

Master of Science in Computer Science and Engineering



VIOLENCE DETECTION IN VIDEOS USING DEEP LEARNING

by

Md. Al-Mamun Provath

ID: 22MCSE004P

This thesis is submitted in partial fulfillment of the requirement for the degree of
MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chattogram-4349, Bangladesh.
April, 2025

CERTIFICATION

The thesis titled **VIOLENCE DETECTION IN VIDEOS USING DEEP LEARNING** submitted by **Md. Al-Mamun Provath**, Roll No. **22MCSE004P**, Session **2022-2023** has been accepted as satisfactory in partial fulfilment of the requirement for the degree of **Master of Science in Computer Science and Engineering** on **29/04/2025**.

BOARD OF EXAMINERS

1. _____
Prof Dr. Kaushik Deb Chairman (Supervisor)
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET).
2. _____
Prof. Dr. Abu Hasnat Mohammad Ashfak Habib Member (Ex-officio)
Professor and Head
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET).
3. _____
Prof. Dr. Mohammad Ibrahim Khan Member
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET).
4. _____
Prof. Dr. Pranab Kumar Dhar Member
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET).
5. _____
Prof. Dr. Boshir Ahmed Member (External)
Professor
Department of Computer Science and Engineering
Rajshahi University of Engineering & Technology (RUET).

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Md. Al-Mamun Provath
ID: 22MCSE004P

Dedicated to,
my beloved parents and elder brother
for their love, endless support, encouragement and sacrifices.

Acknowledgment

My unending gratitude to my thesis support team whose unwavering support and inspiration were key to bring my thesis to this stage. Celebrating this milestone would have no meaning without acknowledging the best people in the world who stood beside me in this journey.

It is with great appreciation that I first acknowledge my supervisor, Prof. Dr. Kaushik Deb, whose profound knowledge, expert guidance, constructive feedback, and unwavering encouragement were pivotal in shaping this research. His mentorship served as an invaluable beacon throughout this academic journey. I am also deeply thankful to the members of the review panel for their thoughtful insights and suggestions, which played a crucial role in refining this work.

Additionally, I extend my sincere gratitude to all the faculty members and staffs of the Department of Computer Science and Engineering at CUET for their consistent support and the resources they provided. A heartfelt thank you also goes to my fellow cohort members and friends—your companionship and encouragement were essential in navigating this challenging yet rewarding path.

Above all, I am profoundly grateful to Almighty Allah for bestowing upon me the strength and perseverance needed to bring this thesis to its completion.

Abstract

Video violence detection is a critical tool for enhancing public safety in smart cities, particularly in managing the rising tide of political violence. Current datasets and conventional methods often fail to address the intricate nature of political violence, resulting in models that lack robustness and precision. To overcome these limitations, this study introduces a novel approach by creating a specialized dataset tailored to political violence, consisting of 480 meticulously labeled video clips spanning four distinct categories: Normal, Clash, Fire, and Shooting. This dataset offers a comprehensive resource for understanding political violence dynamics and serves as a vital benchmark for advancing research in this domain. We fine-tuned the Mobile Video Network (MoViNet-A0), achieving an impressive accuracy of 92.86% with a lightweight architecture of 1.904 million parameters, making it suitable for real-time applications. Additionally, we developed a custom keyframe extraction algorithm that integrates temporal and pixel-level features, enhancing stability and accuracy in identifying critical frames. Our methodology outperforms state-of-the-art techniques across multiple benchmark datasets, including Hockey Fight (98%), RLVS (98.3%), RWF-2000 (88.12%), and Surveillance Camera Fight (84%). These advancements highlight the system’s potential to strengthen security measures, inform policy decisions, and provide a scalable solution for detecting and classifying political violence in video surveillance, ultimately contributing to safer urban environments.

Keywords: Video surveillance, Video violence, Keyframe extraction, Security, Deep learning.

Contents

Acknowledgment	i
Abstract	ii
1 Introduction	1
1.1 Introduction	1
1.2 Video Classification	1
1.3 Motivation	2
1.4 Objectives	3
1.5 Scope of the Work	4
1.6 Importance	5
1.7 Challenges	5
1.8 Contributions	6
1.9 Organization of Thesis	7
1.10 Conclusion	7
2 Literature Review	8
2.1 Introduction	8
2.2 Traditional methods	8
2.3 Methods based on 3D CNNs and 2D CNNs	9
2.4 Methods based on 2D CNNs and LSTM	10
2.5 Methods based on two stream CNNs	11
2.6 Limitations of Current Methods and Research Gaps	12
2.7 Conclusion	15
3 Methodology	16
3.1 Introduction	16
3.2 Overview of Methodology	16
3.3 Definition of the Task	17
3.4 Dataset Developmet Steps	18
3.4.1 Data Curation	18
3.4.2 Annotation	18
3.4.3 Preprocessing	21
3.5 Data Annotation Quality	23
3.6 Dataset Statistics	24
3.7 Problem Formulation	26
3.8 Keyframe Extraction	28
3.9 Feature Extraction	34
3.9.1 ConvLSTM	35

3.9.2	2D CNN + LSTM/BiLSTM	38
3.9.3	Vision Transformer	40
3.9.4	Mobile Video Networks(MoViNet-A0)	42
3.10	Classification	46
3.11	Conclusion	47
4	Results and Discussions	48
4.1	Introduction	48
4.2	Experimental Requirements	48
4.3	Evaluation Measures	49
4.4	Evaluation of Keyframe Extraction Algorithm	49
4.5	Redundancy Reduction of Keyframe Extraction	52
4.6	Impact of Histogram Equalization	56
4.7	Fine-tuning MoViNet-A0	58
4.8	Feature Extraction Methods Comparison	60
4.9	Ablation Study	65
4.10	Generalization Test	66
4.11	Failure Case Analysis	69
4.12	Managerial Implications	69
4.13	Real-Time Performance Analysis	70
4.14	Conclusion	70
5	Conclusion and Future Recommendations	72
5.1	Conclusion	72
5.2	Future Work	73
5.3	List of Publications	74

List of Figures

1.1	Monitoring CCTV cameras in real-life	2
3.1	Overview of the violence detection system workflow.	17
3.2	Data acquisition summary of Political Violence dataset. This pie chart illustrates the distribution of data acquisition across three major social media platforms: Facebook (60%), YouTube (30%), and Instagram (10%). The chart highlights Facebook as the primary source of data, followed by YouTube and Instagram.	19
3.3	Data Annotation process.	20
3.4	Comparison of video frames pre- and post-histogram equalization: The left column shows the original frames as captured by mobile cameras, while the right column demonstrates the clarity achieved through histogram equalization.	22
3.5	Samples from the Political Violence dataset: (a) Clash; (b) Fire; (c) Shooting; (d) Normal class.	25
3.6	Extracted key frames for a Clash class sample video.	34
3.7	Extracted key frames for a Fire class sample video.	34
3.8	Silhouette score visualization: the silhouette scores calculated for various values of K (peak at $K = 12$).	35
3.9	The ConvLSTM architecture for feature architecture.	36
3.10	The 2D CNN + LSTM/ BiLSTM architecture	38
3.11	Overview of the Compact Convolutional Transformer (CCT) pipeline for video violence classification.	40
3.12	Overview of the multi-clip MoViNet-A0 with causal stream buffer architecture.	43
4.1	Clash class clustering results with and without temporal features. Subplots (a) and (b) show 2D PCA-reduced clusters, with improved separation for the proposed algorithm (Davies-Bouldin index $DB = 0.62$, cluster cohesion ratio R_c). Subplot (c) illustrates smoother transitions, with higher temporal smoothness index (S_{temp}) and narrative continuity score (N_{cont}).	50
4.2	Fire class clustering results with and without temporal features.	50
4.3	K-means Cluster Assignment Over Time (k=12) illustrating the assignment of frames into clusters based solely on visual features. The process considers both near and far frames, leading to clusters with diverse actions.	55

4.4	Cluster Assignment Over Time (k=12) using the proposed algorithm, which incorporates both temporal and visual features. The method favors grouping temporally adjacent frames with similar actions, resulting in better keyframe diversity and improved representation of the video timeline.	55
4.5	Intensity vs. Temporal Feature (k=12). The scatter plot shows the distribution of frames based on normalized temporal features and their average intensity. Keyframes selected by the proposed method (red stars) and by standard K-means (green circles) are compared. The blue circles highlight regions where K-means selects multiple redundant keyframes from visually similar and temporally close frames, whereas the proposed method avoids such redundancy by considering both visual similarity and temporal diversity.	56
4.6	Keyframes selected by K-means clustering (k=12).	57
4.7	Keyframes selected by proposed algorithm for clustering (k=12).	57
4.8	Confusion matrix for the CCT transformer model, showing the actual versus predicted classes for Fire (0), Clash (1), Normal (2), and Shooting (3).	60
4.9	Confusion matrix for the 2D CNN + LSTM model, showing the actual versus predicted classes for Fire (0), Clash (1), Normal (2), and Shooting (3).	62
4.10	Confusion matrix for the ConvLSTM model, illustrating the classification performance across four classes: Fire (0), Clash (1), Normal (2), and Shooting (3).	63
4.11	Training performance: accuracy curve on the left and loss curve on the right.	64
4.12	Confusion matrix for Model 2 evaluation.	64
4.13	ROC curve for Model 2 evaluation.	65
4.14	Processing examples of categorizing violence: (a) Clash; (b) Fire; (c) Shooting; (d) Normal.	66
4.15	Example of misclassification: Left shows an individual carrying a gun during Clash action; right shows individuals with closed fists in a public gathering.	69

List of Tables

3.1	Cohen’s Kappa Scores for the Political Violence Dataset Annotations	24
3.2	Distribution of Video Clips in the Political Violence Dataset Across Training, Validation, and Test Sets	25
3.3	Characteristics of various datasets for video violence detection.	26
4.1	Environmental Details of the System Setup: Hardware and Software Specifications	48
4.2	K-means Keyframes (k=12).	53
4.3	Proposed Algorithm Keyframes (k=12).	54
4.4	Impact of histogram equalization on violence detection accuracy.	58
4.5	Accuracy of models with different layers fine-tuned.	59
4.6	Optimized hyperparameters for model training.	59
4.7	Classification report for the CCT transformer model, detailing precision, recall, F1-score, and support for each class, along with overall metrics.	61
4.8	Classification report for the 2D CNN + LSTM model, detailing precision, recall, F1-score, and support for each class, along with overall metrics.	61
4.9	Performance metrics for the ConvLSTM model, including precision, recall, F1-score, and support for each class, along with overall accuracy, macro average, and weighted average.	62
4.10	Classwise performance metrics of Model 2.	63
4.11	Ablation study results with base (B), fine-tuned (F), pixel (P), and temporal (T) information.	66
4.12	Accuracy Comparison of violence detection methods - Hockey Fight dataset.	67
4.13	Accuracy comparison of violence detection methods - RLVS dataset.	67
4.14	Accuracy comparison of Violence detection methods - RWF-2000 Dataset.	67
4.15	Accuracy comparison of violence detection methods - Surveillance Camera Fight.	68

Chapter 1

Introduction

1.1 Introduction

Video violence classification involves the automated identification and categorization of violent activities within video footage, a critical capability for enhancing public safety in cities. This process leverages advanced computational techniques to analyze video data from sources like CCTV cameras, enabling real-time detection and response to incidents such as physical altercations, armed conflicts, and political violence. By employing deep learning models, video violence classification systems can extract complex spatiotemporal features, surpassing the limitations of traditional manual monitoring and feature-based methods.

1.2 Video Classification

Video classification entails assigning predefined labels to video sequences based on their content, a task that has gained prominence with the proliferation of surveillance systems. In the context of violence detection, this involves distinguishing between violent and non-violent behaviors using datasets that represent diverse scenarios. The United Nations Office on Drugs and Crime highlights the scale of this issue, reporting nearly 500,000 homicides globally in 2017, underscoring the need for effective classification tools. With an estimated 770 million surveillance cameras worldwide by 2019, the volume of video data necessitates automated solutions to process and classify content efficiently.

1.3 Motivation

The surge in political unrest across the globe has significantly fueled the motivation for advancing video violence classification, Figure 1.1[1]. As protests, clashes, and politically motivated conflicts become more frequent, the need for effective tools to monitor and manage these incidents has intensified. This rise in unrest, often documented in global crime statistics such as those from the United Nations Office on Drugs and Crime [2], underscores the urgency to develop automated systems capable of identifying and categorizing violent behaviors in real time, particularly in urban environments where political tensions can escalate rapidly.



Figure 1.1: Monitoring CCTV cameras in real-life .

Compounding this need is the sheer volume of video footage generated by modern surveillance infrastructure, with an estimated 770 million cameras operational worldwide as of 2019 [3]. This overwhelming amount of data renders continuous monitoring by human operators impossible, leading to significant scalability issues. Human limitations, such as fatigue and distraction, further exacerbate the problem, as operators frequently miss critical events [4], highlighting the necessity for automated classification systems that can process and analyze video streams efficiently and reliably.

Moreover, while video surveillance systems are widely deployed, they are rarely utilized proactively to prevent criminal activities, despite a strong desire to reduce reaction times to violent incidents. In politically volatile contexts, where rapid response can mitigate escalation and harm, this gap becomes particularly pronounced. The push for automated video violence classification is thus a criti-

cal research focus, aiming to transform passive monitoring into an active tool for intervention, enabling authorities to act swiftly and decisively to maintain public safety. Our major motivation to work in this area are,

- To create a system that detects and classifies video content exhibiting violence threatening public safety, social stability, and urban security.
- If we can identify violent video sequences and pinpoint specific types of violence, it will empower law enforcement to locate perpetrators and prevent escalation.
- Such a system will improve the safety and reliability of surveillance in smart city environments.
- To build models and datasets that advance the research community and technology sectors focused on video analysis and violence detection.

1.4 Objectives

The primary goal of this research is to design an efficient and accurate system for detecting political violence in video surveillance, overcoming the limitations of existing datasets and computational methodologies. The specific objectives are:

1. **Develop a Comprehensive Dataset:** Create a video violence dataset with 480 labeled video clips across four categories (Normal, Clash, Fire, Shooting) to address the scarcity of specialized resources for political violence detection, ensuring robust representation of real-world scenarios.
2. **Design an Efficient Keyframe Extraction Algorithm:** Design a novel keyframe extraction algorithm that integrates temporal and pixel-level features to improve clustering stability and classification accuracy, enhancing the system’s ability to identify critical video frames.
3. **Optimize and Benchmark a Lightweight Model:** Fine-tune the MoViNet-A0 model to achieve 92.86% accuracy with only 1.904 million parameters, enabling real-time deployment in resource-constrained smart city environments, and evaluate the system against benchmark datasets (Hockey Fight, RLVS, RWF-2000, Surveillance Camera Fight) to demonstrate superior accuracy and generalizability.

These objectives collectively aim to deliver a scalable, high-performance solution for automated violence detection, advancing public safety and urban security through intelligent video analysis.

1.5 Scope of the Work

The scope of this research is strategically delineated to address the pressing need for automated detection and classification of violence in video surveillance, with a specific focus on enhancing public safety in smart city environments. This study concentrates on developing a robust system capable of identifying and categorizing violent incidents within video footage, leveraging advanced deep learning techniques to overcome the limitations of existing methodologies and datasets. The research is anchored in the creation of a novel Political Violence Dataset, comprising 480 meticulously labeled video clips categorized into four distinct classes: Normal, Clash, Fire, and Shooting. These categories were selected to encapsulate a comprehensive spectrum of political violence manifestations, reflecting the diverse nature of such incidents in real-world scenarios.

The rationale for selecting these four categories is rooted in their prevalence and significance within the context of political unrest, as well as their distinct visual and temporal characteristics, which pose unique challenges for automated detection. The Normal class represents non-violent scenarios, serving as a baseline to distinguish benign activities from violent ones. The Clash category encompasses physical altercations, such as protests or riots, characterized by dynamic group interactions and aggressive behaviors. The Fire class includes incidents involving arson or explosions, which are visually distinct due to their intense illumination and rapid motion patterns. Finally, the Shooting category covers firearm-related violence, marked by abrupt movements and specific object-based cues, such as weapons. These categories collectively address the multifaceted nature of political violence, enabling the development of a model that can generalize across varied violent events while maintaining specificity in classification.

This research further focuses on optimizing computational efficiency to ensure real-time applicability, a critical requirement for deployment in resource-constrained urban surveillance systems. By fine-tuning the lightweight Mobile Video Network (MoViNet-A0) and designing a custom keyframe extraction algorithm that integrates temporal and pixel-level features, the study aims to achieve high accuracy with minimal computational overhead. The scope is limited to video-based analysis, excluding multimodal inputs such as audio or textual data, to maintain a focused investigation into visual feature extraction and classification. The system’s performance is evaluated against both the newly developed Political Violence Dataset and established benchmark datasets, including Hockey Fight, RLVS, RWF-2000, and Surveillance Camera Fight, to validate its generalizability and superiority over state-of-the-art methods. Through this targeted

scope, the research seeks to provide a scalable, efficient, and accurate solution for automated political violence detection, contributing to enhanced urban security and informed policy-making.

1.6 Importance

Video violence classification is a critical tool for bolstering safety and security in urban environments, enabling proactive measures to mitigate risks and protect lives. Its applications span multiple domains, each contributing to a safer, more resilient society :

- By monitoring transit areas, the system can detect and respond to violent incidents swiftly, reducing harm and improving public safety on critical travel routes.
- Early identification of unrest during political activities allows for timely interventions, helping to maintain order and prevent escalation.
- Surveillance of key assets like power plants and communication networks helps thwart politically motivated attacks, ensuring operational continuity and public welfare.
- With the potential to address violence that claims countless lives, the system strengthens security measures, fostering safer urban communities.

1.7 Challenges

Video violence classification faces several obstacles that must be addressed to ensure its effectiveness and practicality in real-world settings. These challenges span technical, environmental, and resource-related domains, each presenting unique difficulties.

- **Non-Stationary Backgrounds:** Constantly changing backgrounds make it harder to extract reliable features, complicating the identification of violent actions.
- **Dynamic Illumination Variations:** Fluctuating lighting conditions degrade video quality and impair model performance, reducing detection accuracy.

- **Limited Datasets for Political Violence:** The lack of accessible, specialized datasets tailored to political violence slows progress in developing robust models.
- **Computational Efficiency:** Achieving a balance between fast training and inference times is essential for real-time use, yet remains difficult to optimize.
- **Reducing Trainable Parameters:** Minimizing computational complexity while preserving high accuracy is a key technical challenge, particularly in resource-limited smart city systems.

1.8 Contributions

Our main contributions are summarized below:

1. **Developed a new dataset for violence detection**, comprising 480 labeled video clips across four distinct categories of violence incidents. This dataset addresses the critical shortage of publicly available resources in this field and serves as a comprehensive benchmark for the development and evaluation of automated violence detection algorithms.
2. **Designed a custom key frame extraction algorithm** that leverages both temporal and pixel-level features to identify the most informative frames. This approach enhances the stability and accuracy of key frame clusters, which are subsequently utilized in the fine-tuned model for classification.
3. **Fine-tuned the MoViNet-A0 model** on the Political Violence dataset, achieving a notable accuracy of 92.86% with a parameter count of 1.904 million. This result underscores the model’s effectiveness in the domain of video violence detection.
4. **Attained performance exceeding state-of-the-art methods** across four benchmark datasets for video violence detection—Hockey Fight (98% accuracy), RLVS (98.3% accuracy), RWF-2000 (88.12% accuracy), and Surveillance Camera Fight (84% accuracy). These results demonstrate the effectiveness and wide applicability of the proposed approach.

1.9 Organization of Thesis

The remaining of the thesis is organized as follows. Chapter 2 provides a comprehensive review of existing studies related to violence detection and classification in video content. Chapter 3 provides a comprehensive overview of the development of a video violence classification system, integrating dataset creation with the techniques and methodologies for system implementation. Additionally, the chapter details the methodologies for the classification system, including feature extraction to capture spatial and temporal dynamics, keyframe extraction to optimize computational efficiency, and classification algorithms for real-time detection. Experimental results, including both quantitative and qualitative error analysis, are discussed in Chapter 4. Finally, Chapter 5 highlights potential future research directions and concludes the thesis with a summary of findings.

1.10 Conclusion

Chapter 1 provides strong motivation to the work on violence detection on video, especially regarding its role in promoting the public security and urban security in the face of rapid increase of political unrest. It expresses the reason based on the increased demand from violent terrorism around the world and the difficulty of manual monitoring because the lidar volume of video from 770 million surveillance cameras all over the world increases massively. The chapter explicitly states the research objectives, which include creating a specialized Violence data set, proposing an effective key frame extraction technique, and optimizing a lightweight model MoViNet-A0 for real-time deployment. The range is defined in a strategic manner to tackle political violence with 4 categories (Normal, Clash, Fire, Shooting) offered, achieving extensive coverage at real-world scenes while maintaining computational efficiency. The significance of the system is demonstrated across various applications such as transit monitoring and infrastructure protection, and the challenges including non-stationary background and a finite training data set are recognised.

Chapter 2

Literature Review

2.1 Introduction

This chapter explores various techniques used in video violence detection, focusing on traditional methods, advanced neural network models, and hybrid approaches that integrate multiple models for improved performance. It highlights the evolution of these methods from manual feature extraction to sophisticated deep learning models, emphasizing their strengths and limitations in addressing the complexities of detecting violent content in video data.

2.2 Traditional methods

The most popular video violence detection approaches rely heavily on hand-crafted features. These methods can be classified into two general categories: local and global methods. Local methods focus on selective parts of the video, such as frames or segments, identifying points of interest. SPT and Motion SIFT (MoSIFT) [5] [6] are also examples of extending the spatial interest points to the spatiotemporal extent. On the other hand, global methods examine the whole video and utilize statistical models or extract features to determine behavioral patterns. These methods often employ optical flow to identify discriminative features. An example can be Violent Flow (ViF) descriptor, which depends on optical flow magnitude and its later versions have computed also magnitude and orientation [7]-[8].

While traditional methods have achieved reasonable success, they come with several drawbacks. They are typically time-consuming and require significant human effort to design and implement the necessary features. Additionally, these methods can be limiting, as the manually extracted features might not capture

all the data’s complexity or apply to various problems and tasks. Furthermore, traditional methods may face challenges with scalability and adaptability to new or unseen data, as they depend heavily on predefined features. Recently, the Entropy Weight Method (EWM) has gained attention for its ability to assign objective weights to features based on their inherent information entropy, thereby enhancing the feature selection and decision-making processes in classification tasks. For instance, [9] applied EWM in conjunction with machine learning classifiers to improve the accuracy and robustness of violence detection systems by effectively weighing key features extracted from video data. This approach addresses the limitations of traditional classifiers by providing a more nuanced feature weighting mechanism, leading to better performance in complex scenarios.

2.3 Methods based on 3D CNNs and 2D CNNs

Recently, video violence detection methods that use 3D CNNs and their variants have been widely preferred due to their capability for automatic extraction of spatiotemporal features. Since the convolutional operations in 3D CNNs are performed in three dimensions, both the spatial and temporal information could be modeled accordingly. That enables them to be more effective at modeling the temporal dynamics of the video sequences to perform the task of violence detection better [10]. For example, it is shown in [11] that 3D CNNs worked excellently for violence detection. The authors proposed a network trained on the Hockey Fight dataset that outperformed standard methods.

Moreover, the adaptability of CNN-based frameworks extends beyond video violence detection. In medical imaging, CNNs have been enhanced with fuzzy logic to improve diagnostic reasoning [12], and principal component analysis has been integrated to forecast blast-induced ground vibrations [13]. A deep CNN framework has also been proposed to enhance brain tumor diagnosis on MRI scans, demonstrating the potential of CNNs in biomedical applications [14]. Additionally, CNNs have been utilized in optimizing radar systems, with novel machine learning approaches improving radar warning receiver preprogramming [15]. Furthermore, integrating CNNs with fuzzy K-Clustering and fuzzy neural networks has enhanced prediction accuracy in complex systems [16].

However, the use of 3D CNNs has some cons: a lot of computational burden and plenty of parameters involved increase time consumption during training, with huge requirements to computational capabilities [17]. The complexity here makes large datasets necessary to avoid overfitting, which is not always available. Several techniques have been developed to make the 3D CNN less ineffective in

handling these challenges. For example, [18] proposed a technique for interest frame extraction to reduce the size of input fed to 3D CNNs, reducing computational burdens.

While 3D CNNs have been proposed for video analysis by introducing temporal dimensions into their convolutional and pooling operations, adapting 2D CNNs for video analysis has traditionally been used for image analysis by incorporating temporal information through various strategies. These networks are less computationally intensive than the 3D CNNs since they perform convolutions over two dimensions. Several methods have been devised to enable 2D CNNs to capture temporal features, including using optical flow or frame differencing [19]. More recently, keyframe differences based on an optical flow have been considered for template generation in 2D CNNs, which have been competitive in performance for the violence detection task[20].

However, 2D CNNs suffer from some of their representative characteristic features. Their performance could hardly catch the temporal dynamics from video data since they are mostly based on a spatial feature from an independent frame [21]. This may result in the loss of significant temporal information which is necessary for accurately recognizing violent activities. Drawing inspiration from advanced CNN applications in other domains, such as epileptic seizure detection through EEG analysis [22] and software vulnerability detection [23], our approach integrates temporal and spatial feature extraction to mitigate these challenges.

2.4 Methods based on 2D CNNs and LSTM

Recently, the combination between 2D CNNs and LSTM networks has also been [fruitful in violence detection in videos. This is due to each of above models complementing other to contribute power to this complete model: If 2D CNN is responsible for extracting the most important spatial content of a video frame, LSTM is for temporal dependencies over sequences. Such synergy also provides a holistic understanding of the video data which tends to make the multimodal approach highly suitable for predicting violent scenarios [24].

Research [25] was one such representation that combined 2D CNN with LSTM to improve the performance of violence detection. While 2D CNN takes out the spatial features in every frame, LSTM takes that as an input and models temporal relations between them. This has been promising in detecting violent behaviors correctly from video sequences.

Moreover, CNN-based frameworks have been successfully applied in diverse domains, enhancing tasks such as epileptic seizure detection through EEG anal-

ysis [22] and optimizing software vulnerability detection with MDSADNet [23]. These applications underline the adaptability and effectiveness of CNN-LSTM architectures in handling complex, sequence-based data.

Despite their effectiveness, methods combining two-dimensional Convolutional Neural Networks (2D CNNs) with Long Short-Term Memory (LSTM) networks still face notable challenges. These hybrid architectures often result in increased model complexity, demanding significant computational power and lengthy training periods [26]. Additionally, their performance heavily depends on the quality and quantity of training data. Insufficient or low-quality data can lead to overfitting, where models perform well on training sets but struggle with new, unseen data [27].

Furthermore, while LSTMs excel at capturing short-term temporal dependencies, they may struggle with long-range dependencies prevalent in video sequences. This limitation can result in the loss of crucial contextual details [28]. To address these issues, researchers have turned to hybrid models incorporating attention mechanisms. These mechanisms allow models to focus on specific parts of a video, thereby improving their ability to discern relevant features [29].

Our approach draws inspiration from advancements in other domains, such as using deep CNN frameworks to enhance brain tumor detection on MRI scans [14] and employing innovative machine learning techniques for radar warning receiver optimization [15]. By integrating attention mechanisms with CNN-LSTM architectures, we aim to better capture both spatial and temporal features essential for accurate violence detection.

2.5 Methods based on two stream CNNs

The detection of violence in videos is currently trending, taking the traditional form of a two-stream approach in CNN-based methods. A two-stream process would ideally include raw RGB frames in one stream and the optical flow indicating motion in between the frames in another. By combining these two streams, both spatial and temporal features can be leveraged for better violent activity detection, as used in the work by [30].

For instance, authors in [31] proposed a two-stream CNN architecture and achieved a considerable enhancement in action recognition. The network used both spatial and temporal information quite elegantly and performed better than the best single-stream network. In this respect, a two-stream model introduced in [32] for violence detection showed competitive results based on the integration of the motion information with one of the appearance data.

Moreover, the two-stream paradigm has been successfully applied in various other domains. In radar systems, a novel machine learning approach optimized radar warning receiver preprogramming, demonstrating the flexibility of two-stream architectures in handling complex data [15].

Although two-stream CNNs have their merits, it has some disadvantages. One of the significant problems is increased computational complexity as it requires the processing of two different streams, which may increase the time of training as well as resources [33]. Furthermore, optical flow provides more challenges since the computation of optical flow is a complex operation and sometimes provides inaccurate measurements, particularly in video data with low quality or noise [34].

To address these limitations, recent advancements have focused on optimizing two-stream architectures. For example, integrating dual graph neural networks with multi-feature fusion has been explored to enhance named entity recognition. This provides insights into how multi-stream data can be effectively combined [35]-[36]. Similarly, enhanced prediction accuracy in complex systems has been achieved by integrating fuzzy K-Clustering with fuzzy neural networks, which can inspire more robust two-stream models [16].

Most of the prior works in video violence detection have focused on binary classification, leaving much room for improvement in the accuracy of this task. One of the main challenges is indeed to develop models that are not only accurate but also lightweight, with a good balance between performance and computational efficiency. Among the several proposals to date, the MoViNet model seems very promising for solving this challenge, especially its A0 version - the smallest of its other versions, which we have chosen to fine-tune to optimize its effectiveness. Probably the most challenging part of the detection task is the fact that real political incident datasets are virtually non-existent. To remedy this, we provide the Political Violence Dataset to help cover that major gap in resources and permit more significant and pertinent detection capabilities.

2.6 Limitations of Current Methods and Research Gaps

Conventional methodologies of video violence detection are built upon hand-crafted features involving motion, appearance, and trajectory-based descriptors. Most works have adapted machine learning-based classification frameworks such as SVM or KNN [37]. However, these methods fail to model complex spatiotem-

poral patterns and have a lack of generalization ability [38]. Lately, with the evolution of deep learning, 2D CNNs have been widely adopted for video violence detection. These networks are very good at extracting spatial features from the individual frames, which are then aggregated temporally by techniques like average pooling or RNNs [39].

The improved performance of 2D CNNs compared with traditional methods can be achieved because those methods do not capture temporal dynamics and long-range dependencies in videos [40]. Recently, with the view to overcoming deficiencies of 2D CNNs, there has been interest in the use of 3D CNNs due to their abilities for jointly learning spatiotemporal features by applying convolutions across both spatial and temporal dimensions[41]. 3D CNNs thus far show the best performance of any of the methods discussed in the video violence detection, as they can model effectively complex interactions between appearance and motion patterns[42].

However, 3D CNNs have their challenges, especially when it comes to computational efficiency and memory requirements [43]. Most operations, essential to support the accuracy of the 3D video networks, require processing of all frames in the input. Temporal convolution and non-local blocks are a few examples. At the same time, many other approaches impose such a burden-the serious challenge to use an efficient model on a mobile device [44]. Recently presented X3D networks achieved impressive advancements in 3D CNN efficiency. However, either they require large amounts of memory with large temporal windows or utilize smaller temporal windows that lower the performance [45]. Other works have also tried to improve the performance of 2D CNNs by performing temporal aggregations. Still, due to their limited inter-frame interactions, these models also cannot capture long-range temporal dependencies as well as 3D CNNs do [46].

To overcome these challenges, there is a growing need for MoViNets, a family of memory and computation-efficient 3D CNNs [47]. MoViNets are designed to balance accuracy and efficiency, enabling deployment on resource-constrained devices while maintaining high performance in video violence detection tasks. By leveraging techniques such as channel-wise convolutions, temporal shift modules, and efficient temporal modeling, MoViNets can effectively capture spatiotemporal dependencies while minimizing computational overhead [48]. Besides, model fine-tuning is essential for adapting pre-trained models to new datasets by continuing the training process on task-specific data, which enhances performance and generalization capabilities[49].

Existing datasets in this field have many limitations. While datasets like UCF101[50] and ActivityNet [51] cover general human activities, they may not

adequately capture the nuances and complexities of political violence-related events. Datasets such as RWF-2000[52], Hockey Fight[53], and Surveillance Camera Fight Dataset[54] focus on violence detection. Still, they are limited in scope, covering specific types of violence like real-world fights, hockey fights, and surveillance footage. The Real Life Violence Situations(RLVS) dataset[55] provides violent incidents, but it is not specifically focused on political violence and doesn't include a wide range of actions. Understanding and detecting political violence is crucial for applications in areas like security, conflict monitoring, and social science research. Developing effective computer vision models to classify different types of political violence can support early warning systems, event monitoring, and policy decision-making. A comprehensive dataset on political violence can contribute to advancing the state-of-the-art in this field and enable the development of more robust and accurate video classification models.

Keyframe extraction from video is crucial for model training as it allows for selecting representative frames that capture the essential content and dynamics of the video, thereby reducing the computational load and improving the efficiency of training deep learning models. One common method for keyframe extraction is shot boundary detection, which identifies transitions between scenes or shots within a video. However, this method often struggles with gradual transitions and can miss subtle changes in scenes[56]. Optimal frame selection using frame difference involves analyzing sequential frames to identify slight pixel changes, which indicate consistent contextual content. However, this method can supply the model with redundant information, potentially impairing its performance when trained on a varied visual dataset[57]. Another approach is clustering-based extraction, where frames are grouped into clusters based on visual similarity, and keyframes are selected from each cluster. This method may not always accurately represent the temporal dynamics of the video[58]. Adding temporal features with pixel features may stabilize the clusters and increase their effectiveness.

In particular, the study presents a holistic approach in the identification of gaps in existing video classification models and introduces a new political violence dataset specifically focusing on the problem domain, fine-tunes the MoViNet-A0 model to achieve state-of-the-art accuracy on political violence datasets, and implements a custom key frame extraction algorithm that exploits both temporal and pixel features during training. The main research question that guides this work is:

“How can we develop an efficient system for detecting violence in videos by addressing limited dataset diversity, unstable and incoherent keyframe selection in traditional K-means clustering, and computational overhead for real-time ap-

plications?” . The selection of MoViNet-A0 is justified by its superior balance of computational efficiency and performance, making it well-suited for resource-constrained smart city environments and real-time video analysis. Additionally, the custom keyframe extraction algorithm, by integrating temporal and pixel-level features, ensures the identification of the most informative frames, thereby improving the detection of subtle violent incidents. These methodological decisions collectively aim to establish an accurate, efficient, and scalable system for the detection of political violence in video data.

2.7 Conclusion

This chapter covers the history of video violence detection from the old hand-crafted feature based-methods until the advanced deep learning based approach. Meanwhile, the traditional methods, such as Motion SIFT and Violent Flow, are based on hand-crafted features, which is not scalable and costly to tune the level of complexity. Deep learning innovations such as 3D CNNs perform well in spatial temporal modelling while curbing computational overheads. 2D CNNs, although lighter, have difficulties in extracting temporal dynamics, resulting in hybrid models such as 2D CNN-LSTM and two-stream CNN, which process spatial and temporal information separately. Yet challenges yet remain, such as the complexity of models, dependence on quality of data and ability to model dependencies at long ranges. The lightweight model, MoViNet-A0, provides a trade-off for real time applications. Available datasets are not oriented towards political violence, hence the proposed Political Violence Dataset. Shifting, which is a keyframe extraction problem, like unstable clustering is solved by a new temporal-pixel feature algorithm. Doing this first, allows us to lay the ground work for a scalable (both accurate and effective) violence detection system by spotting holes in the space of both the types of data we use and the computational efficiency of our methods.

Chapter 3

Methodology

3.1 Introduction

This chapter presents a comprehensive methodology for detecting and classifying political violence in video footage, integrating the development and analysis of a specialized Political Violence Dataset with a detailed explanation of the proposed detection approach.

3.2 Overview of Methodology

The Figure 3.1 illustrates the workflow for developing a political violence detection system. The process begins with **Data Curation**, sourcing diverse video data from social media platforms such as Facebook, YouTube etc. to ensure real-world relevance. Next, **Annotation** involves manually labeling videos into four input class categories—Shooting, Clash, Fire, and Normal—based on content, establishing ground truth for training. **Preprocessing** follows, applying techniques like histogram equalization to enhance video quality and improve feature clarity. In **Feature Extraction**, spatial and temporal features are derived to capture the dynamics of violent events, represented as a feature vector $x = [x_1, x_2, \dots, x_d]$. Finally, **Keyframe Extraction** identifies the most representative frames using an enhanced algorithm integrating temporal and pixel-level features, reducing computational load while preserving critical information for classification. This pipeline ensures efficient and accurate detection of political violence, aligning with the chapter’s focus on dataset development and algorithmic innovation for real-time smart city surveillance.

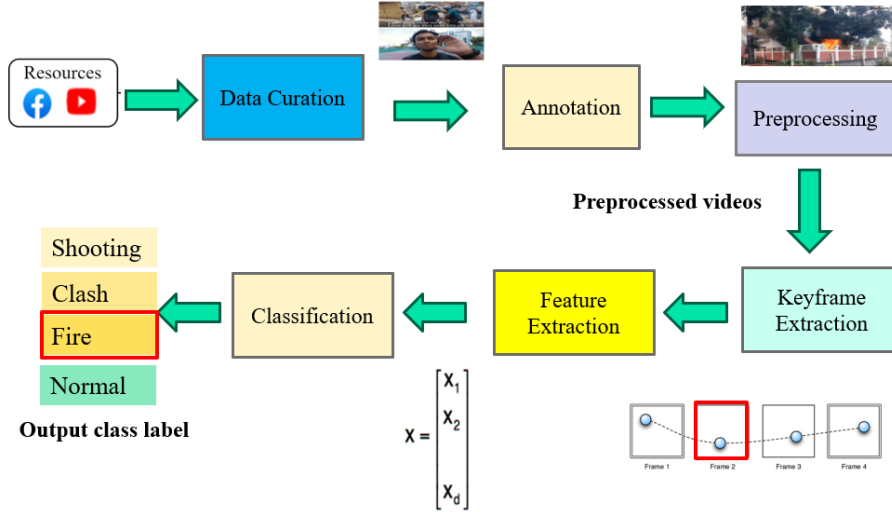


Figure 3.1: Overview of the violence detection system workflow.

3.3 Definition of the Task

The Political Violence Dataset is a comprehensive collection designed to analyze and detect various forms of violence in real-world political contexts. Unlike existing datasets that primarily focus on sports-specific altercations or controlled environments, our dataset captures the complexity and nuances of public gatherings during political incidents. This dataset encompasses a wide range of violent scenarios, providing a more representative sample of real-life situations.

The dataset is divided into four distinct classes:

- **Normal:** This class includes footage of individuals engaging in their daily activities without any signs of violence. It serves as a baseline for comparison against more violent scenarios.
- **Clash:** Instances categorized as “Clash” involve potential or actual confrontations, including hand-to-hand combat, the presence of clubs or other weapons, unusual movements indicative of unrest, police interactions with civilians, and vandalism.
- **Fire:** This class captures events where fires break out in various settings, such as transportation vehicles, institutions, or residential areas. The presence of fire signifies a critical escalation in violent situations.
- **Shooting:** Events in this category involve individuals who aim guns at others or large crowds. This class represents one of the most severe forms of violence and is crucial for understanding the dynamics of armed conflicts during political events.

The Political Violence Dataset stands apart from existing datasets due to its focus on large public gatherings and live political incidents. While many datasets are binary (violent vs. nonviolent), our multiclass approach allows for a more nuanced understanding of different types of violence. The dataset also highlights actions that may not involve direct physical contact but still represent significant threats, such as aggressive posturing or the display of weapons.

3.4 Dataset Developmet Steps

The Political Violence Dataset was created through a rigorous process, orchestrated through a sequence of data curation, annotation and subsequent preprocessing steps. The following subsections describe the meticulous methodologies utilized for collecting, annotating, and preprocessing the video data to ensure effective model training.

3.4.1 Data Curation

Figure 3.2 presents an overview of the data acquisition process for the Political Violence dataset, illustrating the distribution of video content gathered from various social media platforms, including Facebook, YouTube, and Instagram. The reliance on these platforms as data sources is justified by their widespread use and extensive repositories of user-generated content, which provide a rich and diverse pool of real-world video footage relevant to political violence. Facebook, contributing 60% of the dataset, serves as the primary source due to its vast user base and the prevalence of event-related videos shared by individuals and groups. YouTube, accounting for 30%, offers a significant volume of publicly accessible videos, often including detailed documentation of political incidents, making it a valuable secondary source. Instagram, while contributing a smaller portion at 10%, complements the dataset with its focus on short-form, visually rich content that captures spontaneous moments of political unrest. Together, these platforms ensure a broad and representative sample of video data, reflecting varied contexts and perspectives essential for training robust violence detection models.

3.4.2 Annotation

The annotation process for the Political Violence dataset ensures high-quality labeling through a structured workflow, as depicted in Figure 3.3. Initially, each video is assessed to determine if it depicts normal behavior; if so, it is labeled

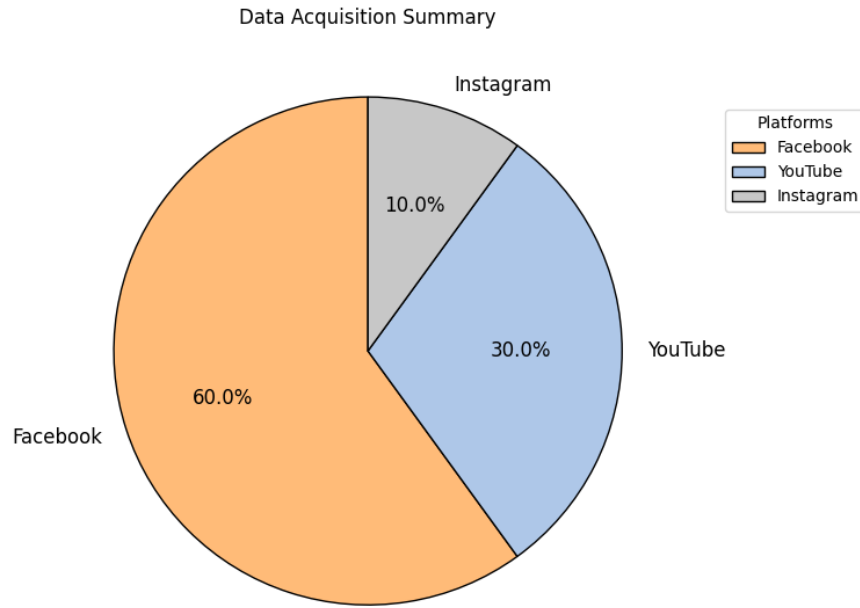


Figure 3.2: Data acquisition summary of Political Violence dataset. This pie chart illustrates the distribution of data acquisition across three major social media platforms: Facebook (60%), YouTube (30%), and Instagram (10%). The chart highlights Facebook as the primary source of data, followed by YouTube and Instagram.

as “Normal.” If not, annotators identify the reasons for violence—such as confrontation, hand-to-hand combat, weapons, vandalism, fire outbreaks, or shooting—based on predefined criteria. Videos that do not match these definitions are discarded. For videos exhibiting violence, annotators check if the reasons point to multiple classes; if not, a fine-grained label (Fire, Clash, or Shooting) is assigned. To maintain consistency, every video is manually labeled by two annotators following uniform guidelines, and in cases of disagreement, an expert facilitates resolution through discussion to finalize the label, ensuring reliability and accuracy in the dataset.

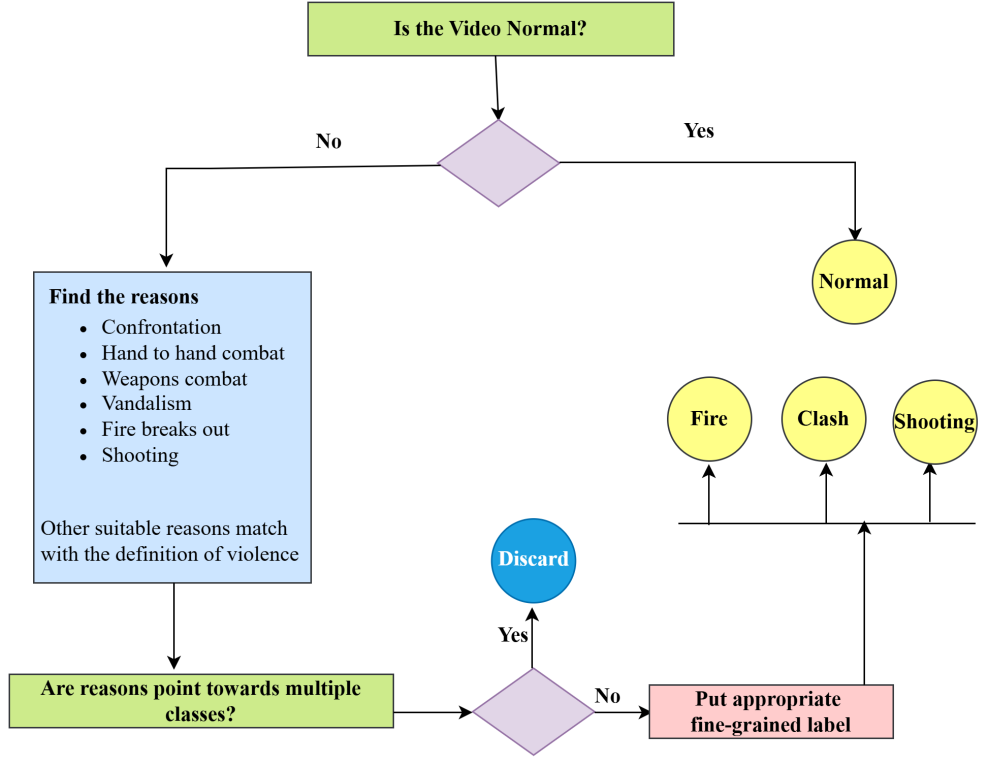


Figure 3.3: Data Annotation process.

Algorithm 1 Political Violence Dataset Labeling Process

Require: Set of videos $M = \{m_1, m_2, \dots, m_{480}\}$ and three annotators $A = \{A_1, A_2, A_3\}$ providing labels for each video

Ensure: Dataset with violence category annotations $PV_Labels = \{(m_i, y_i)\}_{i=1}^{480}$

```

1:  $PV\_Labels \leftarrow \emptyset$ 
2: for  $i \leftarrow 1$   $480$  do
3:    $L_i \leftarrow \{y_{i1}, y_{i2}, y_{i3}\}$  *Labels from annotators  $A_1, A_2, A_3$  for video  $m_i$ 
4:   for each label  $y \in \{Normal, Clash, Fire, Shooting\}$  do
5:      $Count(y) \leftarrow \sum_{j=1}^3 \mathbb{I}(y_{ij} = y)$  *Count occurrences of each label
6:   end for
7:    $y_{majority} \leftarrow \arg \max_y Count(y)$ 
8:   if  $Count(y_{majority}) \geq 2$  then
9:      $PV\_Labels \leftarrow PV\_Labels \cup \{(m_i, y_{majority})\}$ 
10:  else
11:    Expert Resolution:
12:     $y_{final} \leftarrow ExpertReview(L_i)$ 
13:     $PV\_Labels \leftarrow PV\_Labels \cup \{(m_i, y_{final})\}$ 
14:  end if
15: end for
16:  $\kappa \leftarrow CohenKappa(A)$  *Calculate Cohen's Kappa Coefficient for inter-annotator agreement
17: return  $PV\_Labels$ 

```

To ensure the Political Violence Dataset is accurately and consistently la-

beled, we implemented a structured annotation process guided by established best practices [59]. Three trained annotators, each with a background in computer engineering, independently reviewed and categorized each of the 480 videos into one of four classes: Normal, Clash, Fire, or Shooting. Each video was evaluated by all three annotators to promote reliability. If at least two annotators agreed on the classification, the label was assigned directly. In the rare instances where all three annotators disagreed, an expert reviewer consulted the annotators’ reasoning and applied Algorithm 1 to determine the final label. This algorithm systematically handles label assignments and resolves conflict to maintain consistency across the dataset. To assess the quality of the annotations, we calculated the inter-annotator agreement using Cohen’s Kappa coefficient, achieving a mean score of 0.90, which indicates strong agreement. This comprehensive approach, combining triple annotation, expert resolution, and statistical quality checks, ensures the dataset’s reliability and robustness for analyzing political violence in real-world contexts.

3.4.3 Preprocessing

Preprocessing serves as a pivotal step in the analytical pipeline, markedly amplifying the fidelity and consistency of the visual data, which, in turn, critically underpins the effectiveness and computational efficiency of the deep learning algorithms applied thereafter [60] - [61].

The raw video frames can be represented as a matrix of pixel intensities I , as shown in Equation 3.1:

$$I = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,n} \end{bmatrix} \quad (3.1)$$

where $p_{i,j}$ denotes the intensity value of the pixel at position (i, j) , and m and n are the dimensions of the frame.

The histogram $H(I)$ of the image I is defined as:

$$H(I) = \{h_k : k = 0, 1, 2, \dots, L - 1\} \quad (3.2)$$

where h_k is the number of pixels with intensity value k , and L is the number of possible intensity levels.

Histogram equalization aims to transform the pixel intensities to achieve a uniform distribution across the intensity levels. The transformation function

$T : [0, L - 1] \rightarrow [0, L - 1]$ for each pixel intensity $p_{i,j}$ can be defined as Equation 3.3:

$$T(p_{i,j}) = (L - 1) \sum_{k=0}^{p_{i,j}} h_k / N \quad (3.3)$$

where $N = m \times n$ is the total number of pixels in the image.

After applying histogram equalization, we obtain an enhanced image I' :

$$I' = T(I) \quad (3.4)$$

This enhanced image improves feature visibility by increasing contrast among pixel intensities. The discernibility of critical features necessary for accurate violence detection can be mathematically represented as an increase in signal-to-noise ratio (SNR):

$$\text{SNR}_{\text{post}} > \text{SNR}_{\text{pre}} \quad (3.5)$$

Thus, by applying histogram equalization[62] to the raw video frames, we provide our deep learning algorithms with improved input data characterized by enhanced features necessary for reliable classification tasks. Figure 3.4 presents a side-by-side comparison of frames before and after applying histogram equalization. The enhanced frames exhibit increased clarity and improved visibility of details, crucial for the subsequent feature extraction phase.



Figure 3.4: Comparison of video frames pre- and post-histogram equalization: The left column shows the original frames as captured by mobile cameras, while the right column demonstrates the clarity achieved through histogram equalization.

Following enhancement, let I' represent the enhanced video frame with pixel intensities. The normalization process aims to scale these pixel intensities to a

consistent range, typically $[0, 1]$.

Given an enhanced frame I' represented as Equation 3.6:

$$I' = \begin{bmatrix} p'_{1,1} & p'_{1,2} & \cdots & p'_{1,n} \\ p'_{2,1} & p'_{2,2} & \cdots & p'_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p'_{m,1} & p'_{m,2} & \cdots & p'_{m,n} \end{bmatrix} \quad (3.6)$$

where $p'_{i,j}$ denotes the intensity value of the pixel at position (i, j) .

The normalization function $N : I' \rightarrow I''$ for each pixel intensity $p'_{i,j}$ can be defined as Equation 3.7:

$$I''_{i,j} = \frac{p'_{i,j} - \min(I')}{\max(I') - \min(I')} \quad (3.7)$$

where $I''_{i,j}$ is the normalized pixel intensity, and $\min(I')$ and $\max(I')$ are the minimum and maximum intensity values in the enhanced frame I' , respectively.

This transformation ensures that all pixel intensities are scaled to the range $[0, 1]$:

$$0 \leq I''_{i,j} \leq 1 \quad (3.8)$$

Normalization aids in stabilizing the learning process by mitigating issues related to differing scales of input features. This consistent scaling improves model performance by allowing faster convergence during training.

3.5 Data Annotation Quality

The quality and reliability of annotations in the Political Violence Dataset were assessed using Cohen’s Kappa (κ), a statistical measure that evaluates inter-annotator agreement while accounting for chance agreement. Cohen’s Kappa is defined as:

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \quad (3.9)$$

where P_o represents the observed agreement between annotators, and P_e denotes the expected agreement by chance. A κ value between 0.81 and 1.00 indicates “almost perfect” agreement, reflecting high reliability in the annotation process.

The Political Violence Dataset comprises 480 labeled video clips categorized into four classes: Normal, Fire, Clash, and Shooting. The Cohen’s Kappa scores for each class, along with the mean across all labels, are presented in Table 3.1.

The mean κ value of 0.90 signifies an almost perfect level of agreement among annotators, demonstrating the consistency and validity of the fine-grained label annotations.

Table 3.1: Cohen’s Kappa Scores for the Political Violence Dataset Annotations

Class	κ -score	Mean
Normal	0.87	0.90
Fire	0.92	
Clash	0.87	
Shooting	0.94	

The high inter-annotator agreement, as evidenced by the mean κ score of 0.90, underscores the robustness of the Political Violence Dataset as a reliable benchmark for training and evaluating automated political violence detection models. This consistency ensures that the dataset can effectively support applications in security, conflict monitoring, and policy decision-making, where precise and dependable annotations are critical.

3.6 Dataset Statistics

Figure 3.5 shows samples from the Political Violence Dataset, illustrating instances of Clash, Fire, Shooting, and Normal activities.

The Political Violence Dataset is a collection of videos made to help train and check models that spot political violence. It has 480 video clips, split evenly into four groups: Clash, Fire, Shooting, and Normal. Each group has 140 clips, so both violent and non-violent videos are equally included. The dataset is divided into three parts: a training set, a validation set, and a test set. The training set teaches the model to find violence patterns. The validation set helps adjust the model to work better. The test set checks how well the model handles new videos it hasn’t seen before. Table 3.2 shows how the clips are split across these parts for each group.

As shown in Table 3.2, the dataset is split into 360 training samples, 88 validation samples, and 112 test samples, maintaining a consistent ratio across all classes. Specifically, each class contributes 90 samples to the training set, 22 samples to the validation set, and 28 samples to the test set. This distribution ensures that the model is trained on a substantial portion of the data (75% of the total samples), while the validation set (approximately 18%) allows for hyperparameter optimization, and the test set (approximately 23%) provides a robust



Figure 3.5: Samples from the Political Violence dataset: (a) Clash; (b) Fire; (c) Shooting; (d) Normal class.

Table 3.2: Distribution of Video Clips in the Political Violence Dataset Across Training, Validation, and Test Sets

Action	Training	Validation	Test	Total
Clash	90	22	28	140
Fire	90	22	28	140
Shooting	90	22	28	140
Normal	90	22	28	140
Total Samples	360	88	112	480

evaluation of the model’s generalization to unseen data. The balanced class distribution mitigates the risk of bias toward any particular category, enabling the model to learn and evaluate the characteristics of political violence and normal scenarios equitably.

Table 3.3 summarizes the characteristics of various datasets available for video violence detection.

The Political Violence dataset differs from datasets like RWF-2000, Hockey, RLVS, and Surveillance in key ways. It features four action categories (Normal, Clash, Fire, Shooting) versus the binary (Violent/Non-Violent) split in oth-

Dataset	RWF-2000 [52]	Hockey [53]	RLVS [55]	Surveillance [54]	Political Violence
Resolution	Variable	360×360	Variable	Variable	Variable
Min Clip Duration (s)	0.32	1.6	2	2	2
Max Clip Duration (s)	5	1.96	2	2	18
FPS	25	25	15	25	30
Actions	2	2	2	2	4

Table 3.3: Characteristics of various datasets for video violence detection.

ers—RWF-2000 , Hockey , RLVS , Surveillance —offering finer granularity for political violence analysis. Its maximum clip duration of 18 seconds far exceeds RWF-2000 (5s), Hockey (1.96s), RLVS (2s), and Surveillance (2s), capturing evolving events with more context. A 30 FPS frame rate, higher than RWF-2000 (25), Hockey (25), RLVS (15), and Surveillance (25), enhances temporal detail for fast-paced actions. With 480 samples, it balances size and specificity, focusing on political contexts unlike the broader or domain-specific scopes of others, making it uniquely suited for targeted detection models.

3.7 Problem Formulation

In this study, we address the task of multiclass violence video classification, where the objective is to accurately categorize videos into multiple violence-related classes based on their content. Let us define the components of this classification task systematically.

Consider a dataset comprising a collection of videos, denoted by the set in Equation 3.10,

$$V = \{v_1, v_2, \dots, v_{|V|}\}, \quad (3.10)$$

where each video v_i is an individual instance within the dataset, and $|V|$ represents the total number of videos. Each video v_i is characterized by a feature vector in Equation 3.11,

$$\mathbf{X}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} \in \mathbb{R}^d \quad (3.11)$$

where $\mathbf{X}_i$ is a d -dimensional real-valued vector encapsulating various attributes or features extracted from the video. The dimensionality d reflects the number of distinct features utilized to represent each video, which may include visual

descriptors, motion patterns, audio signals, and other relevant metadata.

We define a set of classes C representing different categories of violence in Equation 3.12,

$$C = \{c_1, c_2, \dots, c_{|C|}\}, \quad (3.12)$$

where each c_j corresponds to a specific violence-related category, and $|C|$ denotes the total number of such classes. Examples of these classes might include physical assault, weapon usage, verbal threats, and other forms of violent behavior.

For each video v_i , there is an associated label vector

$$\mathbf{Y}_i \in \{0, 1\}^{|C|}, \quad (3.13)$$

which indicates the membership of the video across the defined classes. Specifically, each element Y_{ij} of the label vector $\mathbf{Y}_i$ is a binary indicator where

$$Y_{ij} = \begin{cases} 1 & \text{if video } v_i \text{ belongs to class } c_j, \\ 0 & \text{otherwise.} \end{cases}$$

This formulation supports multi-label classification, allowing a single video to be associated with multiple violence-related classes simultaneously.

The primary goal is to learn a mapping function

$$f : \mathbb{R}^d \rightarrow [0, 1]^{|C|}, \quad (3.14)$$

which predicts the probability of each class c_j being present in a given video based on its feature vector $\mathbf{X}_i$. Mathematically, the function f maps the input feature vector $\mathbf{X}_i$ to an output vector $f(\mathbf{X}_i)$, where each component $f(\mathbf{X}_i)_j$ represents the predicted probability that video v_i belongs to class c_j .

To train the model, we define a loss function that quantifies the discrepancy between the predicted probabilities and the actual class memberships. A commonly used loss function for multiclass classification tasks is the sparse categorical cross-entropy loss, which for a single video v_i is expressed as

$$L(f(\mathbf{X}_i), \mathbf{Y}_i) = - \sum_{j=1}^{|C|} Y_{ij} \log(f(\mathbf{X}_i)_j). \quad (3.15)$$

In this equation, the loss L accumulates the negative log-likelihood of the true classes present in the video, effectively penalizing the model for incorrect predictions.

Given a dataset of videos

$$D = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^N, \quad (3.16)$$

where N is the total number of videos, the objective is to minimize the overall loss across all samples to enhance the model’s ability to generalize to unseen data. This is formalized by the objective function

$$J(f) = \frac{1}{N} \sum_{i=1}^N L(f(\mathbf{X}_i), \mathbf{Y}_i), \quad (3.17)$$

which represents the average loss over the entire dataset. Minimizing $J(f)$ entails adjusting the parameters of the mapping function f such that the predicted probabilities $f(\mathbf{X}_i)_j$ closely align with the actual class memberships Y_{ij} for all videos in the dataset.

3.8 Keyframe Extraction

Keyframe extraction plays a pivotal role in video violence classification by reducing the computational complexity and redundancy inherent in processing full video sequences, while preserving the essential spatio-temporal information needed to distinguish classes such as “Normal,” “Clash,” “Fire,” and “Shooting.” Videos typically contain a large number of frames, many of which are highly correlated or contain minimal changes, making exhaustive frame-by-frame analysis inefficient. By selecting keyframes—representative frames that capture significant events or transitions (e.g., the onset of a clash or the appearance of fire)—the models focus computational resources on the most informative content. For instance, in the Vision Transformer, 12 keyframes are extracted from a variable-length RGB sequence to form a manageable input of shape $(12, 224, 224, 3)$, enabling the convolutional tokenizer and transformer encoder to efficiently process spatial and temporal features. Similarly, ConvLSTM and 2D CNN + LSTM/BiLSTM architectures benefit from keyframes to streamline the sequence $X = \{X_1, X_2, \dots, X_T\}$ into a subset that retains critical dynamics, such as motion patterns or intensity shifts, for recurrent processing. In MoViNet-A0, keyframe-based sub-clips enhance the causal stream buffer’s ability to model temporal dependencies without overwhelming memory constraints. Moreover, keyframe extraction improves robustness by emphasizing visually distinct moments, reducing noise from irrelevant frames, and facilitating the subsequent feature extraction and softmax classification stages across all models.

Algorithm 2 Keyframe Extraction Using K-Means Clustering with Temporal Features

Require: • Video V consisting of N frames $\{F_1, F_2, \dots, F_N\}$

• Number of keyframes/clusters K (e.g., $K = 12$)

Ensure: • Set of keyframes $\{F_{\text{key},1}, F_{\text{key},2}, \dots, F_{\text{key},K}\}$

1: **Initialize** an empty list for feature vectors: $\mathcal{F} = []$

2: **for** each frame F_i where $i = 1, 2, \dots, N$ **do**

3: Convert F_i from RGB to HSV color space.

4: Compute the color histogram $\phi_{\text{color}}(F_i)$ with h bins:

$$\phi_{\text{color}}(F_i) \in \mathbb{R}^h$$

5: Normalize the color histogram to account for varying lighting conditions:

$$\phi_{\text{color}}^{\text{norm}}(F_i) = \frac{\phi_{\text{color}}(F_i)}{\|\phi_{\text{color}}(F_i)\|_1}$$

6: Extract the temporal feature $\phi_{\text{temp}}(F_i)$:

$$\phi_{\text{temp}}(F_i) = \frac{i}{N}$$

7: Concatenate color and temporal features to form the final feature vector:

$$\phi(F_i) = [\phi_{\text{color}}^{\text{norm}}(F_i), \phi_{\text{temp}}(F_i)] \in \mathbb{R}^d, \quad d = h + 1$$

8: Append $\phi(F_i)$ to the feature list $\mathcal{F}$

9: **end for**

10: **Standardize** the feature vectors to have zero mean and unit variance:

$$\phi'(F_i) = \text{Standardize}(\phi(F_i)) \quad \forall F_i \in V$$

11: **Initialize** K-Means clustering with K clusters and random centroids.

12: **Perform** K-Means clustering on the standardized feature vectors $\{\phi'(F_i)\}_{i=1}^N$:

$$\min_{C_1, C_2, \dots, C_K} \sum_{k=1}^K \sum_{x \in C_k} \|x - \mu_k\|^2$$

where μ_k is the centroid of cluster C_k .

13: **Initialize** an empty set for keyframes: $F_{\text{key}} = \emptyset$

14: **for** each cluster C_k where $k = 1, 2, \dots, K$ **do**

15: Identify the frame $F_{\text{key},k}$ closest to the centroid μ_k :

$$F_{\text{key},k} = \arg \min_{F_i \in C_k} \|\phi'(F_i) - \mu_k\|$$

16: Add $F_{\text{key},k}$ to the keyframe set F_{key}

17: **end for**

18: **Sort** the keyframes in F_{key} based on their original frame indices to preserve temporal order:

$$F_{\text{key}} = \text{Sort}(\{F_{\text{key},1}, F_{\text{key},2}, \dots, F_{\text{key},K}\})$$

19: **Return** the sorted set of keyframes F_{key}

The proposed algorithm for keyframe extraction, as detailed in Algorithm 2, builds upon the foundational principles of K-means clustering [63]-[64] and introduces a novel modification by integrating both color histograms and temporal features to optimize the selection of representative frames from a video sequence. Unlike traditional K-means clustering, which relies solely on static visual features such as pixel intensities or color distributions, this enhanced approach incorporates temporal dynamics inherent in video data, ensuring that the clustering process captures both the visual content and the sequential context of frames. This results in more coherent and stable cluster assignments across the video timeline, addressing the limitations of conventional methods that often produce redundant or temporally disjointed keyframes.

The traditional K-means algorithm minimizes the within-cluster variance based on visual features, defined as:

$$J = \sum_{i=1}^K \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (3.18)$$

where $x \in \mathbb{R}^h$ is a frame's feature vector (e.g., a color histogram with h bins), $\mu_i \in \mathbb{R}^h$ is the centroid of cluster C_i , and K is the number of clusters. This objective optimizes for visual similarity, often grouping frames with minimal variation (e.g., consecutive frames of a static scene), leading to redundant keyframes due to its inability to capture temporal dynamics.

In contrast, the proposed algorithm constructs a hybrid feature vector that combines a normalized color histogram in the HSV color space with a temporal feature representing the frame's relative position within the video. For a frame F_i in a video with N frames, the color histogram $h_i \in \mathbb{R}^h$ is computed with a specified number of bins (e.g., 256) and normalized to ensure scale invariance:

$$h_i(j) = \frac{\text{count}_j}{\sum_{k=1}^h \text{count}_k}, \quad j = 1, \dots, h \quad (3.19)$$

where count_j is the number of pixels in bin j . The temporal feature is defined as the normalized frame index:

$$\phi_{\text{temp}}(F_i) = \frac{i}{N} \in [0, 1] \quad (3.20)$$

Additionally, to capture motion dynamics, the algorithm incorporates temporal features such as frame differences ($\Delta I_t = |I_t - I_{t-1}|$), which provide a robust

representation of temporal evolution. The composite feature vector is:

$$x' = [h_i, \phi_{\text{temp}}(F_i), x_{\text{temporal}}] \in \mathbb{R}^{h+1+m} \quad (3.21)$$

where $h_i \in \mathbb{R}^h$ is the normalized color histogram, $\phi_{\text{temp}}(F_i) \in \mathbb{R}$ is the temporal position, $x_{\text{temporal}} \in \mathbb{R}^m$ includes motion-based features (e.g., frame differences), and $h + 1 + m$ is the augmented feature space dimensionality. To balance the contributions of visual and temporal components, the feature vectors are standardized:

$$x'_j = \frac{x'_j - (x'_j)}{(x'_j)}, \quad j = 1, \dots, h + 1 + m \quad (3.22)$$

where (x'_j) and (x'_j) are the mean and standard deviation of the j -th feature dimension across all frames.

The new objective function for K-means clustering is:

$$J' = \sum_{i=1}^K \sum_{x' \in C_i} \|x' - \mu'_i\|^2 \quad (3.23)$$

where $\mu'_i \in \mathbb{R}^{h+1+m}$ is the centroid in the augmented feature space. The inclusion of temporal features increases the feature space dimensionality, enabling better separation of frames with similar visual content but distinct temporal behaviors, thus reducing redundancy in keyframe selection.

To optimize the number of clusters, the silhouette score is employed:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (3.24)$$

where $a(i) = \frac{1}{|C_i|-1} \sum_{j \in C_i, j \neq i} \|x'_i - x'_j\|^2$ is the average intra-cluster distance for frame i , and $b(i) = \min_{k \neq C_i} \frac{1}{|C_k|} \sum_{j \in C_k} \|x'_i - x'_j\|^2$ is the average distance to the nearest neighboring cluster. The peak silhouette score at $K = 12$ (Figure 3.8) indicates optimal cluster separation, balancing the contributions of color histograms and temporal features.

Stability of Clustering: Clustering stability is critical for ensuring that keyframes reflect the sequential narrative of the video. Without temporal features, cluster assignments exhibit frequent transitions due to minor variations in color histograms, leading to high variance in cluster labels:

$$\sigma_{\text{labels}}^2 = \frac{1}{T} \sum_{t=1}^T (l_t - \bar{l})^2 \quad (3.25)$$

where $l_t \in \{1, \dots, K\}$ is the cluster label of frame t , $\bar{l} = \frac{1}{T} \sum_{t=1}^T l_t$ is the mean label (approximated as a continuous variable), and T is the number of frames. High σ_{labels}^2 indicates unstable clustering, as observed in Figure 4.1 and Figure 4.2 without temporal features, resulting in redundant keyframes.

With temporal features, the augmented feature vector x' incorporates sequential context, reducing abrupt label changes. The temporal components ($\phi_{\text{temp}}(F_i)$ and x_{temporal}) ensure that frames with similar motion patterns or temporal positions are grouped together, minimizing label variance:

$$\sigma_{\text{labels}}'^2 = \frac{1}{T} \sum_{t=1}^T (l'_t - \bar{l}')^2 \quad (3.26)$$

where l'_t is the cluster label with temporal features, and $\bar{l}' = \frac{1}{T} \sum_{t=1}^T l'_t$. The reduced $\sigma_{\text{labels}}'^2$ reflects stable clustering, as shown in Figure 4.1 and Figure 4.2, where smoother cluster transitions indicate lower variance.

Stability is further quantified by the transition probability matrix P , where $P_{ij} = P(l_{t+1} = j | l_t = i)$ represents the probability of transitioning from cluster i to cluster j . Without temporal features, P exhibits high entropy:

$$H(P) = - \sum_{i=1}^K \sum_{j=1}^K P_{ij} \log P_{ij} \quad (3.27)$$

With temporal features, the transition matrix P' has lower entropy, as $P'_{ii} \approx 1$, reflecting the smoother transitions in Figure 4.1 and Figure 4.2 (subplot c).

Necessity of Smoother Transitions : Smoother transitions ensure that keyframes maintain narrative continuity, critical for capturing significant moments (e.g., scene changes or action peaks) in violence detection. Abrupt transitions lead to disjointed keyframes, undermining temporal coherence. This is quantified by the temporal coherence metric:

$$C_{\text{temp}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \exp \left(- \frac{\|x'_{t+1} - x'_t\|^2}{\sigma^2} \right) \quad (3.28)$$

where $\|x'_{t+1} - x'_t\|^2$ is the squared Euclidean distance between consecutive frames' feature vectors, and σ^2 is a scaling parameter. Higher C_{temp} indicates smoother transitions, as temporal features reduce the distance between consecutive frames.

The smoothness is also modeled by the expected change in cluster assignments:

$$\Delta_{\text{trans}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{I}(l'_t \neq l'_{t+1}) \quad (3.29)$$

where $\mathbb{I}$ is the indicator function. Temporal features reduce Δ_{trans} , ensuring keyframes capture the continuous evolution of events.

Superiority over K-means: The proposed algorithm’s superiority stems from the augmented feature space $\mathbb{R}^{h+1+m}$, which enables better separation of frames with similar color histograms but distinct temporal behaviors. The traditional K-means objective ($\|x - \mu_i\|^2$) is limited to visual similarity, often grouping redundant frames. The proposed objective ($\|x' - \mu'_i\|^2$) incorporates temporal dynamics, increasing inter-frame distances for temporally distinct frames. Variance decomposition illustrates this:

$$\begin{aligned} \|x' - \mu'_i\|^2 &= \|h_i - \mu_{i,\text{hist}}\|^2 + \|\phi_{\text{temp}} - \mu_{i,\text{temp}}\|^2 \\ &\quad + \|x_{\text{temporal}} - \mu_{i,\text{temporal}}\|^2 \\ &\quad + 2\text{Cov}(h_i, \phi_{\text{temp}}, x_{\text{temporal}}) \end{aligned} \quad (3.30)$$

The temporal variance component:

$$\text{Var}_{\text{temp}} = \sum_{x' \in C_i} (\|\phi_{\text{temp}} - \mu_{i,\text{temp}}\|^2 + \|x_{\text{temporal}} - \mu_{i,\text{temporal}}\|^2) \quad (3.31)$$

ensures coherent clustering, reducing redundancy.

The Davies-Bouldin (DB) index quantifies cluster separation, defined as:

$$DB = \frac{1}{K} \sum_{i=1}^K \max_{j \neq i} \left(\frac{s_i + s_j}{d_{ij}} \right) \quad (3.32)$$

where $s_i = \frac{1}{|C_i|} \sum_{x' \in C_i} \|x' - \mu'_i\|^2$ is the average intra-cluster distance for cluster C_i , and $d_{ij} = \|\mu'_i - \mu'_j\|^2$ is the squared Euclidean distance between centroids μ'_i and μ'_j . A lower DB indicates better cluster separation, with compact clusters and large inter-cluster distances. In the context of keyframe extraction, the proposed algorithm, which integrates color histograms and temporal features (e.g., normalized frame index $\phi_{\text{temp}}(F_i) = \frac{i}{N}$, frame differences $\Delta I_t = |I_t - I_{t-1}|$), achieves a lower DB compared to traditional K-means, as evidenced by the improved cluster separation in Figure 4.1 and Figure 4.2 (subplots a and b) [63]-[64].

For traditional K-means, the DB index was computed as $DB = 1.05$, reflecting moderate separation due to visual similarity alone. The proposed algorithm achieved a lower $DB = 0.62$, indicating superior cluster separation. These values confirm the effectiveness of the augmented feature space in reducing intra-cluster variance and increasing inter-cluster distances.

The silhouette coefficient serves as a discerning metric for evaluating the efficacy of clustering methodologies. For an individual data point i , its formulation



Figure 3.6: Extracted key frames for a Clash class sample video.

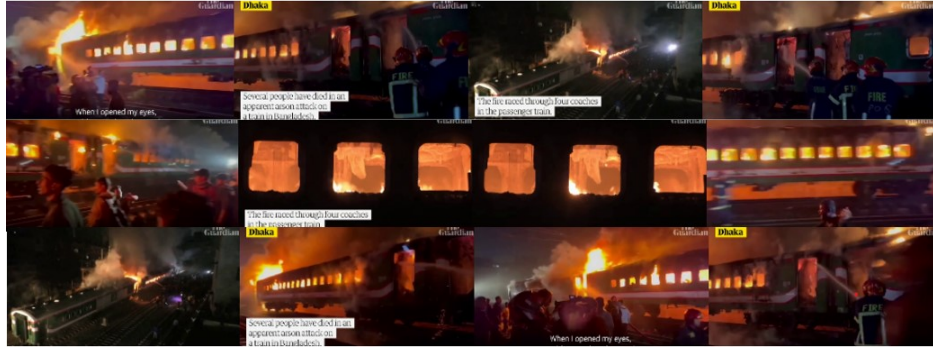


Figure 3.7: Extracted key frames for a Fire class sample video.

is articulated as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3.33)$$

where:

- $a(i)$: the mean Euclidean separation from i to all fellow constituents within its designated cluster, encapsulating the notion of intra-cluster coherence,
- $b(i)$: the minimal average distance from i to the constituents of any disparate cluster, embodying the principle of inter-cluster divergence.

The value of $s(i)$ oscillates between -1 and 1, with elevated magnitudes signifying superior clustering fidelity. The aggregate silhouette coefficient for a specified K is derived by computing the arithmetic mean of $s(i)$ across the entire dataset. In our investigation, the apex of this coefficient was observed at $K = 12$, as delineated in Figure 3.8.

3.9 Feature Extraction

Feature extraction is a critical step in video violence classification, enabling the identification of spatial and temporal patterns within video sequences to dis-

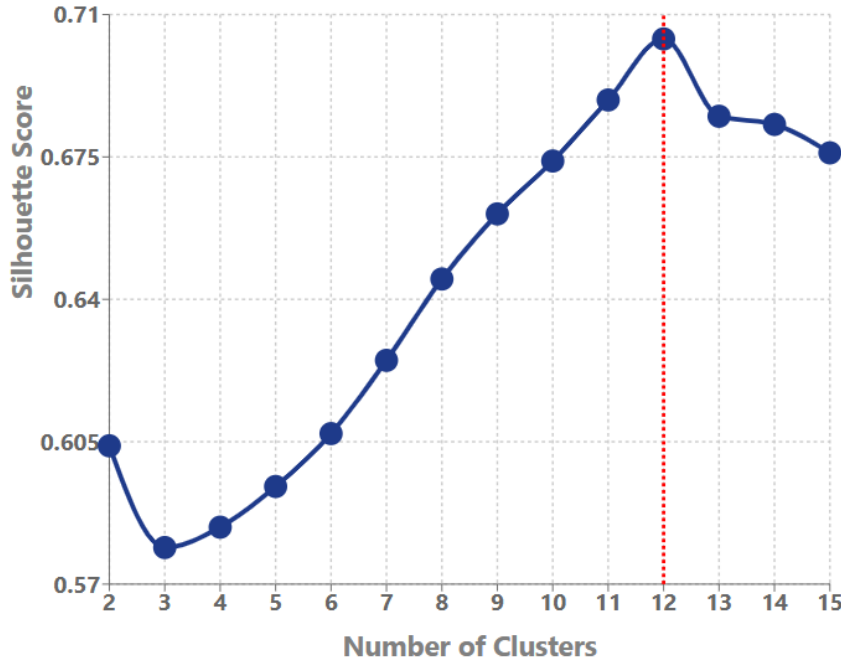


Figure 3.8: Silhouette score visualization: the silhouette scores calculated for various values of K (peak at $K = 12$).

tinguish between classes such as “Normal,” “Clash,” “Fire,” and “Shooting.” In the proposed architectures—ConvLSTM, 2D CNN + LSTM/BiLSTM, Vision Transformer, and MoViNet-A0—diverse strategies are employed to capture these patterns effectively. ConvLSTM integrates convolutional operations within recurrent LSTM cells to jointly model spatial details (e.g., fire textures) and temporal dynamics (e.g., clash escalation) across frames. The 2D CNN + LSTM/BiLSTM approach separates these tasks, using a 2D CNN (e.g., MobileNetV2) for spatial feature extraction per frame, followed by LSTM or BiLSTM for temporal dependency modeling. The Vision Transformer adapts a hybrid convolutional-transformer framework, tokenizing keyframes with a convolutional tokenizer and leveraging multi-head self-attention to encode spatio-temporal relationships. Meanwhile, MoViNet-A0 employs a lightweight, mobile-optimized design with causal convolutions and stream buffers to extract temporal features across sub-clips, enhancing efficiency and receptive field coverage.

3.9.1 ConvLSTM

The ConvLSTM architecture is engineered to meticulously dissect a video sequence denoted as $X = \{X_1, X_2, \dots, X_T\}$, wherein each frame $X_t \in \mathbb{R}^{H \times W \times C}$ encapsulates dimensions of height H , width W , and channels C . By orchestrat-

ing convolutional operations within LSTM cells, this framework adeptly captures intricate spatial configurations, such as the presence of flames, alongside temporal evolutions, exemplified by the intensification of confrontations. The ConvLSTM modulates information propagation through a sophisticated ensemble of gating mechanisms.

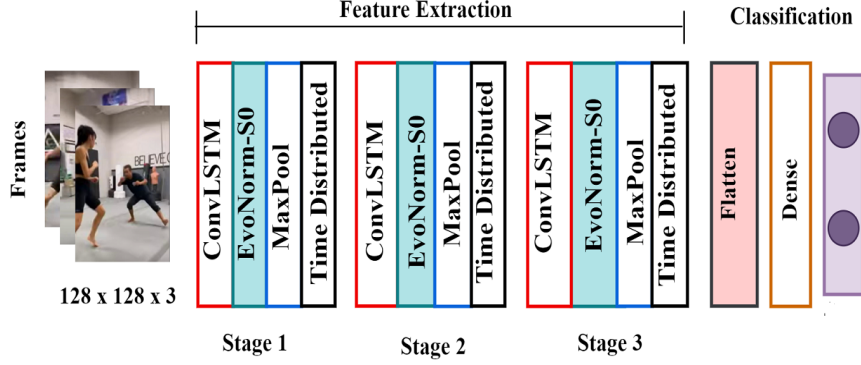


Figure 3.9: The ConvLSTM architecture for feature architecture.

The input gate, delineated in Equation (3.34), is formulated as:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i), \quad (3.34)$$

where $i_t \in \mathbb{R}^{H' \times W' \times D}$ represents the activation of the input gate at temporal instance t , dictating the extent to which novel input is assimilated into the cell state; σ denotes the sigmoid function, constraining outputs to the interval $[0, 1]$; $W_{xi} * X_t$ signifies the convolution of the current frame X_t with the input-to-input weight tensor W_{xi} , extracting salient spatial attributes; $W_{hi} * H_{t-1}$ encapsulates the convolution of the antecedent hidden state H_{t-1} with the hidden-to-input weight tensor W_{hi} , integrating historical spatial context; $W_{ci} \odot C_{t-1}$ involves the Hadamard product of the prior cell state C_{t-1} with the cell-to-input weight tensor W_{ci} , infusing memory-driven influences; and b_i constitutes the bias vector for the input gate.

Subsequently, the forget gate, articulated in Equation (3.35), is expressed as:

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f), \quad (3.35)$$

where $f_t \in \mathbb{R}^{H' \times W' \times D}$ embodies the forget gate activation, adjudicating which facets of the preceding cell state warrant retention or obliteration; $W_{xf} * X_t$ represents the convolution of the input X_t with the input-to-forget weight tensor W_{xf} , gauging the pertinence of contemporary input; $W_{hf} * H_{t-1}$ is the convolution of the prior hidden state H_{t-1} with the hidden-to-forget weight tensor W_{hf} , reflecting antecedent information; $W_{cf} \odot C_{t-1}$ entails the Hadamard product of

the previous cell state C_{t-1} with the cell-to-forget weight tensor W_{cf} , modulating memory preservation; and b_f serves as the bias vector for the forget gate.

The cell state update, explicated in Equation (3.36), is delineated as:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \quad (3.36)$$

where $C_t \in \mathbb{R}^{H' \times W' \times D}$ constitutes the refreshed cell state at time t , enshrining protracted memory; $f_t \odot C_{t-1}$ involves the Hadamard product of the forget gate f_t with the antecedent cell state C_{t-1} , determining retained information; $i_t \odot \tanh(\cdot)$ reflects the Hadamard product of the input gate i_t with the candidate cell state, governing the influx of novel data; $W_{xc} * X_t$ denotes the convolution of the input X_t with the input-to-cell weight tensor W_{xc} , distilling emergent spatial features; $W_{hc} * H_{t-1}$ is the convolution of the prior hidden state H_{t-1} with the hidden-to-cell weight tensor W_{hc} , embedding historical context; $\tanh$ represents the hyperbolic tangent function, normalizing the candidate state to $[-1, 1]$; and b_c is the bias vector for the cell state.

The output gate, as specified in Equation (3.37), is formulated thus:

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o), \quad (3.37)$$

where $o_t \in \mathbb{R}^{H' \times W' \times D}$ signifies the output gate activation, arbitrating which segments of the cell state are propagated to the output; $W_{xo} * X_t$ is the convolution of the input X_t with the input-to-output weight tensor W_{xo} , appraising the influence of the current input; $W_{ho} * H_{t-1}$ represents the convolution of the prior hidden state H_{t-1} with the hidden-to-output weight tensor W_{ho} , factoring in historical data; $W_{co} \odot C_t$ entails the Hadamard product of the current cell state C_t with the cell-to-output weight tensor W_{co} , incorporating memory contributions; and b_o is the bias vector for the output gate.

Ultimately, the hidden state, encapsulating spatio-temporal characteristics, is derived via Equation (3.38):

$$H_t = o_t \odot \tanh(C_t), \quad (3.38)$$

where $H_t \in \mathbb{R}^{H' \times W' \times D}$ constitutes the hidden state at time t , functioning as the ConvLSTM cell's output; $o_t \odot \tanh(C_t)$ represents the Hadamard product of the output gate o_t with the $\tanh$ -transformed cell state $\tanh(C_t)$, calibrating the final emission; and $\tanh$ ensures the cell state values are confined to $[-1, 1]$. The hidden state H_t , with dimensions contingent upon the convolutional kernel, undergoes subsequent processing through architectural layers such as Flatten and

Dense to facilitate classification tasks. This configuration is illustrated in Figure 3.9, comprising three ConvLSTM modules (Block 1 to Block 3), each integrating ConvLSTM, Evo-Norm-S0, MaxPool, and Time Distributed layers.

3.9.2 2D CNN + LSTM/BiLSTM

The 2D CNN + LSTM architecture is a two-stage approach designed for feature extraction and classification in video violence detection, effectively handling the Political Violence dataset’s four classes by separating spatial and temporal processing. This section outlines the methodology, including the mathematical formulations of the spatial feature extraction using a 2D CNN and the temporal modeling using an LSTM or bidirectional LSTM (BiLSTM), Figure 3.10.

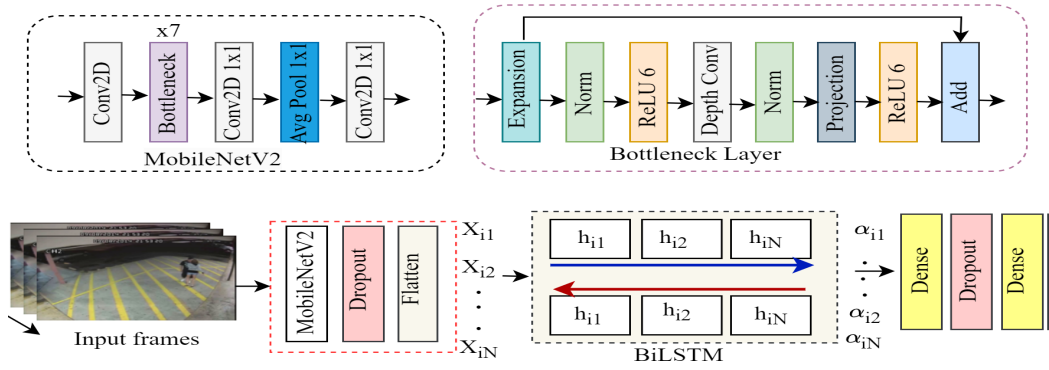


Figure 3.10: The 2D CNN + LSTM/ BiLSTM architecture .

3.9.2.1 Spatial Feature Extraction with 2D CNN

The process begins with a 2D convolutional neural network (CNN) extracting spatial features from each frame of a video sequence $X = \{X_1, X_2, \dots, X_T\}$, where $X_t \in \mathbb{R}^{H \times W \times C}$ represents a frame at time t with height H , width W , and channels C (e.g., 3 for RGB). The CNN processes each frame independently as follows:

$$F_t = \text{CNN}(X_t) \quad (3.39)$$

Here, $\text{CNN}(\cdot)$ denotes the convolutional operation (e.g., using MobileNetV2), involving convolution, pooling, and flattening, resulting in a feature vector $F_t \in \mathbb{R}^D$, where D is the dimensionality of the feature space. These spatial features $F = \{F_1, F_2, \dots, F_T\}$ capture patterns such as edges and textures within individual frames but do not account for temporal relationships.

3.9.2.2 Temporal Dynamics Elucidation via LSTM

The continuum of spatial feature representations F is subsequently channeled into a Long Short-Term Memory (LSTM) apparatus to meticulously delineate temporal interdependencies spanning sequential frames. The LSTM orchestrates the evolution of its internal states at each temporal juncture t through an intricate series of computational transformations, articulated as follows:

$$f_t = \sigma(W_f \cdot [H_{t-1}, F_t] + b_f) \quad (3.40)$$

where f_t constitutes the activation of the forget gate, adjudicating the selective expurgation of information from the antecedent cell state; $\sigma(x) = \frac{1}{1+e^{-x}}$ represents the sigmoid activation function, constraining outputs within the continuum $[0, 1]$; W_f and b_f denote the weight matrix and bias vector, respectively; and $[H_{t-1}, F_t]$ signifies the concatenation of the preceding hidden state H_{t-1} with the contemporaneous input F_t .

$$i_t = \sigma(W_i \cdot [H_{t-1}, F_t] + b_i) \quad (3.41)$$

where i_t embodies the input gate's activation, exercising dominion over the assimilation of nascent information, with W_i and b_i serving as its corresponding weight matrix and bias vector.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c \cdot [H_{t-1}, F_t] + b_c) \quad (3.42)$$

where C_t delineates the reconstituted cell state, perpetuating memory across time; $\odot$ indicates the Hadamard product, facilitating element-wise modulation; C_{t-1} represents the prior cell state; and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ imposes a hyperbolic tangent normalization upon the candidate update, regulated by the weight matrix W_c and bias vector b_c .

$$o_t = \sigma(W_o \cdot [H_{t-1}, F_t] + b_o) \quad (3.43)$$

where o_t manifests as the output gate's activation, arbitrating the selective dissemination of cell state constituents, governed by the parameters W_o and b_o , the weight matrix and bias vector, respectively.

$$H_t = o_t \odot \tanh(C_t) \quad (3.44)$$

where $H_t \in \mathbb{R}^{D_h}$ emerges as the hidden state, endowed with dimensionality D_h , encapsulating the temporal dynamics requisite for subsequent classification en-

deavors.

3.9.2.3 Bidirectional LSTM Extension

In some implementations, a bidirectional LSTM (BiLSTM) is used to process the sequence in both forward and backward directions, enhancing contextual understanding:

$$H_t = [\vec{H}_t, \overleftarrow{H}_t] \quad (3.45)$$

where $\vec{H}_t$ and $\overleftarrow{H}_t$ are the hidden states from the forward and backward passes, respectively, concatenated to form H_t .

3.9.3 Vision Transformer

The core architecture synergizes convolutional neural networks (CNNs) for spatial feature extraction with transformers for temporal modeling, classifying videos into categories such as “Fire,” “Clash,” “Normal,” and “Shooting.”

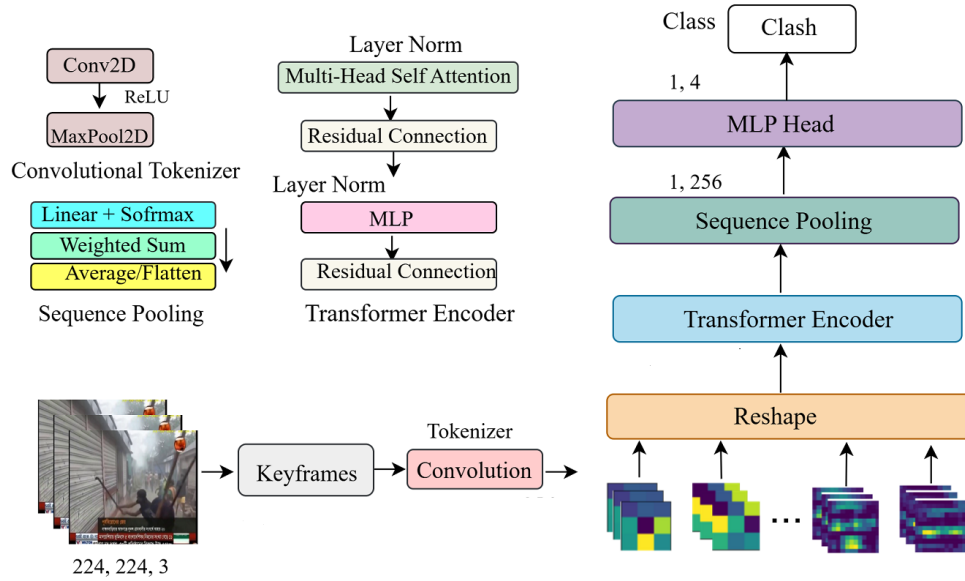


Figure 3.11: Overview of the Compact Convolutional Transformer (CCT) pipeline for video violence classification.

3.9.3.1 Input Processing and Convolutional Tokenizer

The input video, a variable-length RGB frame sequence, is processed to extract 12 keyframes (shape: (12, 224, 224, 3)). The convolutional tokenizer employs a

Conv2D layer with 64 filters, a 3x3 kernel, stride 1, and “same” padding, followed by ReLU activation:

$$x_1 = \max(0, W_1 * x_0 + b_1) \quad (3.46)$$

where $x_1 \in \mathbb{R}^{112 \times 112 \times 64}$, W_1 is the weight matrix, and b_1 is the bias. A MaxPool2D layer (2x2 pool size, stride 2) reduces spatial dimensions:

$$x_2 = \text{MaxPool}(x_1) \quad (3.47)$$

The input video, a variable-length RGB frame sequence, is processed to extract 12 keyframes (shape: (12, 224, 224, 3)). The convolutional tokenizer employs a Conv2D layer with 64 filters, a 3x3 kernel, stride 1, and “same” padding, followed by ReLU activation. After further processing, the output is tokenized into a sequence of shape (12, 112, 112, 256), adapting the original CCT for video analysis.

3.9.3.2 Transformer Encoder

The transformer encoder processes the tokenized sequence (12, N , d), where $N = 112 \times 112 = 12,544$ and $d = 256$, using multi-head self-attention (MHSA) with 4 heads:

$$\text{MHSA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_4)W^O \quad (3.48)$$

Each attention head is computed as:

$$\text{head}_i = \text{Softmax}\left(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{64}}\right)VW_i^V \quad (3.49)$$

Layer normalization stabilizes training:

$$\text{LN}(x) = \frac{x - \mu}{\sigma} \cdot \gamma + \beta \quad (3.50)$$

where μ and σ are the mean and standard deviation, and γ and β are learnable parameters. Residual connections preserve gradient flow:

$$x_{\text{out}} = x + \text{MHSA}(\text{LN}(x)) \quad (3.51)$$

The encoder outputs a sequence (12, N , d) with 5.94 million parameters, balancing efficiency and performance.

3.9.3.3 Sequence Pooling

Sequence pooling condenses the encoded sequence by averaging across the temporal dimension:

$$S_{\text{pooled}} = \frac{1}{12} \sum_{t=1}^{12} S_{\text{out},t} \quad (3.52)$$

producing a vector of shape (1, 256), efficiently summarizing spatio-temporal dynamics.

3.9.3.4 Classification Head

The classification head maps the pooled vector to 4 classes via a linear layer:

$$z = W_{\text{class}} S_{\text{pooled}} + b_{\text{class}} \quad (3.53)$$

where $W_{\text{class}} \in \mathbb{R}^{4 \times 256}$ and $b_{\text{class}} \in \mathbb{R}^4$. Softmax activation yields class probabilities:

$$\hat{y} = \text{Softmax}(z) \quad (3.54)$$

defined as:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^4 e^{z_j}} \quad (3.55)$$

Our model builds on the CCT’s hybrid convolutional-transformer framework but introduces significant divergences tailored for video violence classification. Unlike the original CCT, which focuses on static image classification, our approach extends to video by incorporating temporal modeling. This is achieved through sequence-based attention across keyframes, adapting the transformer’s self-attention mechanism:

$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (3.56)$$

where $d_k = 64$, contrasting with CCT’s spatial attention within a single image.

3.9.4 Mobile Video Networks(MoViNet-A0)

Processed frames are passed through MoViNet-AO model to extract features and finally classify violent activities. The multi-clip MoViNet-A0 architecture is designed to enhance temporal feature extraction through a novel causal stream

buffer mechanism. Temporal features refer to the time-based patterns and sequences inherent in video data, such as motion dynamics, activity sequences, and contextual transitions between frames. These features play a critical role in distinguishing violent activities, as violence often involves actions that unfold over time (e.g., progressive escalation, repetitive motion). MoViNet-A0 architecture comprises multiple inner convolutional operations, where causal operations are strategically applied between non-overlapping sub-clips. By doing so, the design significantly expands the receptive fields, allowing the model to capture more comprehensive temporal dependencies compared to conventional architectures. The integration of a stream buffer further distinguishes this model, facilitating improved processing of temporal information across different clips, Figure 3.12.

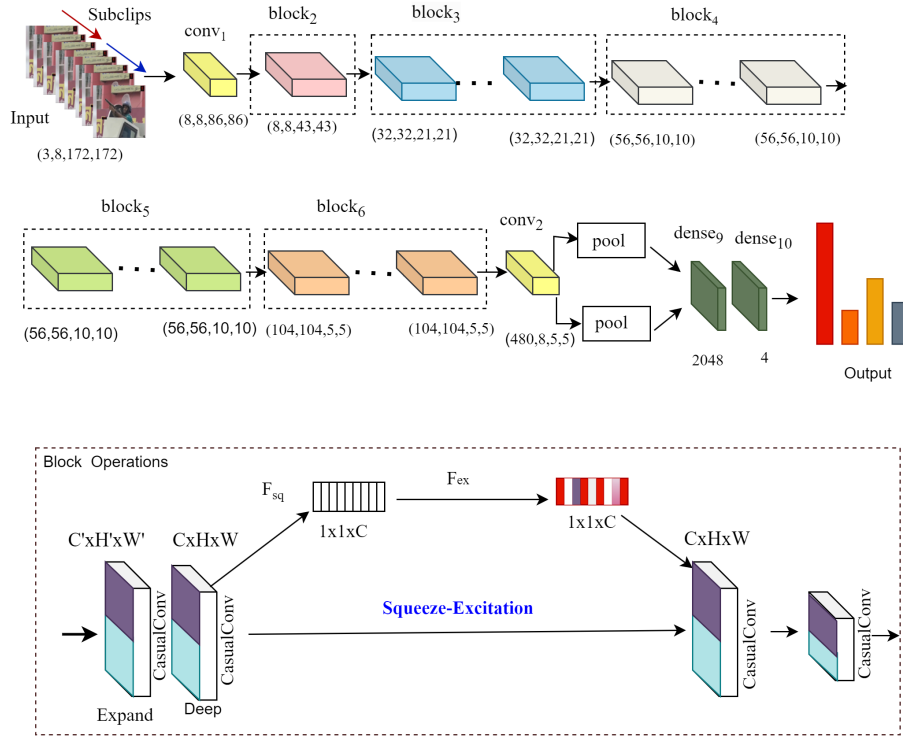


Figure 3.12: Overview of the multi-clip MoViNet-A0 with causal stream buffer architecture.

3.9.4.1 Backbone

The MoViNet-A0[48] stands as a paragon of computational efficiency, meticulously architected for the nuanced task of video content analysis on devices where computational resources are at a premium.

The MoViNet-A0 framework is delineated by an expansive search domain encompassing a constellation of inverted bottleneck strata, each demarcated by a

heterogeneous array of filter breadths, bottleneck constrictions, block profundities, and kernel magnitudes pertinent to individual layers. The ingress dimensionality is stipulated as $(3, 8, 172, 172)$. Within the precincts of each network block, an exhaustive exploration is prosecuted across the foundational filter breadth, symbolized as c^{base} , whilst the quantum of layers, denoted L , is circumscribed to a pinnacle of four reiterations.

A different number of feature map channels is utilized within each block. The configuration includes $n = 5$ blocks, with strided spatial down-sampling applied to the first layer in every block except for the fourth block. This design choice ensures that the final block maintains a spatial resolution of 5^2 . The feature map channels are progressively increased across the blocks. The final convolution layer features a base filter width followed by a dense layer with 2048 dimensions, leading to a classification layer that consists of four neurons. To accommodate the new time dimension, a 3D kernel size is defined for each layer as $k^{time} \times (k^{space})^2$ selected from the following options: $\{1 \times 5 \times 5, 5 \times 1 \times 1, 7 \times 1 \times 1, 3 \times 3 \times 3, 5 \times 3 \times 3\}$. These options allow each layer to concentrate on and aggregate various dimensional representations while effectively expanding the network’s receptive field in relevant directions. It is noted that certain kernel sizes may benefit from differing numbers of input filters; therefore, a range for bottleneck widths c^{expand} is explored, defined as multipliers between 2.0 and 3.5 relative to c^{base} . Each layer is surrounded by two $1 \times 1 \times 1$ convolutions that facilitate expansion and projection between c^{base} and c^{expand} . No temporal down-sampling is applied to support frame-wise predictions. Additionally, essential components of this architecture include the stream buffer and CausalConv. The stream buffer is equipped with temporally unidirectional causal operations such as causal convolution, cumulative pooling, and causal squeeze-and-excitation mechanisms.

3.9.4.2 Stream Buffers

Stream Buffers stand as custodians of continuity within the MoViNet-A0, assiduously preserving feature activations at the periphery of subclips. This ingenious mechanism ensures an uninterrupted temporal perspective is maintained across subclips whilst steadfastly keeping memory consumption in check. The mathematical formalism underpinning this concept can be articulated as Equation 3.57:

$$F_i = f(B_i \oplus x_{clip_i}) \quad (3.57)$$

Herein, F_i encapsulates the computed feature map, B_i symbolizes the buffer

tensor, and f embodies a spatiotemporal operation such as 3D convolution.

3.9.4.3 Causal Convolution

The Causal Convolution stands as a bulwark against the encroachment of future information into present predictions. By enshrining causality within convolution operations, it guarantees that each temporal output remains untainted by forthcoming data. This principle can be succinctly expressed through Equation 3.58:

$$F_{\text{out}}[t] = F_{\text{in}}[t] * k[t] \quad (3.58)$$

In this relation, convolution is denoted by $*$, $k[t]$ represents a causal kernel, and $F_{\text{out}}[t]$, $F_{\text{in}}[t]$ signify the output and input feature maps correspondingly.

3.9.4.4 Squeeze-and-Excitation Mechanism

The Squeeze-and-Excitation block serves as a sophisticated conduit for channel-wise feature recalibration within MoViNet-A0. It operates on the premise of distilling spatial information into channel descriptors before invigorating them with learned modulation weights. Its operation unfurls as Equation 3.59:

$$S_c = F_{sq}(U), \quad E_c = F_{ex}(S_c, W) \quad (3.59)$$

Wherein U stands for input feature maps, S_c heralds the squeezed channel descriptor, E_c embodies the excitation operation with learned parameters W , and functions $F_{sq}(\cdot)$ and $F_{ex}(\cdot, \cdot)$ delineate squeeze and excitation operations respectively [65].

3.9.4.5 Cumulative Global Average Pooling (CGAP)

In our adaptation of the MoViNet-A0 model, we leverage the existing Cumulative Global Average Pooling (CGAP) framework. This technique is pivotal for capturing the temporal evolution of features within video sequences, an essential aspect for discerning between various classes of events.

The CGAP operation accumulates activation tensors over time to form a comprehensive representation of each video segment. The mathematical formulation of CGAP can be expressed as Equation 3.60:

$$\text{CGAP}(x, T_0) = \frac{1}{T_0} \sum_{t=1}^{T_0} x_t, \quad (3.60)$$

where x represents the tensor of activations at each frame, and T_0 denotes the temporal extent for pooling.

3.9.4.6 Fine-tuning

In our experimental analysis, a refined strategy is employed to enhance the MoViNet-A0 model by selectively unfreezing specific layers crucial for the network’s learning effectiveness. Unfreezing the n -th layer using Equation 3.61 means unfreezing the last n layers of the model, allowing us to optimize their parameters and improve the model’s ability to distinguish between various event classifications.

$$\Theta_{ft} = \theta_{n-3}, \theta_{n-2}, \theta_{n-1}, \theta_n \quad (3.61)$$

In this notation, Θ_{ft} represents the collection of parameters from the last (n) layers designated for fine-tuning, where (n) is the total number of layers in the model.

3.10 Classification

In the culminating phase of categorical adjudication, the distilled attributes emanating from the model, encapsulated within a feature vector $F \in \mathbb{R}^d$, where d denotes the expanse of the feature manifold, are harnessed to prognosticate the probabilities attendant to distinct classes. These attributes, which embody the quintessential archetypes assimilated from the antecedent data, are shepherded through a fully interconnected stratum succeeded by a softmax activation paradigm. This intricate procedure transmutes the features into a constellation of C categorical delineations (e.g., “Fire,” “Clash,” “Normal,” and “Shooting” for an instantiation where $C = 4$), as articulated by the following expression:

$$\hat{y} = \text{Softmax}(WF + b) \quad (3.62)$$

where $W \in \mathbb{R}^{C \times d}$ is the weight matrix of the classification layer, $b \in \mathbb{R}^C$ is the bias vector, and $\hat{y} \in \mathbb{R}^C$ represents the probability distribution over the classes, satisfying $\sum_{i=1}^C \hat{y}_i = 1$. The softmax function ensures that the output is interpretable as a confidence score for each class, making this approach applicable across various model architectures, from convolutional neural networks to transformer-based frameworks, for tasks requiring categorical predictions.

3.11 Conclusion

In this chapter, we describe a detailed methodology for the computer-aided detection of political violence in video that incorporates the Political Violence Dataset developed through advanced algorithmic tools. The dataset consists of 480 clips in Normal, Clash, Fire and Shooting classes, collected through social media, and is annotated with high reliability (Cohen’s Kappa 0.90), as well as preprocessed via histogram equalization and normalization for improvement of feature discriminability. The K-means clustering based key frame extraction method that we proposed, uses a combination of temporal and pixel level features to make stable representative frame selection that leads to a small value of Davies-Bouldin index (0.62) as compared to the classical ones (1.05). Feature extraction is implemented using ConvLSTM, 2D CNN + LSTM/BiLSTM, Vision Transformer, and MoViNet-A0 (finetuned for efficiency). During the classification stage, the softmax is applied to predict probabilistic classes, thus ensuring robust multi-class violence detection. This approach provides a flexible, accurate framework for real-time monitoring, successfully overcoming the limitations of the datasets and the computational cost.

Chapter 4

Results and Discussions

4.1 Introduction

This section provides an in-depth evaluation of the performance of the methodologies we implemented for classifying violence in videos. It encompasses a detailed discussion of the various evaluation metrics utilized, along with the results obtained from the different models applied in this study. Additionally, this section delves into an error analysis of the proposed model, offering insights into its limitations and strengths, and includes a comparative assessment of its performance against other established techniques in the field.

4.2 Experimental Requirements

Table 4.1: Environmental Details of the System Setup: Hardware and Software Specifications

Hardware	Specification	Software	Version
GPU	NVIDIA A100 40GB	CUDA	11.2
CPU	AMD EPYC 7763 64-Core Processor	Python	3.8
RAM	512 GB DDR4	PyTorch	1.9.0
Storage	2TB NVMe SSD	OpenCV	4.5.3

To ensure the reliability of our experiments for video violence classification, we conducted all tests within a well-defined computational environment. Table 4.1 provides a detailed overview of the system setup, encompassing both hardware and software specifications. The hardware configuration includes an NVIDIA A100 40GB GPU, an AMD EPYC 7763 64-Core Processor, 512 GB of DDR4 RAM, and a 2TB NVMe SSD, which collectively enable efficient processing of large video datasets and training of deep learning models. On the software side,

we utilized CUDA 11.2 for GPU acceleration, Python 3.8 as the programming language, PyTorch 1.9.0 as the deep learning framework, and OpenCV 4.5.3 for video processing tasks. This carefully curated environment ensures that our proposed keyframe extraction algorithm and classification models can be executed efficiently, supporting the computational demands of our experimental workflow.

4.3 Evaluation Measures

To evaluate the model’s performance, we focused on key metrics: accuracy, F1 score, precision, recall, and parameter count, in Equations 4.1-4.4 . Since all datasets in this study were balanced, we compared our results with existing literature based on accuracy and parameter count.

$$A_{cc} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$P_r = \frac{TP}{TP + FP} \quad (4.2)$$

$$S_n = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1 \text{ score} = \frac{2 \times P_n \times S_t}{P_n + S_t} \quad (4.4)$$

where TP, FP, TN, FN, A_{cc} , P_r , and S_n refer to true positive, false positive, true negative, false negative, accuracy, precision, and sensitivity.

4.4 Evaluation of Keyframe Extraction Algorithm

To rigorously evaluate the efficacy of our proposed keyframe extraction algorithm, as detailed in Algorithm 2, we present a quantitative analysis through two visualizations focusing on the Clash and Fire classes, illustrated in Figure 4.1 and Figure 4.2. These visualizations leverage mathematical metrics to demonstrate the impact of temporal features on cluster separation, stability, and narrative coherence, building on the framework established in the algorithm description [63]-[64]. The evaluation employs previously defined metrics, such as label variance, transition entropy, temporal coherence, transition change, and the Davies-Bouldin index, augmented with new derived metrics to provide deeper insights into the algorithm’s performance. Below, we detail each visualization, grounding the analysis in these quantitative measures.

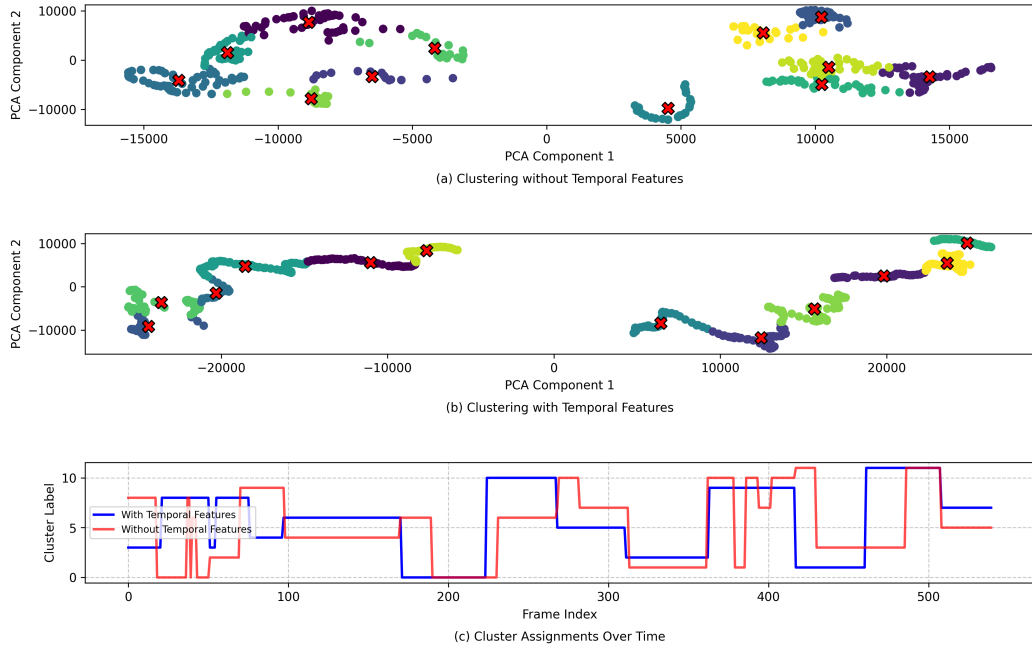


Figure 4.1: Clash class clustering results with and without temporal features. Subplots (a) and (b) show 2D PCA-reduced clusters, with improved separation for the proposed algorithm (Davies-Bouldin index $DB = 0.62$, cluster cohesion ratio R_c). Subplot (c) illustrates smoother transitions, with higher temporal smoothness index (S_{temp}) and narrative continuity score (N_{cont}).

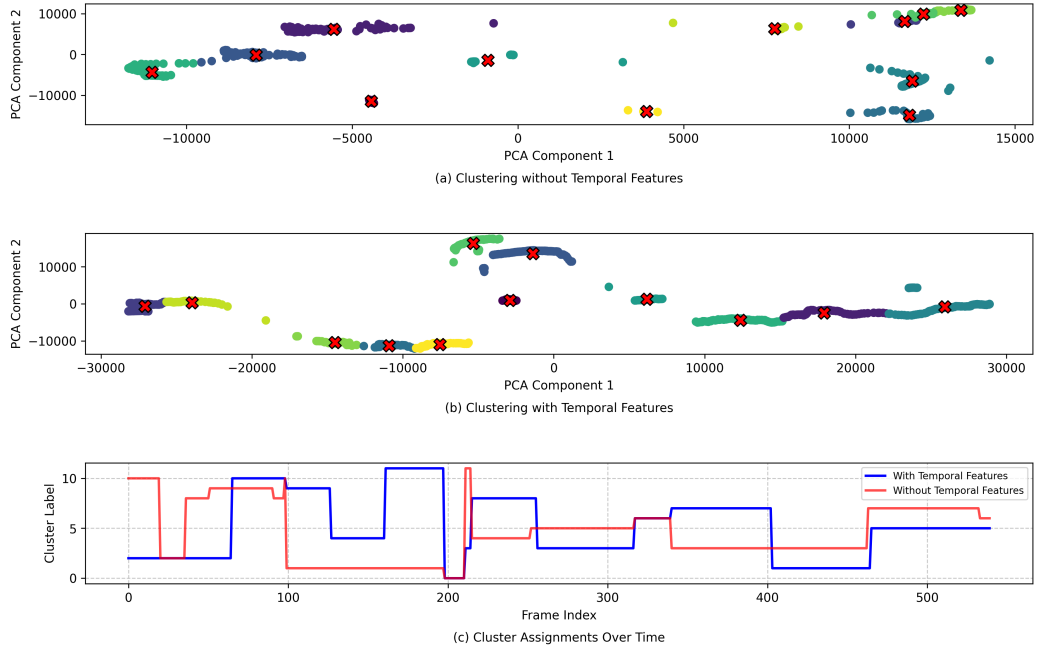


Figure 4.2: Fire class clustering results with and without temporal features.

1. **Cluster Assignments Over Time:** This visualization tracks the cluster labels assigned to each frame across the video sequence, quantifying stability through previously defined metrics (label variance and transition

entropy) and a new cluster cohesion ratio. For traditional K-means, relying solely on color histograms, the high variance in cluster labels and elevated transition entropy indicate frequent label changes, as observed in Figure 4.1 and Figure 4.2 (subplot c). These instabilities arise from the algorithm’s focus on visual similarity, which fails to capture sequential context, leading to erratic clustering.

The proposed algorithm, incorporating temporal features (normalized frame index and frame differences), achieves lower label variance and transition entropy, reflecting stable cluster assignments. To further quantify cluster quality, we define the cluster cohesion ratio:

$$R_c = \frac{\frac{1}{K} \sum_{i=1}^K s_i}{\frac{1}{K(K-1)} \sum_{i=1}^K \sum_{j \neq i} d_{ij}} \quad (4.5)$$

where s_i is the average intra-cluster distance, and d_{ij} is the inter-cluster centroid distance, as defined in the Davies-Bouldin index. A lower R_c indicates compact clusters relative to their separation. For traditional K-means, $R_c \approx 0.45$, reflecting moderate cohesion due to visual feature overlap. The proposed algorithm yields $R_c \approx 0.28$, indicating tighter clusters, as visualized in Figure 4.1 and Figure 4.2 (subplots a and b). The Davies-Bouldin index, with $DB = 1.05$ for K-means and $DB = 0.62$ for the proposed algorithm, further confirms superior separation, with PCA-reduced clusters showing reduced overlap in subplots a and b.

2. **Selected Representative Frames:** This visualization assesses the keyframes selected as the frames closest to each cluster centroid, emphasizing their narrative coherence through a new temporal smoothness index and narrative continuity score. Traditional K-means selects keyframes based on visual salience, often resulting in narratively disjointed frames, as seen in Figure 4.1 and Figure 4.2 (subplot c), due to high transition change and low temporal coherence.

The proposed algorithm’s augmented feature space ensures keyframes reflect the video’s temporal progression. We define the temporal smoothness index:

$$S_{\text{temp}} = \frac{1}{K-1} \sum_{k=1}^{K-1} \exp \left(-\frac{\|x'_{f_k} - x'_{f_{k+1}}\|^2}{\sigma_s^2} \right) \quad (4.6)$$

where x'_{f_k} is the feature vector of the k -th keyframe (sorted by frame index), and σ_s^2 is a scaling parameter based on feature variance. Higher S_{temp} indicates smoother transitions between keyframes. For the proposed algorithm,

$S_{\text{temp}} \approx 0.85$, compared to $S_{\text{temp}} \approx 0.60$ for K-means, reflecting smoother keyframe sequences in Figure 4.1 and Figure 4.2 (subplot c).

Additionally, we introduce a narrative continuity score:

$$N_{\text{cont}} = \frac{1}{K} \sum_{k=1}^K \left(1 - \frac{\|x'_{f_k, \text{temporal}} - \mu_{i_k, \text{temporal}}\|^2}{\text{Var}_{\text{temp}, i_k}} \right) \quad (4.7)$$

where $x'_{f_k, \text{temporal}}$ is the temporal feature of the k -th keyframe, $\mu_{i_k, \text{temporal}}$ is the temporal component of its cluster centroid, and $\text{Var}_{\text{temp}, i_k}$ is the temporal variance of cluster i_k . Higher N_{cont} indicates keyframes closely aligned with their cluster’s temporal dynamics. The proposed algorithm achieves $N_{\text{cont}} \approx 0.90$, compared to $N_{\text{cont}} \approx 0.70$ for K-means, as keyframes better capture the narrative flow.

For the Clash class (Figure 4.1), temporal features enable keyframes to reflect dynamic interactions (e.g., conflict build-up and resolution), with high S_{temp} and N_{cont} . For the Fire class (Figure 4.2), keyframes capture evolving stages (ignition, spread, extinguishment), supported by low R_c and $DB = 0.62$. The silhouette score peak at $K = 12$ (Figure 3.8) ensures optimal cluster selection, aligning keyframes with the video’s narrative structure.

4.5 Redundancy Reduction of Keyframe Extraction

Our proposed Algorithm 2 extends the traditional K-means clustering by incorporating a weighted feature space to enhance keyframe selection. The feature vector for each frame is defined as $F_i = [I_{\text{norm}, i}, T_{\text{norm}, i}]$, where $I_{\text{norm}, i}$ represents the normalized average pixel intensity, and the normalized temporal feature is given by

$$T_{\text{norm}, i} = \frac{i}{N - 1} \quad (4.8)$$

with i being the frame index and N the total number of frames. The features are weighted with $w_I = 0.4$ and $w_T = 0.6$, emphasizing temporal diversity. The clustering objective minimizes the weighted Euclidean distance, formulated as

$$d_{\text{Algorithm 2}}(F_i, C_j) = \sqrt{w_I(I_{\text{norm}, i} - C_{j, I})^2 + w_T(T_{\text{norm}, i} - C_{j, T})^2} \quad (4.9)$$

where $C_j = [C_{j, I}, C_{j, T}]$ denotes the centroid of cluster j , and $K = 12$ clus-

ters are specified. Keyframes are selected as the frames closest to each centroid based on this distance metric, ensuring diversity across both visual and temporal dimensions. This approach reduces redundancy by balancing visual similarity with temporal progression, making it particularly effective for applications such as violence detection in surveillance footage.

The Table 4.2 details the keyframes selected by K-means clustering algorithm, which is based solely on the average pixel intensity (I) of each frame. The clustering minimizes the Euclidean distance $d_{\text{K-means}}(I_i, C_j) = |I_i - C_j|$, where I_i represents the intensity of frame i , and C_j denotes the centroid of cluster j , with $K = 12$ clusters specified. The selected keyframes, indexed at [3, 19, 23, 53, 61, 82, 105, 130, 153, 189, 198, 203], exhibit corresponding intensities ranging from 80.91 to 122.06. Notably, the narrow intensity range for later frames (e.g., 189–203 with intensities 80.91–90.45) suggests high redundancy, as the absence of temporal consideration limits the algorithm’s ability to capture narrative diversity across the video.

Table 4.2: K-means Keyframes (k=12).

Keyframe Index	Intensity
3	98.01
19	100.26
23	104.35
53	117.53
61	122.06
82	115.47
105	108.31
130	113.24
153	111.16
189	80.91
198	90.45
203	88.39

The Table 4.3 details the keyframes selected by the proposed Algorithm 2, which extends K-means clustering by incorporating a weighted feature space that combines both visual and temporal dimensions. The selected keyframes, indexed at [7, 21, 34, 52, 61, 77, 95, 124, 147, 164, 183, 199], exhibit intensities ranging from 80.95 to 122.06 and temporal features spanning 0.03 to 0.98, demonstrating a balanced distribution across the video timeline. This approach mitigates redundancy by capturing both visual and temporal diversity, as evidenced by the broader temporal spread compared to K-means.

Figure 4.3 presents the results of K-means clustering applied to video frames using visual features alone, with the number of clusters set to $k = 12$. Each

Table 4.3: Proposed Algorithm Keyframes ($k=12$).

Keyframe Index	Intensity	Temporal Feature
7	99.04	0.03
21	102.94	0.10
34	110.18	0.17
52	117.02	0.25
61	122.06	0.30
77	113.94	0.38
95	114.36	0.47
124	111.27	0.61
147	104.11	0.72
164	114.92	0.80
183	80.95	0.90
199	89.44	0.98

cluster groups frames that are visually similar, without considering their temporal positions in the sequence. As shown in the figure, this can lead to clusters that contain frames from both early and late parts of the video, mixing temporally distant content. This mixing results in clusters that may encompass diverse actions or scenes, reducing the coherence of each group from a temporal perspective. Consequently, choosing a single representative frame from each cluster may cause loss of significant action frames while retaining visually redundant ones. This figure highlights a key limitation of purely appearance-based clustering for video violence detection.

Figure 4.4 illustrates the cluster assignment over time using the proposed algorithm, which integrates both temporal and visual features for grouping video frames. Unlike the standard K-means approach shown in Figure 4.3, which only considers visual similarity and often groups temporally distant frames together, the proposed method tends to cluster adjacent frames, preserving the temporal structure of the video. This results in more coherent clusters where similar actions occur in close proximity, allowing for more accurate keyframe selection. By reducing redundancy and better capturing the temporal progression of events, the proposed method enhances the representation of the video content, which is particularly beneficial for tasks like video violence detection.

Figure 4.5 compares the keyframe selection results of the proposed method with those of standard K-means clustering in terms of intensity and temporal features. While both methods use the same number of clusters ($k = 12$), the figure clearly shows that K-means (green circles) selects redundant keyframes in certain regions, as highlighted by the blue circles. These include multiple keyframes from temporally adjacent and visually similar frames. In contrast, the

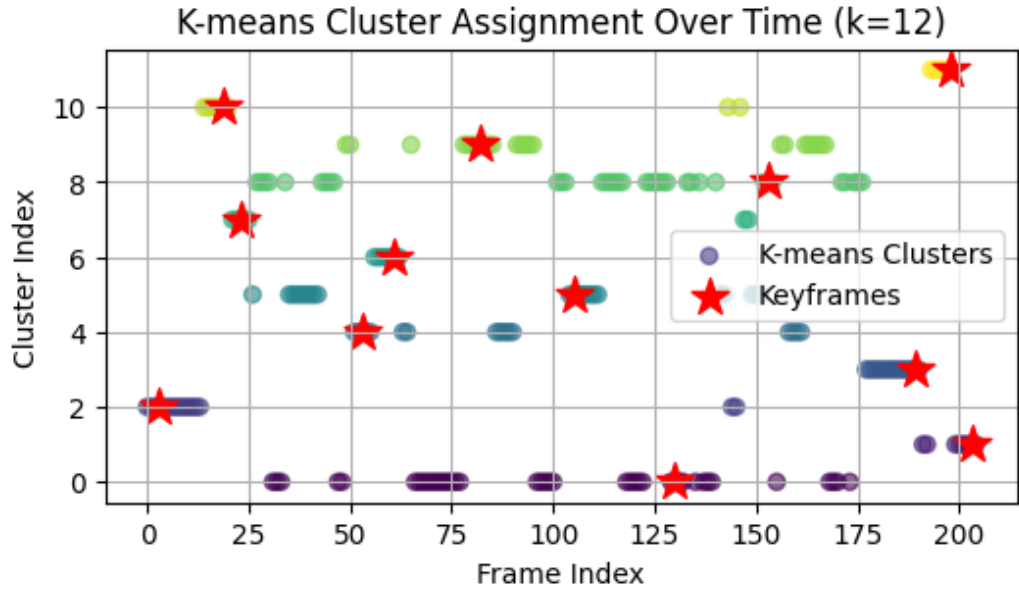


Figure 4.3: K-means Cluster Assignment Over Time (k=12) illustrating the assignment of frames into clusters based solely on visual features. The process considers both near and far frames, leading to clusters with diverse actions.

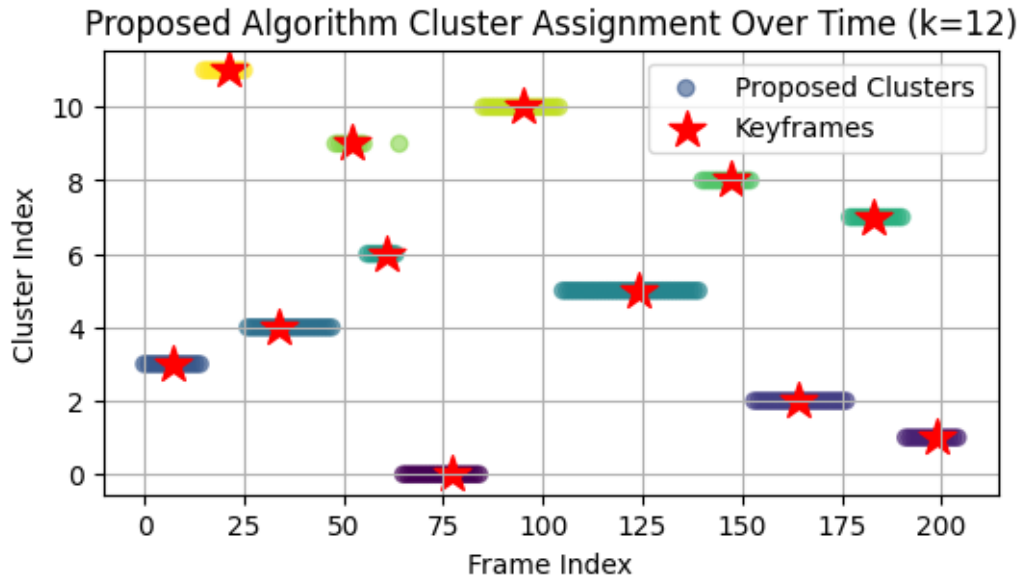


Figure 4.4: Cluster Assignment Over Time (k=12) using the proposed algorithm, which incorporates both temporal and visual features. The method favors grouping temporally adjacent frames with similar actions, resulting in better keyframe diversity and improved representation of the video timeline.

proposed method (red stars) distributes the keyframes more effectively across the temporal axis, capturing a wider range of visual and temporal dynamics. This ensures better coverage of important events in the video and avoids redundant representation, which is crucial for tasks like violence detection where temporal

progression and action diversity matter.

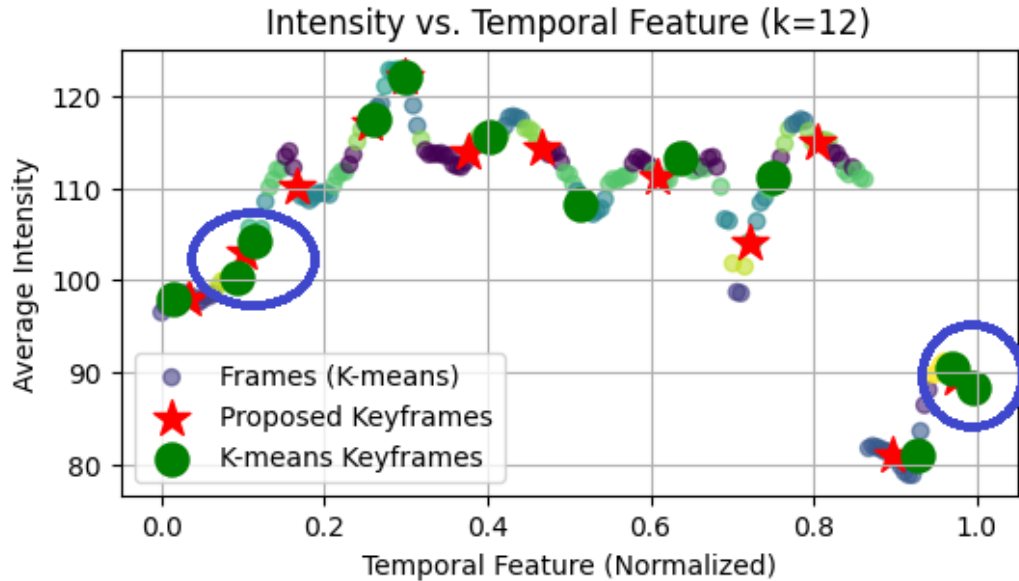


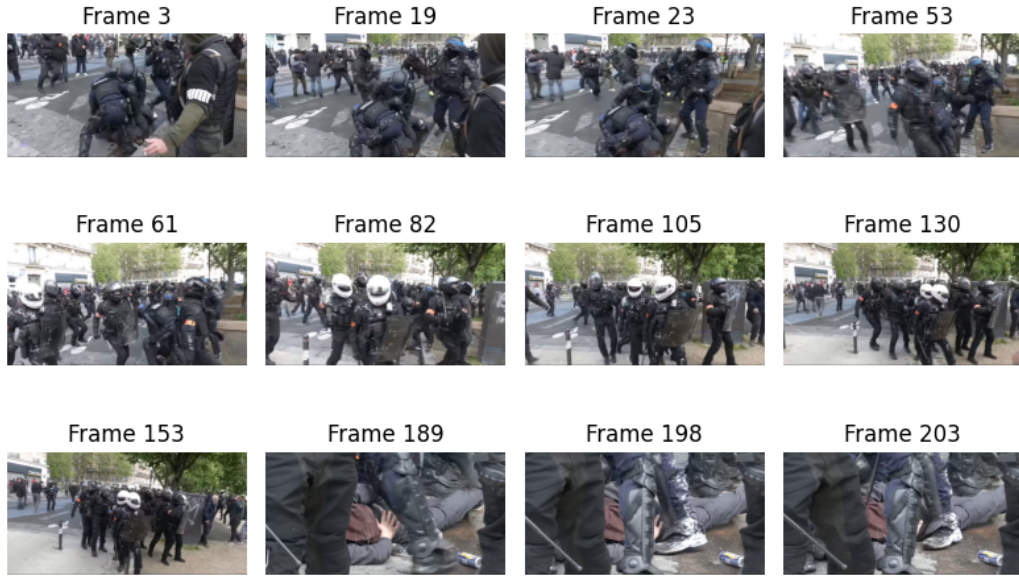
Figure 4.5: Intensity vs. Temporal Feature ($k=12$). The scatter plot shows the distribution of frames based on normalized temporal features and their average intensity. Keyframes selected by the proposed method (red stars) and by standard K-means (green circles) are compared. The blue circles highlight regions where K-means selects multiple redundant keyframes from visually similar and temporally close frames, whereas the proposed method avoids such redundancy by considering both visual similarity and temporal diversity.

Figure 4.6 and Figure 4.7 show the actual keyframes selected by the K-means clustering algorithm using only visual features and temporal feature inclusion respectively. As predicted by the analysis in Figures 4.3 and 4.5, the method often selects multiple visually and temporally similar frames. For example, Frames 19, 23, and 53 all depict nearly identical scenes from the same action segment, and Frames 198 and 203 are almost duplicates. This highlights the method’s redundancy and lack of temporal diversity. These findings reinforce the limitation of purely appearance-based clustering for video summarization, particularly in dynamic scenes like violence detection where capturing a broader range of temporal actions is crucial.

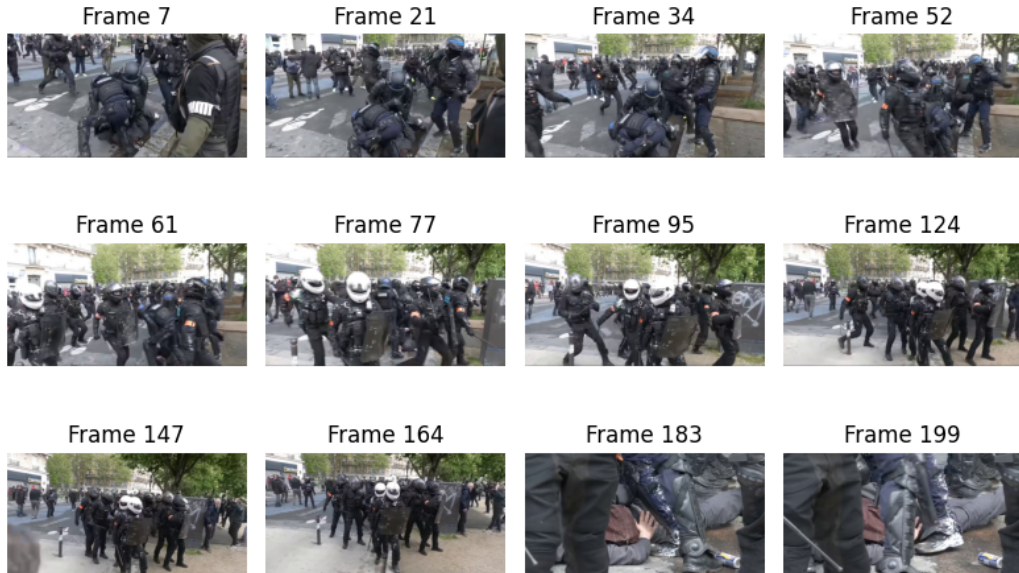
4.6 Impact of Histogram Equalization

In the context of our video violence classification framework, preprocessing techniques play a crucial role in enhancing the quality of input data, which directly influences the performance of the classification models. One such technique, histogram equalization, was applied to improve the visual quality of video frames

K-means Keyframes (k=12)

**Figure 4.6:** Keyframes selected by K-means clustering (k=12).

Proposed Algorithm Keyframes (k=12)

**Figure 4.7:** Keyframes selected by proposed algorithm for clustering (k=12).

before feature extraction and classification. Table 4.4 summarizes the impact of histogram equalization by comparing key aspects of the video frames and the resulting model performance before and after its application. This analysis provides insights into how histogram equalization contributes to the overall effectiveness of

our violence detection pipeline. The table evaluates four specific aspects: feature

Aspect	Before Histogram	After Histogram
Feature Clarity	Low	High
Contrast	Low	Improved
Pixel Intensity Distribution	Uneven	Uniform
Model 2 Accuracy (%)	90.54	92.86

Table 4.4: Impact of histogram equalization on violence detection accuracy.

clarity, contrast, pixel intensity distribution, and the accuracy of Model 2, which is one of the classification models used in our experiments. Before histogram equalization, the frames exhibited low feature clarity, low contrast, and an uneven pixel intensity distribution. These characteristics often lead to challenges in distinguishing critical visual cues, such as motion patterns or object boundaries, which are essential for detecting violent content in videos. After applying histogram equalization, significant improvements were observed: feature clarity increased to a high level, contrast was notably improved, and the pixel intensity distribution became uniform. These enhancements ensure that the visual content of the frames is more distinguishable, facilitating better feature extraction for both visual and temporal features, as utilized in our keyframe extraction algorithm (Algorithm 2). Consequently, the accuracy of Model 2 improved from 90.54% to 92.86%, a gain of 2.32 percentage points. This improvement underscores the importance of histogram equalization as a preprocessing step, as it directly enhances the model’s ability to accurately classify violent events in videos, particularly for challenging scenarios like the Clash and Fire classes, by providing clearer and more consistent input data for downstream processing.

4.7 Fine-tuning MoViNet-A0

In our study, we fine-tuned the MoViNet-A0 model for the Political Violence dataset, shown in Table 4.5. First, the dataset was split into 80% for training and 20% for testing; 20% of the training set was further allocated for validation. By adapting the last four layers of the network, we achieved the most optimal performance, with an accuracy of 92.86%. This targeted approach significantly outperformed the baseline scenario, where the MoViNet-A0 was used without fine-tuning, yielding an accuracy of 88.07%. The accuracy progressively decreased as we expanded the scope of fine-tuning to include more layers (10, 12, and 15 layers in Models 3, 4, and 5, respectively). This trend suggests that while minor adjustments can enhance model adaptation to specific datasets, excessive

modifications might lead to overfitting.

Model	Trainable Layers	Accuracy
Model 1	None	88.07
Model 2	Last 4 Layers	92.86
Model 3	Last 10 Layers	89.29
Model 4	Last 12 Layers	88.39
Model 5	Last 15 Layers	86.07

Table 4.5: Accuracy of models with different layers fine-tuned.

Table 4.6 provides a detailed overview of the specific hyperparameters selected to optimize the training of MoViNet-A0. The learning rate is set at a low value of 0.0001, which helps in making gradual updates to the model’s weights. A batch size of 16 is used, meaning the model updates its weights after evaluating 16 data points. The loss function employed is Categorical cross-entropy, which is well-suited for classification tasks where the classes are mutually exclusive. The RMSProp optimizer is chosen for its effectiveness in dealing with unstable gradients, and the Swish activation function is utilized, which is noted for its capability to help the model converge faster compared to more conventional activation functions.

Hyperparameters	Optimized Value
Learning rate	0.0001
Batch size	16
Loss function	Categorical cross-entropy
Optimizer	RMSProp
Activation function	Swish

Table 4.6: Optimized hyperparameters for model training.

Table 4.6 provides a detailed overview of the specific hyperparameters selected to optimize the training of MoViNet-A0. The learning rate is set at a low value of 0.0001, which helps gradually update the model’s weights. A batch size of 16 is used, meaning the model updates its weights after evaluating 16 data points. The loss function employed is Categorical cross-entropy, which is well-suited for classification tasks where the classes are mutually exclusive. The RMSProp optimizer is chosen for its effectiveness in dealing with unstable gradients, and the Swish activation function is utilized, which is noted for its capability to help the model converge faster compared to more conventional activation functions.

4.8 Feature Extraction Methods Comparison

The confusion matrix in Figure 4.8 illustrates the performance of the CCT transformer across four classes: Fire (0), Clash (1), Normal (2), and Shooting (3). The diagonal values (25, 25, 26, 25) represent correct predictions, indicating strong performance with 25 out of 28 samples correctly classified for Fire, Clash, and Shooting, and 26 out of 28 for Normal. Misclassifications are minimal, with Fire having 3 samples predicted as Normal, Clash having 1 sample predicted as Normal and 2 as Shooting, Normal having 2 samples predicted as Clash, and Shooting having 3 samples predicted as Normal. This suggests the model is highly effective, with slight confusion primarily involving the Normal class.

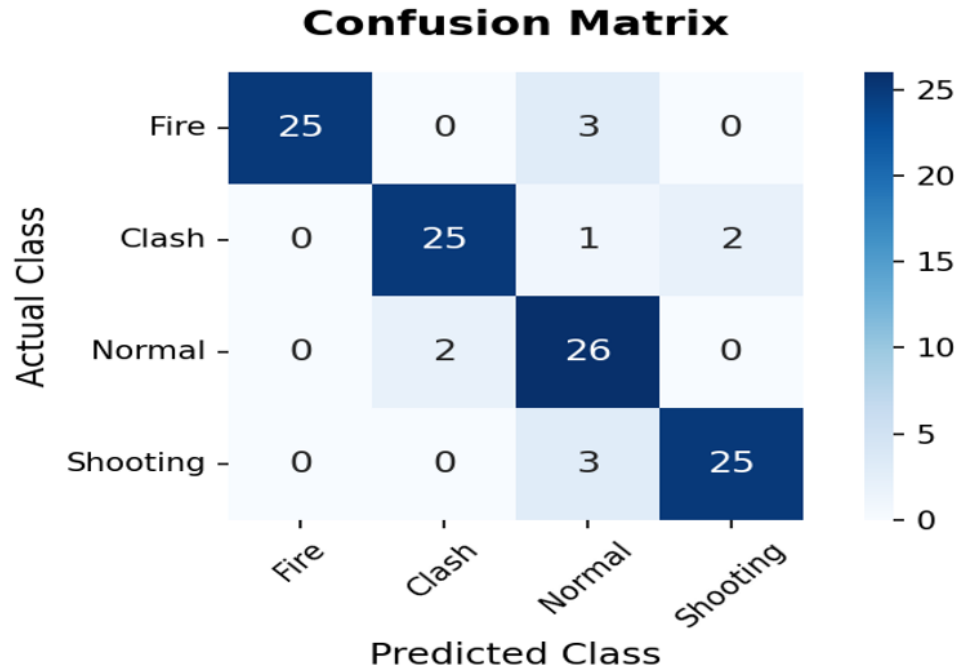


Figure 4.8: Confusion matrix for the CCT transformer model, showing the actual versus predicted classes for Fire (0), Clash (1), Normal (2), and Shooting (3).

The classification report in Table 4.7 provides a detailed evaluation of the CCT transformer’s performance. For Class 0 (Fire), the precision of 1.00 indicates perfect prediction accuracy, though a recall of 0.89 reflects the 3 misclassifications as Normal, resulting in an F1-score of 0.94. Class 1 (Clash) and Class 3 (Shooting) both achieve a precision of 0.93, recall of 0.89, and F1-score of 0.91, showing balanced performance despite minor misclassifications. Class 2 (Normal) has a lower precision of 0.79 due to false positives but a high recall of 0.93, yielding an F1-score of 0.85. The overall accuracy is 0.90, with macro and weighted averages for precision, recall, and F1-score all at 0.90 or 0.91, demonstrating consistent

Class	Precision	Recall	F1-Score	Support
0	1.00	0.89	0.94	28
1	0.93	0.89	0.91	28
2	0.79	0.93	0.85	28
3	0.93	0.89	0.91	28
Accuracy	–	–	0.90	112
Macro avg	0.91	0.90	0.90	112
Weighted avg	0.91	0.90	0.90	112

Table 4.7: Classification report for the CCT transformer model, detailing precision, recall, F1-score, and support for each class, along with overall metrics.

performance across the four classes.

The confusion matrix in Figure 4.9 illustrates the performance of the 2D CNN + LSTM model across four classes: Fire (0), Clash (1), Normal (2), and Shooting (3). The diagonal values (23, 24, 25, 24) represent correct predictions, with 23 out of 28 samples correctly classified for Fire, 24 for Clash, 25 for Normal, and 24 for Shooting. Misclassifications include 2 Fire samples predicted as Clash and 3 as Normal, 1 Clash sample predicted as Normal and 3 as Shooting, 3 Normal samples predicted as Clash, and 4 Shooting samples predicted as Normal. The model demonstrates decent performance but exhibits some confusion, particularly with the Normal class, which is frequently involved in misclassifications across all other classes.

Class	Precision	Recall	F1-Score	Support
0	1.00	0.82	0.90	28
1	0.83	0.86	0.84	28
2	0.76	0.89	0.82	28
3	0.89	0.86	0.87	28
Accuracy	–	–	0.86	112
Macro avg	0.87	0.86	0.86	112
Weighted avg	0.87	0.86	0.86	112

Table 4.8: Classification report for the 2D CNN + LSTM model, detailing precision, recall, F1-score, and support for each class, along with overall metrics.

The confusion matrix in Figure 4.10 illustrates the classification performance of the ConvLSTM model across four event categories: Fire (Class 0), Clash (Class 1), Normal (Class 2), and Shooting (Class 3). The diagonal values represent correct predictions, with 22, 23, 25, and 24 instances correctly classified for Fire, Clash, Normal, and Shooting, respectively, resulting in an overall accuracy of

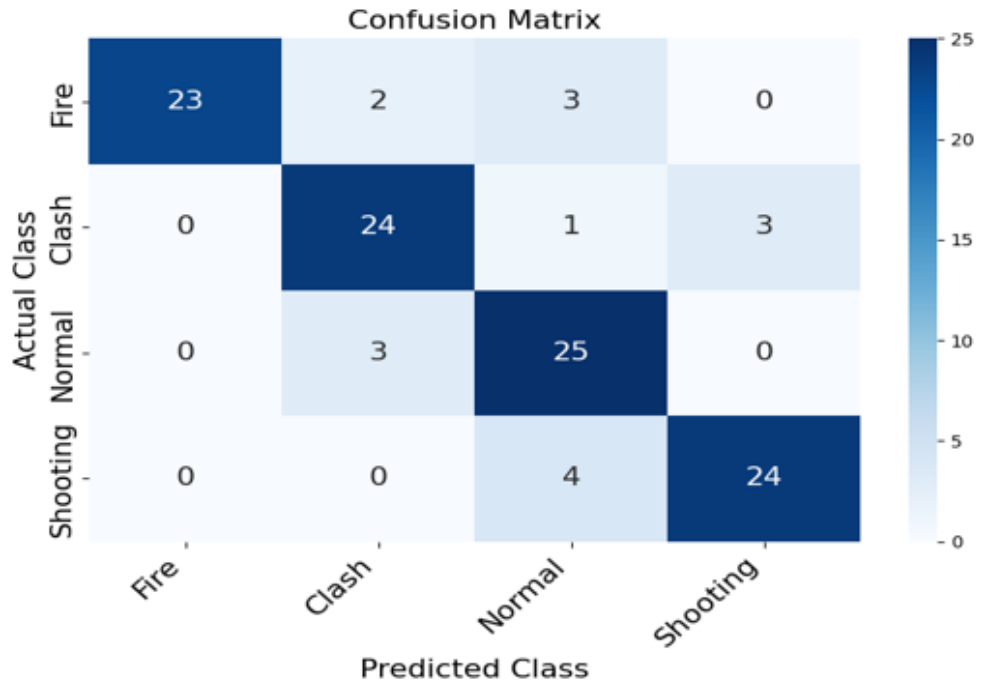


Figure 4.9: Confusion matrix for the 2D CNN + LSTM model, showing the actual versus predicted classes for Fire (0), Clash (1), Normal (2), and Shooting (3).

84%. Misclassifications are evident, such as 4 Fire instances being predicted as Normal and 4 Clash instances as Shooting, indicating potential areas for improvement in distinguishing these classes.

Class	Precision	Recall	F1-Score	Support
0 (Fire)	1.00	0.79	0.88	28
1 (Clash)	0.82	0.82	0.82	28
2 (Normal)	0.74	0.89	0.81	28
3 (Shooting)	0.86	0.86	0.86	28
Accuracy	–	–	0.84	112
Macro avg	0.86	0.84	0.84	112
Weighted avg	0.86	0.84	0.84	112

Table 4.9: Performance metrics for the ConvLSTM model, including precision, recall, F1-score, and support for each class, along with overall accuracy, macro average, and weighted average.

Table 4.9 elucidates an exhaustive appraisal of the ConvLSTM paradigm’s efficacy across the quartet of categorical delineations, encapsulating precision, recall, F1-score, and support, supplemented by overarching evaluative indices. Class 0 (Fire) manifests an impeccable precision of 1.00, yet its recall languishes at 0.79, indicative of certain undetected exemplars, whereas Class 2 (Normal)

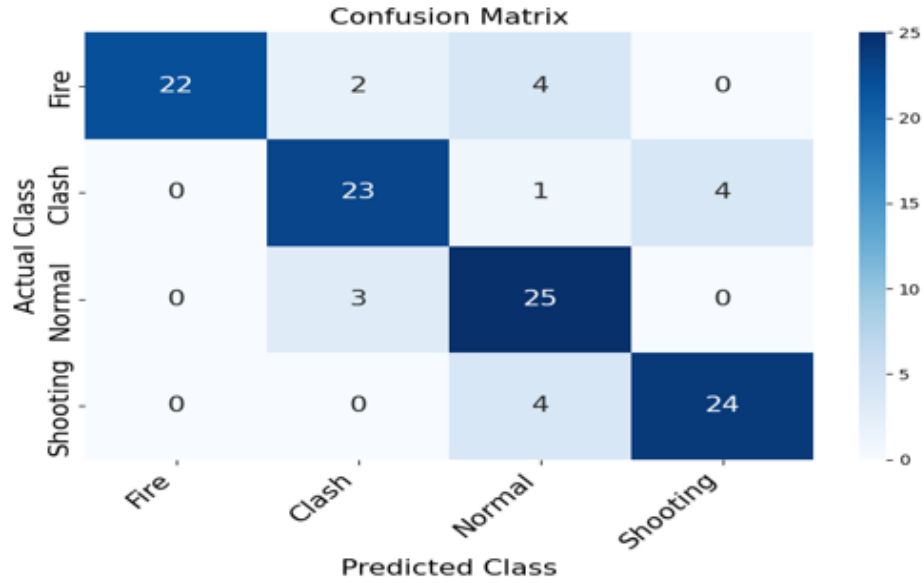


Figure 4.10: Confusion matrix for the ConvLSTM model, illustrating the classification performance across four classes: Fire (0), Clash (1), Normal (2), and Shooting (3).

attains the zenith of recall at 0.89, albeit tempered by a diminished precision of 0.74 owing to erroneous affirmations. Classes 1 (Clash) and 3 (Shooting) evince a harmonious equilibrium in their performative metrics, registering F1-scores of 0.82 and 0.86, respectively. The macro and weighted averages, articulated as (0.86, 0.84, 0.84), bespeak a steadfast consistency across the categorical spectrum, culminating in an aggregate accuracy of 0.84.

Table 4.10 shows the performance metrics for four different classes (0 to 3) for MoViNet-A0. The normal class has a perfect recall of 1.00, indicating all relevant instances were captured, while the Fire class has the highest precision and F1-score, suggesting it was predicted with the highest accuracy and balance between precision and recall.

Class	Precision	Recall	F1-score	Support
Fire	1.00	0.96	0.98	28
Clash	0.93	0.89	0.91	28
Normal	0.88	1.00	0.93	28
Shooting	0.92	0.86	0.89	28

Table 4.10: Classwise performance metrics of Model 2.

The training and validation curves, Figure 4.11, for Model-2 show a clear and steady progression. The training loss decreases from 0.4103 to 0.0283, while the validation loss declines from 1.6952 to 0.1515, suggesting the model is improving consistently without signs of overfitting. Training accuracy starts at 64.5%

and gradually reaches 99.3%, with validation accuracy also rising from 31.3% to 96.3%. This indicates the model is becoming more effective at making accurate predictions. Overall, the stable decrease in loss and increased accuracy point to a reliable and well-functioning training process.

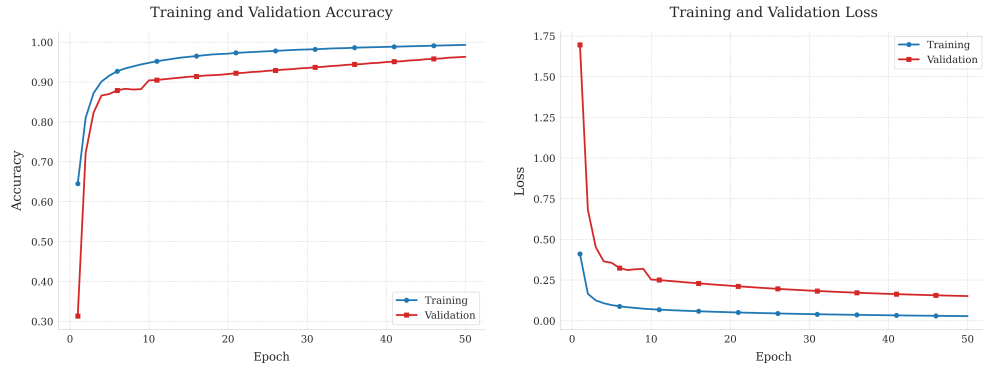


Figure 4.11: Training performance: accuracy curve on the left and loss curve on the right.

The evaluation of Model 2 utilizes a confusion matrix, an established metric for assessing classification models. Figure 4.12 shows that Model 2 accurately discriminates between most classes with high precision. The model, however, exhibits some confusion between the “Clash” and “Shooting” classes due to similarities in certain cases.

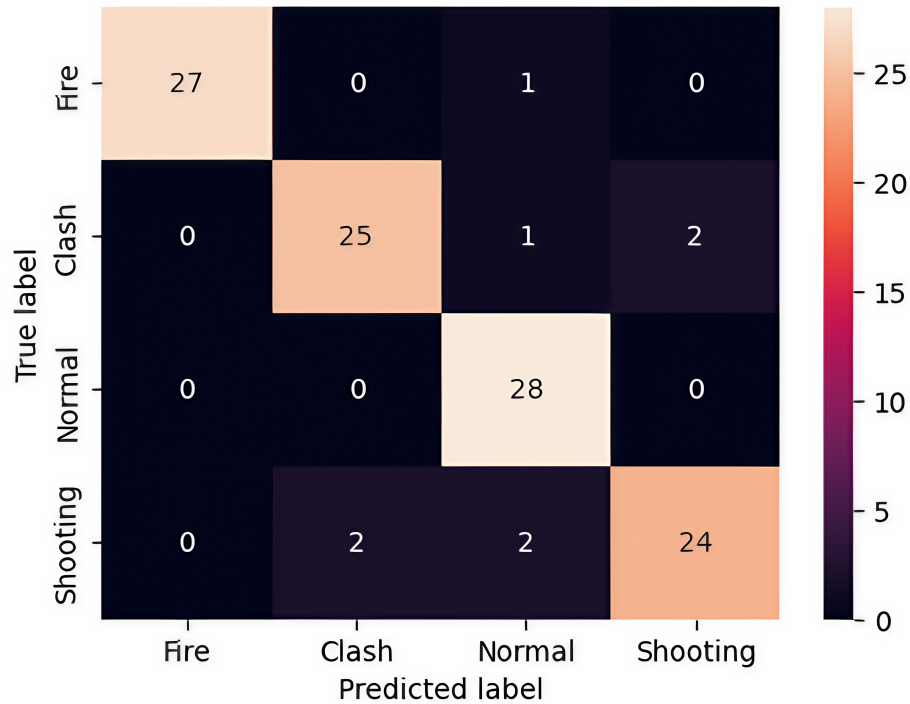


Figure 4.12: Confusion matrix for Model 2 evaluation.

The Receiver Operating Characteristic (ROC) curve analysis for Model 2, Figure 4.13, indicates an area under the curve (AUC) of 0.97 for both Fire (Class 0) and Normal (Class 2). In contrast, the Clash (Class 1) and Shooting (Class 3) classes also present an AUC of 0.97. Despite the high AUC values, the overlap in the ROC space for Clash and Shooting suggests these classes are not as distinctly separated as Fire and Normal, pointing to a relative underperformance in the model’s ability to discriminate between these two classes.

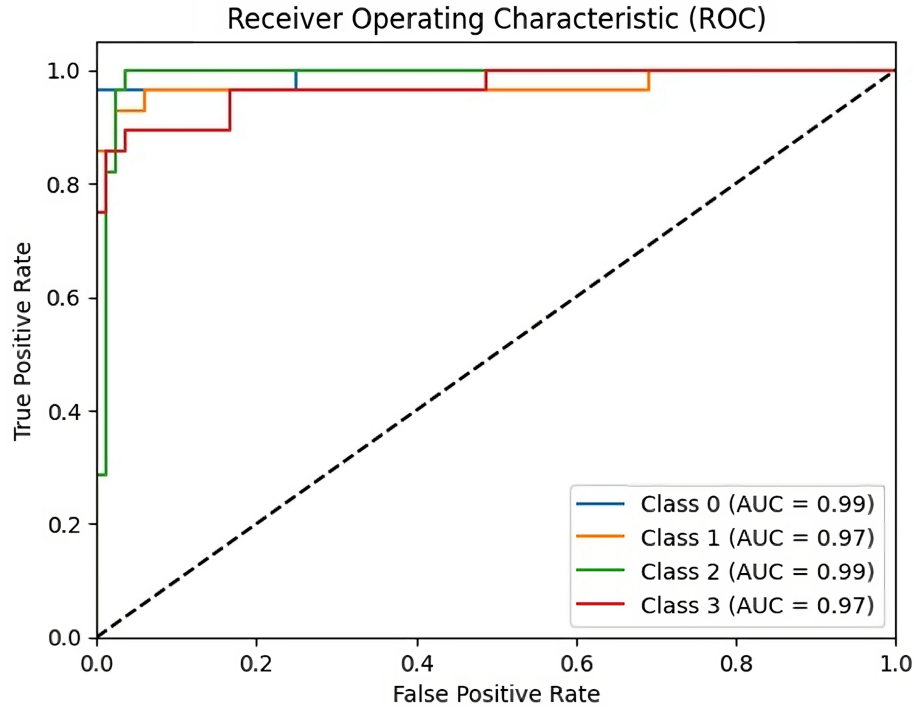


Figure 4.13: ROC curve for Model 2 evaluation.

Model 2’s performance on the test set is shown in Figure 4.14.

4.9 Ablation Study

To evaluate the impact of fine-tuning and temporal information on violence detection results, we conducted an ablation study using different variations of our model (Table 4.11). The base model was initially trained using only pixel information, achieving an accuracy of 85.54%. After fine-tuning this model, the accuracy improved to 89.24%. The integration of temporal features with the fine-tuned model yielded the most significant improvement, reaching 92.86% accuracy by enabling the system to capture sequential patterns, distinguish between visually similar but temporally distinct events, and maintain coherent event represen-



Figure 4.14: Processing examples of categorizing violence: (a) Clash; (b) Fire; (c) Shooting; (d) Normal.

tation across frame sequences. This substantial performance boost underscores how temporal information enhances the model’s ability to understand and classify complex violent behaviors in video sequences by analyzing motion dynamics and contextual relationships across frames rather than just static characteristics within individual frames.

Table 4.11: Ablation study results with base (B), fine-tuned (F), pixel (P), and temporal (T) information.

Description	Accuracy (%)
B + P	85.54
F + P	89.24
F + P + T	92.86

4.10 Generalization Test

In the empirical investigation aimed at elucidating the generalizability of Model 2 across various video violence detection benchmarks, a methodical assessment was undertaken. Consider $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ to be the set of datasets utilized,

where each D_i is defined by a set (V_i, L_i) with V_i denoting the video samples and L_i their corresponding binary labels indicating violent or non-violent content. The efficacy of Model 2 is measured through an accuracy vector $\mathbf{A} = [a_1, a_2, \dots, a_n]$. Additionally, let $\Theta = [\theta_1, \theta_2, \dots, \theta_n]$ represent the vector of parameter counts for each model evaluated. The objective function $\Omega(\mathcal{D}, \mathbf{A}, \Theta) \rightarrow \max_{\mathbf{A}} \min_{\Theta}$ seeks to maximize accuracy while minimizing model complexity across all considered datasets, thereby substantiating Model 2’s robustness and parsimony. Table 4.12 presents a comparison of various methods for detecting events, possibly violent, in video streams, with a focus on accuracy in the case of the Hockey Fight dataset. Our Model 2 outperformed the previous methods with a 98% accuracy.

Table 4.12: Accuracy Comparison of violence detection methods - Hockey Fight dataset.

Method	Year	Accuracy (%)
Background Subtraction + MobileNet [66]	2022	96
Semantic Segmentation + VGG16 [67]	2022	95
X3D-M [68]	2023	97
Residual +ResNet50+POT [69]	2024	95
Ours	2025	98

Table 4.13: Accuracy comparison of violence detection methods - RLVS dataset.

Method	Year	Accuracy (%)
X3D-M [68]	2023	96.7
Residual +ResNet50+POT [69]	2024	96.5
Ours	2025	98.3

Table 4.14: Accuracy comparison of Violence detection methods - RWF-2000 Dataset.

Method	Year	Accuracy (%)	#Param
MobileNetV2 + LSTM [70]	2022	82.00	4.074M
BrutNet [71]	2024	86.43	3.416M
Ours	2025	88.12	1.904M

Upon meticulous evaluation against benchmark datasets in the domain of violence detection, our method has consistently outperformed other methods. Specifically, within the RLVS dataset [55] in Table 4.13, our model demonstrated a notable accuracy of 98.3%, eclipsing the X3D-M’s [68] 96.7% and the Residual-ResNet50-POT’s [69] 96.5%, signaling a significant stride in model precision. In

Table 4.15: Accuracy comparison of violence detection methods - Surveillance Camera Fight.

Method	Year	Accuracy (%)
Vision-based fight detection [27]	2019	72
Object detection + LSTM [72]	2022	74
Edge Vision [73]	2022	75
Ours	2025	84

the context of the RWF-2000 dataset [52] in Table 4.14, not only did our approach achieve an accuracy of 88.12%, it also showcased a leaner architecture with just 1.904 million parameters, reflecting an optimal balance between accuracy and model efficiency compared to MobileNetV2 + LSTM [70] and BrutNet [71]. Lastly, on the Surveillance Camera Fight dataset [54] in Table 4.15, our algorithm’s performance peaked at an 84% accuracy rate, thereby setting a new standard and underscoring its efficacy over prior methodologies like vision-based fight detection [27], Object detection + LSTM [72], and Edge Vision [73]. These findings affirm our model’s exceptional capability to generalize across varied datasets, marking it a potent tool for enhancing security measures through automated video surveillance.

The contributions outlined in this study are comprehensively validated through a series of methodical developments and empirical evaluations. Firstly, the creation of the Political Violence Dataset, encompassing 480 labeled video clips across four nuanced categories, effectively addresses the existing gap in publicly available resources for political violence detection, as evidenced by its balanced representation and high inter-annotator agreement score of 0.90. Secondly, the innovative key frame extraction algorithm, which integrates both temporal and pixel-level features, significantly enhances the stability and accuracy of frame clustering, as demonstrated by improved silhouette scores and more coherent representative frame selections in Clash and Fire class samples. Thirdly, the fine-tuning of the MoViNet-A0 model on this specialized dataset achieves an impressive accuracy of 92.86% with a parameter count of only 1.904 million, highlighting the model’s efficiency and effectiveness in capturing complex violent behaviors through optimized feature extraction and classification. Lastly, the proposed approach consistently outperforms existing state-of-the-art methods across four benchmark datasets—including Hockey Fight, RLVS, RWF-2000, and Surveillance Camera Fight—thereby underscoring its broad applicability and superior performance in diverse violent event detection scenarios. Collectively, these contributions not only advance the technical capabilities in automated political

violence detection but also provide a robust foundation for future research and practical implementations in enhancing public safety through intelligent video surveillance systems.

4.11 Failure Case Analysis

Misclassification in video violence detection Models, often arises from several factors, shown in Figure 4.15.



Figure 4.15: Example of misclassification: Left shows an individual carrying a gun during Clash action; right shows individuals with closed fists in a public gathering.

One significant reason is the lack of contextual understanding within the models used for classification. For instance, actions such as carrying a gun while engaging in Clash class behaviors (e.g., kicking or pushing) can be misinterpreted as Shooting. Additionally, the visual features associated with aggression—such as closed fists in public gatherings—might not be distinct enough models to differentiate between normal class and clash class. Consequently, enhancing the training datasets to include diverse scenarios could significantly reduce misclassification rates and improve the reliability of these systems.

4.12 Managerial Implications

The proposed video violence detection methodology offers substantial benefits for managers and decision-makers involved in public safety and urban security within smart cities. By integrating the fine-tuned MoViNet-A0 model into existing surveillance infrastructures, organizations can achieve real-time monitoring and automatic classification of violent activities, thereby enhancing the efficiency and reliability of security operations. This capability enables proactive incident response, allowing law enforcement agencies to deploy appropriate resources swiftly based on the specific type of violence detected, such as clashes,

shootings, or fires. Additionally, the system facilitates optimal resource allocation by identifying high-risk areas and times, ensuring that security personnel and equipment are utilized where they are most needed. The comprehensive data generated supports data-driven policy-making, providing valuable insights into patterns and trends of violent incidents, which can inform the development of effective prevention and intervention strategies. Furthermore, the scalability and adaptability of the proposed solution allow it to be deployed across various urban settings, accommodating different sizes and complexities without significant additional costs. By enhancing public safety measures and building trust within the community through reliable and timely violence detection, managers can foster a safer and more secure urban environment. Thus, the implementation of this methodology not only improves operational efficiency and effectiveness but also supports strategic decision-making processes aimed at mitigating violence and ensuring the well-being of citizens in smart cities.

4.13 Real-Time Performance Analysis

The proposed video violence detection algorithm is designed to operate in real-time, a crucial requirement for effective surveillance in smart city environments. This capability stems from several optimizations: (1) the lightweight MoViNet-A0 model, with 1.904 million parameters, minimizes computational overhead; (2) a custom keyframe extraction algorithm reduces the number of frames processed by selecting only the most representative ones using temporal and pixel-level features; (3) preprocessing via histogram equalization enhances input quality, streamlining feature extraction; (4) fine-tuning only the last four layers of MoViNet-A0 achieves 92.86% accuracy without excessive complexity; and (5) optimized hyperparameters and hardware configurations, ensure efficient processing. These design choices enable the system to process video streams with low latency, making it an effective tool for real-time violence detection and rapid incident response.

4.14 Conclusion

In this work, the proposed video violence detection method is thoroughly tested and the effectiveness is shown by extensive experiments with multiple performance measures and comparisons. The Political Violence Dataset with 480 annotated clips has a high annotation reliability as measured by Cohen’s Kappa 0.90 and enables robust model training. Custom keyframe extraction algorithm that

combines temporal and pixel-level features improved clustering stability (Davies-Bouldin index 0.62 vs. 1.05 for standard K-means) and narrative coherence, based on improved temporal smoothness and continuity factor. When the MoViNet-A0 is fine-tuned, it achieves an impressive 92.86% accuracy using only 1.904M parameters and outperforms all the baseline models and the state-of-art results over the benchmark datasets (e.g., 98% on Hockey Fight, 98.3% on RLVS). Even without repetition, preprocessing with histogram equalization increase accuracy by 2.32%, which improve the sharpness of the feature. The importance of temporal clues is verified by ablation experiments and the bottleneck of our work is also addressed in failure cases, e.g., context comprehension. The real-time capacity and practical application of the approach respectively underpin its ability to contribute towards urbans safety.

Chapter 5

Conclusion and Future Recommendations

In this chapter, we concisely summarize the primary findings of this research and highlight its significant contributions to video violence detection, emphasizing the advancements made in accuracy, efficiency, and applicability to real-world scenarios. And we also outline potential directions for extending this research, suggesting avenues to enhance the system’s capabilities, broaden its scope, and ensure its practical deployment in diverse and challenging environments

5.1 Conclusion

This research successfully developed an efficient and accurate system for detecting political violence in videos, addressing key challenges in dataset diversity, keyframe selection, and computational efficiency. By creating the Political Violence Dataset with 480 labeled clips across four categories—Normal, Clash, Fire, and Shooting—we provided a robust resource that captures the nuances of political unrest, filling a critical gap in existing datasets. The fine-tuned MoViNet-A0 model achieved a remarkable 92.86% accuracy with only 1.904 million parameters, demonstrating its suitability for real-time applications in resource-constrained smart city environments. Our custom keyframe extraction algorithm, leveraging temporal and pixel-level features, improved clustering stability and frame representativeness, enhancing detection precision. Comparative evaluations against benchmark datasets—Hockey Fight (98%), RLVS (98.3%), RWF-2000 (88.12%), and Surveillance Camera Fight (84%)—confirmed our approach’s superiority over state-of-the-art methods. These findings underscore the system’s potential to transform passive surveillance into an active tool for public safety, enabling rapid response to violent incidents and supporting data-driven policy-making. By bal-

ancing accuracy and efficiency, this work lays a strong foundation for scalable violence detection systems, offering significant implications for urban security and future advancements in video analysis technology.

5.2 Future Work

Future research can build on this study by expanding the Political Violence Dataset to include a broader range of scenarios, such as protests in diverse geographical and cultural contexts, to enhance model generalizability. Integrating multimodal data, such as audio cues or social media metadata, could further improve detection accuracy by providing additional context to visual features. Efforts should also focus on optimizing the MoViNet-A0 architecture for edge devices, reducing latency and power consumption for real-time deployment in IoT-based surveillance networks. Exploring advanced attention mechanisms or hybrid models combining transformers and convolutional networks may address current limitations in capturing long-range temporal dependencies, particularly for complex, prolonged violent events. Additionally, incorporating explainable AI (XAI) techniques could enhance interpretability, allowing stakeholders to understand model decisions and build trust in automated systems. Finally, conducting field tests in real-world smart city environments will validate the system’s practical efficacy, paving the way for broader adoption and integration into existing security frameworks.

5.3 List of Publications

The following publication is a direct consequence of the research carried out during the elaboration of the thesis, and give an idea of the progression that has been achieved.

1. **M. A.-M. Provath**, M. Rahman, K. Deb, P. K. Dhar, and T. Shimamura, “DeepGuard: Enhancing violence detection in smart cities through deep learning,” *IEEE Access*, vol. 13, pp. 1–19, 2025, doi: 10.1109/ACCESS.2025.3541059.
2. **M. A.-M. Provath**, K. Deb, and K. H. Jo, “XAI-driven lightweight multi-scale ConvLSTM architecture for video violence detection,” in *Proc. 33rd International Symposium on Industrial Electronics (ISIE)*, Ulsan, South Korea, Jun. 18-21, 2024, pp. 1-6, doi: 10.1109/ISIE54533.2024.
3. **M. A.-M. Provath**, M. F. Hossain Saikat, and K. Deb, “Smart surveillance: Real-time violence detection using deep learning,” in *Proc. 3rd International Conference on Mathematical Analysis and Applications in Modeling (ICMAAM-2024)*, Chattogram, Bangladesh, Dec. 5–7, 2024, pp. 1–6.

Bibliography

- [1] AITubo AI, “Flux ai image generator.” [Online]. Available: <https://aitubo.ai/flux-image-generator/>, 2024. [Accessed: 1-Sept-2024].
- [2] United Nations Office on Drugs and Crime, “Global study on homicide.” [Online]. Available: <https://www.unodc.org/unodc/en/data-and-analysis/global-study-on-homicide.html>, 2019. [Accessed: 10-Mar-2024].
- [3] L. Bischoff, “Surveillance camera statistics: which city has the most cctv cameras?.” [Online]. Available: <https://www.comparitech.com/vpn-privacy/the-worlds-most-surveilled-cities/>, Jun 2021. [Accessed: 10-Mar-2024].
- [4] A. Nazir, S. Rao, L. Wu, and L. Sun, “Issues and challenges of aspect-based sentiment analysis: A comprehensive survey,” *IEEE Transactions on Affective Computing*, 2020.
- [5] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar, “Violence detection in video using computer vision techniques,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6855, pp. 332–339, 2011.
- [6] L. Xu, C. Gong, J. Yang, Q. Wu, and L. Yao, “Violent video detection based on mosift feature and sparse coding,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings*, pp. 3538–3542, 2014.
- [7] J. Mahmoodi and A. Salajeghe, “A classification method based on optical flow for violence detection,” *Expert Systems with Applications*, vol. 127, pp. 121–127, 2019.
- [8] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu, “Violence detection using oriented violent flows,” *Image and Vision Computing*, vol. 48, pp. 37–41, 2016.
- [9] Y. Zhu, D. Tian, and F. Yan, “Effectiveness of entropy weight method in decision-making,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–5, Mar 2020.
- [10] C. Ding, S. Fan, M. Zhu, W. Feng, and B. Jia, “Violence detection in video by using 3d convolutional neural networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8888, pp. 551–558, 2014.
- [11] 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.

- [12] E. Ak and O. Tutsoy, "Medical image reasoning with the convolutional neural network-based fuzzy logic," in *Proc. IEEE Access*, Nov 2022.
- [13] T. Pradeep, N. S. Chandrahas, and Y. Fissaha, "A principal component-enhanced neural network framework for forecasting blast-induced ground vibrations," *J. Civ. Hydraul. Eng.*, vol. 2, no. 4, pp. 206–219, 2024.
- [14] J. D. Bodapati, S. F. Ahmed, Y. Y. Chowdary, and K. R. Sekhar, "A deep convolutional neural network framework for enhancing brain tumor diagnosis on mri scans," *Information Dynamics and Applications*, pp. 42–50, 2023.
- [15] M. Demircan and A. G. Pakfiliz, "A novel machine learning approach for optimizing radar warning receiver preprogramming," *Acadlore Trans. Mach. Learn.*, vol. 3, no. 4, pp. 202–213, 2024.
- [16] T. C. Wang, X. W. Wang, and H. Li, "Enhanced prediction accuracy in complex systems: An approach integrating fuzzy k-clustering and fuzzy neural network," *Int. J. Knowl. Innov. Stud.*, vol. 1, no. 1, pp. 30–43, 2023.
- [17] J. Su, P. Her, E. Clemens, E. Yaz, S. Schneider, and H. Medeiros, "Violence detection using 3d convolutional neural networks," in *AVSS 2022—18th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2022.
- [18] X. Xu, X. Wu, G. Wang, and H. Wang, "Violent video classification based on spatial-temporal cues using deep learning," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 01, pp. 319–322, 2018.
- [19] A. S. S. Keçeli and A. Kaya, "Violent activity detection with transfer learning method," *Electron Lett*, vol. 53, pp. 1047–1048, 2017.
- [20] M. S. Kang, R. H. Park, and H. M. Park, "Efficient spatio-temporal modeling methods for real-time violence recognition," *IEEE Access*, vol. 9, pp. 76270–76285, 2021.
- [21] A. Traore and M. A. Akhloufi, "Violence detection in videos using deep recurrent and convolutional neural networks," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, vol. 2020-Octob, pp. 154–159, 2020.
- [22] N. Yucel, H. B. Mutlu, F. Durmaz, E. Cengil, and M. Yildirim, "A cnn approach for enhanced epileptic seizure detection through eeg analysis," *Healthcraft Frontiers*, vol. 1, no. 1, pp. 33–43, 2023.
- [23] S. R. Vemula, M. Vemula, and R. Vatambeti, "Optimizing software vulnerability detection with mdsad-net: A multi-scale convolutional approach enhanced by mantis-inspired optimization," *Inf. Dyn. Appl.*, vol. 3, no. 2, pp. 125–137, 2024.
- [24] A.-M. R. Abdali and R. F. Al-Tuma, "Robust real-time violence detection in video using cnn and lstm," in *2019 2nd Scientific Conference of Computer Sciences (SCCS)*, pp. 104–108, 2019.
- [25] M. Asad, J. Yang, J. He, P. Shamsolmoali, and X. He, "Multi-frame feature-fusion-based model for violence detection," *Visual Computer*, vol. 37, pp. 1415–1431, 2021.

- [26] W. Ullah, T. Hussain, Z. A. Khan, U. Haroon, and S. W. Baik, "Intelligent dual stream cnn and echo state network for anomaly detection," *Knowledge-Based Systems*, vol. 253, p. 109456, 2022.
- [27] S. Akti, G. A. Tataroglu, and H. K. Ekenel, "Vision-based fight detection from surveillance cameras," *IEEE Access*, pp. 1–6, 2019.
- [28] W. Ullah, F. U. M. Ullah, Z. A. Khan, and S. W. Baik, "Sequential attention mechanism for weakly supervised video anomaly detection," *Expert Systems with Applications*, vol. 230, p. 120599, 2023.
- [29] M. Shoaib, A. Ullah, I. A. Abbasi, F. Algarni, and A. S. Khan, "Augmenting the robustness and efficiency of violence detection systems for surveillance and non-surveillance scenarios," *IEEE Access*, vol. 11, pp. 123295–123313, 2023.
- [30] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.
- [32] X. Xu, X. Wu, G. Wang, and H. Wang, "Violent video classification based on spatial-temporal cues using deep learning," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 319–322, 2018.
- [33] H. Wang, A. Kläser, C. Schmid, and C. L. Liu, "Action recognition using optical flow and deep learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 609–620, 2014.
- [34] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Real-time detection of violent crowd behavior," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–6, 2012.
- [35] Q. Sun, W. Yuan, Q. Zhang, and Z. Zhang, "Enhancing session-based recommendations with popularity-aware graph neural networks," *Acadlore Trans. AI Mach. Learn.*, pp. 22–29, 2022.
- [36] H. Z. Gu, J. L. Ma, Y. R. Zhao, and A. Khadka, "Enhanced named entity recognition based on multi-feature fusion using dual graph neural networks," *Acadlore Trans. AI Mach. Learn.*, vol. 3, no. 2, pp. 84–93, 2024.
- [37] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–6, 2012.
- [38] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *Computer Analysis of Images and Patterns*, pp. 332–339, 2011.
- [39] M. Ramzan *et al.*, "A review on state-of-the-art violence detection techniques," *IEEE Access*, vol. 7, pp. 107560–107575, 2019.

- [40] P. Mettes, D. C. Koelma, and C. G. M. Snoek, “The imagenet shuffle: Reorganized pre-training for video event detection,” in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pp. 175–182, 2016.
- [41] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3d residual networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5534–5542, 2017.
- [42] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.
- [43] M. Asad, J. Yang, J. He, P. Shamsolmoali, and X. He, “Multi-frame feature-fusion-based model for violence detection,” *The Visual Computer*, vol. 37, no. 6, pp. 1415–1431, 2021.
- [44] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6546–6555, 2018.
- [45] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.
- [46] C. Feichtenhofer, “X3d: Expanding architectures for efficient video recognition,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 203–213, 2020.
- [47] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, “Temporal relational reasoning in videos,” in *Computer Vision – ECCV 2018*, pp. 831–846, 2018.
- [48] D. Kondratyuk *et al.*, “Movinets: Mobile video networks for efficient video recognition,” 2021.
- [49] W. Ullah, T. Hussain, Z. A. Khan, U. Haroon, and S. W. Baik, “Intelligent dual stream cnn and echo state network for anomaly detection,” *Knowledge-Based Systems*, vol. 253, p. 109456, 2022.
- [50] K. Soomro, A. R. Zamir, and M. Shah, “Ucf101: A dataset of 101 human action classes from videos in the wild.” Center for Research in Computer Vision (CRCV), CRCV-TR-12-01, 2012.
- [51] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–970, 2015.
- [52] Y. Cheng, J. Wang, and H. Zhang, “Rwf-2000: An open large-scale video database for violence detection,” in *ICPR 2020 Proceedings*, pp. 4183–4190, IEEE, 2021.
- [53] J. Nievas, A. Fernández, and A. R. Rodríguez, “Violence detection in video using computer vision techniques,” in *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, pp. 332–339, Springer, 2011.

- [54] Aktı, G. A. Tataroğlu, and H. K. Ekenel, "Vision-based fight detection from surveillance cameras," in *2019 IEEE/eurasip 9th International Conference on Image Processing Theory, Tools and Applications*, (Istanbul, Turkey), pp. 1–6, Nov 2019.
- [55] M. Soliman, M. Kamal, M. Nashed, Y. Mostafa, B. Chawky, and D. Khattab, "Violence recognition from videos using deep learning techniques," in *Proceedings of the 9th International Conference on Intelligent Computing and Information Systems (ICICIS'19)*, (Cairo), pp. 79–84, 2019.
- [56] J. Su, P. Her, E. Clemens, E. Yaz, S. Schneider, and H. Medeiros, "Violence detection using 3d convolutional neural networks," in *AVSS 2022—18th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2022.
- [57] M. A. Provath, K. Deb, and K. H. Jo, "Xai-driven lightweight multiscale convlstm architecture for video violence detection," in *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*, pp. 1–6, 2024.
- [58] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, "Keyframe extraction using clustering-based approach," *International Journal of Electrical and Computer Engineering*, vol. 4, no. 6, pp. 963–970, 2014.
- [59] Y.-H. Liao, A. Kar, and S. Fidler, "Towards good practices for efficiently annotating large-scale image classification datasets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4350–4359, 2021.
- [60] M. A. Provath, K. Deb, P. K. Dhar, and T. Shimamura, "Classification of lung and colon cancer histopathological images using global context attention based convolutional neural network," *IEEE Access*, vol. 11, pp. 110164–110183, 2023.
- [61] M. Rahman, K. Deb, P. K. Dhar, and T. Shimamura, "Adbnet: An attention-guided deep broad convolutional neural network for the classification of breast cancer histopathology images," *IEEE Access*, 2024.
- [62] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.
- [63] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.
- [64] S. Hossain, K. Deb, S. Sakib, and I. H. Sarker, "A hybrid deep learning framework for daily living human activity recognition with cluster-based video summarization," *Multimedia Tools and Applications*, pp. 1–54, 2024.
- [65] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018.
- [66] T. I. Hussain, A. Iqbal, B. Yang, and A. Hussain, "Real time violence detection in surveillance videos using convolutional neural networks," *Multimedia Tools and Applications*, vol. 81, no. 26, pp. 38151–38173, 2022.
- [67] L. Zhang, Z. Zhao, S. Wu, S. Yang, and M. Liu, "A violent video detection method based on image semantic segmentation," *Mobile Information Systems*, 2022.

- [68] V. D. Huszár, V. K. Adhikarla, I. Négyesi, and C. Krasznay, "Toward fast and accurate violence detection for automated video surveillance applications," *IEEE Access*, vol. 11, pp. 18772–18793, 2023.
- [69] N. Honarjoo, A. Abdari, and A. Mansouri, "Violence detection in compressed video," *Multimedia Tools and Applications*, 2024.
- [70] R. Vijeikis, V. Raudonis, and G. Dervinis, "Efficient violence detection in surveillance," *Sensors*, vol. 22, no. 6, p. 2216, 2022.
- [71] M. Haque, H. Nyeem, and S. Afsha, "Brutnet: A novel approach for violence detection and classification using dcnn with gru," *The Journal of Engineering*, 2024.
- [72] F. U. M. Ullah, M. S. Obaidat, K. Muhammad, A. Ullah, S. W. Baik, F. Cuzzolin, J. J. Rodrigues, and V. H. C. de Albuquerque, "An intelligent system for complex violence pattern analysis and detection," *International Journal of Intelligent Systems*, 2021.
- [73] F. U. M. Ullah, K. Muhammad, I. U. Haq, N. Khan, A. A. Heidari, S. W. Baik, and V. H. C. de Albuquerque, "Ai-assisted edge vision for violence detection in iot-based industrial surveillance networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5359–5370, 2022.

Appendix