from day one to more effectively track experiments and safeguard the project's codebase.

# 8 REFERENCES

- [ 1 ] techxmedia, “Presight Partners with Abu Dhabi Police on AI Policing,” TECHx Media - Online media and publishing platform for the technology community, covering top news and trends from MEA region’s tech and business world., Jul. 28, 2025. <https://techxmedia.com/en/presight-partners-with-abu-dhabi-police-on-ai-policing/> (accessed Jul. 28, 2025).
- [ 2 ] M. Bowen, “Presight and Abu Dhabi Police drive AI-enabled law enforcement and Smart City innovation – Intelligent CIO Middle East,” Intelligentcio.com, Jul. 05, 2025. <https://www.intelligentcio.com/me/2025/07/28/presight-and-abu-dhabi-police-drive-ai-enabled-law-enforcement-and-smart-city-innovation/> (accessed Jul. 28, 2025).
- [ 3 ] M. Cheng, K. Cai, and M. Li, “RWF-2000: An Open Large Scale Video Database for Violence Detection,” arXiv.org, 2019. <https://arxiv.org/abs/1911.05913>
- [ 4 ] PBS Organisation, “A look at data as concerns grow around surging violent crime,” Pbs.org, Mar. 12, 2024. <https://www.pbs.org/video/violent-crime-1710278686/>
- [ 5 ] FBI National Press Office, “FBI Releases 2024 Quarterly Crime Report and Use-of-Force Data Update | Federal Bureau of Investigation,” Federal Bureau of Investigation, 2024. <https://www.fbi.gov/news/press-releases/fbi-releases-2024-quarterly-crime-report-and-use-of-force-data-update>
- [ 6 ] WeeTech Solution, “AI in Security Camera Systems: Benefits, Challenges and Future Trends,” WeeTech Solution Pvt Ltd, Jan. 28, 2025. <https://www.weetechsolution.com/blog/ai-in-security-camera-systems> (accessed Jul. 28, 2025).
- [ 7 ] Università degli Studi di Bari, “Police Accountability in the,” 2024. Accessed: Jul. 28, 2025. [Online]. Available: <https://www.adir.unifi.it/repolity/research-report-1-en.pdf>
- [ 8 ] Office of Public Affairs , “FACT SHEET: National Law Enforcement Accountability Database,” Justice.gov, Dec. 19, 2024. <https://www.justice.gov/archives/opa/pr/fact-sheet-national-law-enforcement-accountability-database>
- [ 9 ] A. R. Abdali and A. A. Aggar, “DEVTrV2: Enhanced Data-Efficient Video Transformer For Violence Detection,” 2022 7th International Conference on Image,

Vision and Computing (ICIVC), Jul. 2022, doi: <https://doi.org/10.1109/icivc55077.2022.9886172>.

- [ 10 ] Z. Amos, “5 challenges in implementing AI in video surveillance - DataScienceCentral.com,” Data Science Central, May 13, 2025. <https://www.datasciencecentral.com/5-challenges-in-implementing-ai-in-video-surveillance/>
- [ 11 ] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, “Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network,” Sensors, vol. 19, no. 11, p. 2472, May 2019, doi: <https://doi.org/10.3390/s19112472>.
- [ 12 ] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” arXiv.org, Dec. 10, 2015. <https://arxiv.org/abs/1512.03385>
- [ 13 ] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” arXiv.org, 2018. <https://arxiv.org/abs/1801.04381>
- [ 14 ] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” arXiv.org, 2014. <https://arxiv.org/abs/1412.0767>
- [ 15 ] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 1, pp. 1–42, Jan. 1997, doi: <https://doi.org/10.1162/neco.1997.9.1.1>.
- [ 16 ] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” arXiv:1506.04214 [cs], vol. 2, Sep. 2015, Available: <https://arxiv.org/abs/1506.04214>
- [ 17 ] T. T. Le, Karim Abed-Meraim, Philippe Ravier, Olivier Buttelli, and Ales Holobar, “Joint INDSCAL Decomposition Meets Blind Source Separation,” ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5570–5574, Mar. 2024, doi: <https://doi.org/10.1109/icassp48485.2024.10447387>.
- [ 18 ] Senadeera, Damith Chamalke, X. Yang, D. Kollias, and G. Slabaugh, “CUE-Net: Violence Detection Video Analytics with Spatial Cropping, Enhanced UnifomerV2 and Modified Efficient Additive Attention,” arXiv.org, 2024. <https://arxiv.org/abs/2404.18952> (accessed Jul. 28, 2025).
- [ 19 ] Moch Arief Soeleman, Catur Supriyanto, Dwi Puji Prabowo, and Pulung Nurtantio Andono, “Video Violence Detection Using LSTM and Transformer Networks Through Grid Search-Based Hyperparameters Optimization,” International Journal of Safety and Security Engineering, vol. 12, no. 05, pp. 615–622, Nov. 2022, doi: <https://doi.org/10.18280/ijsse.120510>.
- [ 20 ] S. Sudhakaran and O. Lanz, “Learning to detect violent videos using convolutional long short-term memory,” 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Aug. 2017, doi: <https://doi.org/10.1109/avss.2017.8078468>.
- [ 21 ] M. Asad, J. Yang, J. He, Pourya Shamsolmoali, and X. He, “Multi-frame feature-fusion-based model for violence detection,” The Visual Computer, vol. 37, no. 6, pp. 1415–1431, Jun. 2020, doi: <https://doi.org/10.1007/s00371-020-01878-6>.
- [ 22 ] M. Shoaib and N. Sayed, “A Deep Learning Based System for the Detection of Human Violence in Video Data,” Traitement du Signal, vol. 38, no. 6, pp. 1623–1635, Dec. 2021, doi: <https://doi.org/10.18280/ts.380606>.

- [ 23 ] D. J. Samuel R. et al., “Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM,” Computer Networks, vol. 151, pp. 191–200, Mar. 2019, doi: <https://doi.org/10.1016/j.comnet.2019.01.028>.
- [ 24 ] M. A. Soeleman, C. Supriyanto, and D. P. Prabowo, “An Empirical Study of CNN-LSTM on Class Imbalance Datasets for Violence Video Detection,” Proceedings of the 2021 International Conference on Computer, Control, Informatics and Its Applications, pp. 81–85, Oct. 2021, doi: <https://doi.org/10.1145/3489088.3489126>.
- [ 25 ] E. Veltmeijer, M. Franken, and C. Gerritsen, “Real-time violence detection and localization through subgroup analysis,” Multimedia Tools and Applications, vol. 84, May 2024, doi: <https://doi.org/10.1007/s11042-024-19144-5>.
- [ 26 ] W. Song, D. Zhang, X. Zhao, J. Yu, R. Zheng, and A. Wang, “A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks,” IEEE Access, vol. 7, pp. 39172–39179, 2019, doi: <https://doi.org/10.1109/ACCESS.2019.2906275>.
- [ 27 ] S. Akti, G. A. Tataroglu, and H. K. Ekenel, “Vision-based Fight Detection from Surveillance Cameras,” 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), Nov. 2019, doi: <https://doi.org/10.1109/ipta.2019.8936070>.
- [ 28 ] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, and A. F. Dragoni, “Deep Learning for Automatic Violence Detection: Tests on the AIRTLab Dataset,” IEEE Access, vol. 9, pp. 160580–160595, 2021, doi: <https://doi.org/10.1109/access.2021.3131315>.
- [ 29 ] P. Zhou, Q. Ding, H. Luo, and X. Hou, “Violent Interaction Detection in Video Based on Deep Learning,” Journal of Physics: Conference Series, vol. 844, p. 012044, Jun. 2017, doi: <https://doi.org/10.1088/1742-6596/844/1/012044>.
- [ 30 ] A. Traore and M. A. Akhloufi, “Violence Detection in Videos using Deep Recurrent and Convolutional Neural Networks,” Computer Vision and Pattern Recognition, Oct. 2020, doi: <https://doi.org/10.1109/smvc42975.2020.9282971>.
- [ 31 ] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, “Action Recognition with Dynamic Image Networks,” arXiv.org, 2024. <https://arxiv.org/abs/1612.00738>
- [ 32 ] A. Jain and D. K. Vishwakarma, “Deep NeuralNet For Violence Detection Using Motion Features From Dynamic Images,” IEEE Xplore, Aug. 01, 2020. <https://ieeexplore.ieee.org/abstract/document/9214153> (accessed Nov. 11, 2022).
- [ 33 ] A. Vaswani et al., “Attention Is All You Need,” arXiv, Jun. 12, 2017. <https://arxiv.org/abs/1706.03762>
- [ 34 ] M. Akil, M. M. Rahman, and S. Das, “VioNet: An Enhanced Violence Detection Approach for Videos Using a Fusion Model of Vision Transformer with Bi-LSTM and 3D Convolutional Neural Networks,” Lecture notes in networks and systems, pp. 139–151, Jan. 2024, doi: [https://doi.org/10.1007/978-981-99-8937-9\\_10](https://doi.org/10.1007/978-981-99-8937-9_10).
- [ 35 ] T. Zhang, W. Jia, K.-N. Wu, J. Yang, X. He, and Z. Zheng, “MoWLD: a robust motion image descriptor for violence detection,” Multimedia Tools and Applications, vol. 76, no. 1, pp. 1419–1438, Jan. 2017, doi: <https://doi.org/10.1007/s11042-015-3133-0>.
- [ 36 ] J. Mahmoodi and A. Salajeghe, “A classification method based on optical flow for violence detection,” Expert Systems with Applications, vol. 127, pp. 121–127, Aug. 2019, doi: <https://doi.org/10.1016/j.eswa.2019.02.032>.

- [ 37 ] P. Zhou, Q. Ding, H. Luo, and X. Hou, “Violence detection in surveillance video using low-level features,” PLOS ONE, vol. 13, no. 10, p. e0203668, Oct. 2018, doi: <https://doi.org/10.1371/journal.pone.0203668>.
- [ 38 ] D. Kollias et al., “DVD: A Comprehensive Dataset for Advancing Violence Detection in Real-World Scenarios,” arXiv.org, 2025. <https://arxiv.org/abs/2506.05372v1> (accessed Jul. 28, 2025).
- [ 39 ] I. A. Vezakis, G. I. Lambrou, and G. K. Matsopoulos, “Deep Learning Approaches to Osteosarcoma Diagnosis and Classification: A Comparative Methodological Approach,” Cancers, vol. 15, no. 8, p. 2290, Apr. 2023, doi: <https://doi.org/10.3390/cancers15082290>.
- [ 40 ] thesuriya, “Violence-Detection,” Kaggle.com, Mar. 06, 2024. <https://www.kaggle.com/code/thesuriya/violence-detection> (accessed Jul. 28, 2025).
- [ 41 ] S. Sharma, B. Sudharsan, S. Naraharisetti, V. Trehan, and K. Jayavel, “A fully integrated violence detection system using CNN and LSTM,” International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 4, p. 3374, Aug. 2021, doi: <https://doi.org/10.11591/ijece.v11i4.pp3374-3380>.
- [ 42 ] T. Aremu, L. Zhiyuan, R. Alameeri, M. Khan, and S. A. El, “SSIVD-Net: A Novel Salient Super Image Classification & Detection Technique for Weaponized Violence,” arXiv.org, 2022. <https://arxiv.org/abs/2207.12850> (accessed Jul. 28, 2025).
- [ 43 ] theiet, “Rules of Conduct,” www.theiet.org, 2024. <https://www.theiet.org/about/governance/rules-of-conduct>
- [ 44 ] British Computer Society, “BCS Code of Conduct | BCS,” www.bcs.org, 2025. <https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct>
- [ 45 ] L. Shen , J. Hu , S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” *arXiv:1709.01507 [cs]*, May 2019, Available: <https://arxiv.org/abs/1709.01507>
- [ 46 ] Q. Xia, P. Zhang, J. Wang, M. Tian, and C. Fei, “Real Time Violence Detection Based on Deep Spatio-Temporal Features,” *Lecture notes in computer science*, pp. 157–165, Jan. 2018, doi: [https://doi.org/10.1007/978-3-319-97909-0\\_17](https://doi.org/10.1007/978-3-319-97909-0_17)

