Applied Intelligence (2026) 56:72
https://doi.org/10.1007/s10489-026-07122-3

Video violence detection using pre-trained VGG19 combined with manual logic, LSTM layers and Bi-LSTM layers

Pablo Negre[3] · **Ricardo S. Alonso**[2,3] · **Javier Prieto**[1] · **Oscar García**[3]

Received: 14 September 2025 / Accepted: 19 January 2026

Abstract

Video violence detection using artificial intelligence plays a key role in public safety applications. Although convolutional and recurrent neural networks are widely adopted for this task, the actual contribution of temporal modeling over strong frame-level representations remains insufficiently analyzed. This work provides a systematic study of video violence detection models under a unified experimental framework. We investigate whether violence can be reliably detected from individual frames without explicit temporal modeling, evaluate the effectiveness of combining CNNs with LSTM and Bi-LSTM layers, and analyze the impact of architectural and hyperparameter choices, including neuron configuration and backbone selection (VGG-16 vs. VGG-19). Experiments are conducted on three widely used benchmark datasets. Our results show that frame-level analysis using a pre-trained VGG-19 network, combined with a simple aggregation strategy, achieves competitive performance, reaching 95% accuracy on Hockey Fights and 96% on Violent Flow. While Bi-LSTM layers can provide moderate improvements of up to 4% over standard LSTM models in certain datasets, these gains are not consistent across all scenarios. Furthermore, variations in hyperparameter configurations do not systematically lead to improved performance. Overall, this study highlights that increased architectural complexity does not always translate into better results and that, in several cases, simple frame-based approaches can rival more complex temporal models. These findings provide practical insights into the cost–benefit trade-off of temporal modeling for video-based violence detection.

Keywords Convolutional Neural Networks (CNN) · Long Short Term Memory (LSTM) · Manual feature · Physical aggression · Video violence detection

1 Introduction

Physical aggressions are a significant issue in our society, impacting individuals globally. This problem affects multiple facets of life, including direct victims and their mental health [1], their families, communities, and the everyday functioning of society (e.g., mobility, tourism, and commerce) [2]. The causes of aggressive behaviour are diverse and include difficulties in emotional regulation, interpersonal conflicts [3], as well as socio-economic and demographic factors [4]. While long- and medium-term social interventions have been proposed, real-time video violence detection using artificial intelligence (AI) offers a direct and scalable technological solution to safeguard victims, reduce human costs, and enable continuous monitoring [5]. In this work, the term *violence* refers to actions involving physical aggression, as defined by human annotations in the benchmark datasets. Accordingly, the role of artificial intelligence is to approximate this human-level labeling

✉ Pablo Negre
pablo.negre@unir.net

Ricardo S. Alonso
ralonso@air-institute.com; ricardoserafin.alonso@unir.net

Javier Prieto
javierp@usal.es

Oscar García
oscar.garcia.garcia@unir.net

[1] BISITE Research Group, Universidad de Salamanca, Patio de Escuelas, 1, Salamanca 37008, Castilla y León, Spain

[2] AIR Institute, Paseo de Belen 9A, Valladolid 47011, Castilla y León, Spain

[3] UNIR (International University of La Rioja), Av. de la Paz, 137, Logroño 37008, La Rioja, Spain

Published online: 03 February 2026

rather than to provide an absolute or universal definition of violence. Among the algorithms used for AI-based video violence detection, a dominant approach combines Convolutional Neural Networks (CNNs) for spatial feature extraction with Long Short-Term Memory (LSTM) layers for temporal modeling [6]. Despite their strong performance, several important **challenges** remain insufficiently explored. In particular, it is still unclear whether explicitly modeling temporal dependencies is always necessary when strong frame-level spatial representations are available. A key open challenge is to determine whether violence detection based solely on frame-by-frame CNN inference can achieve performance comparable to CNN–RNN architectures. Furthermore, although several studies report improvements when combining CNNs with LSTM or Bi-LSTM layers, the actual contribution of bidirectional temporal modeling and the sensitivity of these architectures to hyperparameter choices (such as the number of recurrent neurons) are not yet well understood. Additionally, most existing works rely on VGG-16 as the backbone for spatial feature extraction, while deeper architectures such as VGG-19 have received comparatively little attention. This raises the question of whether increased representational depth can compensate, at least partially, for simplified or absent temporal modeling.

Based on these considerations, the main **motivations** of this study are:

- To systematically compare frame-by-frame CNN-based violence detection against CNN–RNN architectures under a unified experimental framework.
- To assess the actual performance gains obtained by integrating CNNs with LSTM and Bi-LSTM layers, beyond commonly reported accuracy improvements.
- To analyze the impact of architectural and hyperparameter choices, particularly the number of neurons in recurrent and dense layers, on model performance.
- To investigate whether the increased depth of VGG-19 provides more discriminative spatial features than VGG-16 for video violence detection.

In contrast to previous works, this study does not aim to introduce a new or more complex architecture, but rather to provide a systematic and controlled analysis of architectural design choices for video violence detection. The **novelty** of this work lies in demonstrating that competitive performance can be achieved using simple frame-level representations based on VGG-19, even in the absence of explicit temporal modeling. In addition, we explicitly compare the behaviour of VGG-16 and VGG-19 in order to assess whether increased network depth leads to consistent performance gains. The study further analyzes the impact of architectural and hyperparameter choices, including the number of neurons in recurrent and dense layers, on overall performance. Moreover, we compare LSTM and Bi-LSTM architectures under identical experimental conditions to quantify the actual benefit of bidirectional temporal modeling. Finally, by evaluating all configurations across multiple benchmark datasets within a unified experimental framework, this work provides reproducible insights into how dataset characteristics and model complexity jointly affect performance in video violence detection.

To address these objectives, **three architectures are designed** based on a VGG-19 network pretrained on *ImageNet*:

- VGG-19 combined with a handcrafted aggregation strategy based on a minimum number of violent frames.
- VGG-19 combined with LSTM layers for temporal modeling.
- VGG-19 combined with Bi-LSTM layers to evaluate bidirectional temporal dependencies.

Each architecture is trained using multiple configurations of recurrent and dense layer neurons to study their influence on performance. The proposed models are evaluated on three widely used benchmark datasets for video violence detection, and their results are compared against state-of-the-art CNN–RNN approaches.

Overall, the structure of the paper is as follows. Section 2 presents the state of the art and discusses the main challenges in video violence detection. Section 3 describes the proposed methodology. Section 4 reports and discusses the experimental results. Finally, Section 5 presents the conclusions and outlines future research directions.

2 State of the art and challenges

This Section explains the state of the art of video violence detection by means of artificial intelligence, as well as the challenges and motivations of this work. Section 2.1 outlines the solutions that have been applied to address violence in societies. Section 2.2 outlines the basic steps for AI-mediated violence detection. Section 2.3 exposes the most widely used datasets of violence videos, as well as the different types of violence detection algorithms that have been used in the state of the art. Section 2.4 presents recent papers using the combination of Convolutional Nerual Networks (CNN) and Long Short-Term Memory (LSTM).

Section 2.5 discusses the challenges and motivations of this work based on the state of the art analyzed.

2.1 Approaches to addressing violence

This Section discusses various approaches to addressing physical aggression in societies. Several studies focus on understanding the situations or environments that provoke violent behavior, aiming to find long-term solutions to remedy these triggers [1]. Other research looks at the connection between crime rates and urban environments [7]. However, the final defense against violence is real-time detection to alert authorities and gather evidence such as video or images for later identification of those involved.

The implementation of artificial intelligence for video violence detection is crucial for large-scale monitoring without relying on many people to monitor cameras or patrol the streets [8]. This field is growing [6], driven by three key factors: increased use of security cameras [9], advancements in big data platforms [10], and the development of AI algorithms [11–15] for analyzing images and videos. Video violence detection using AI is part of computer vision and a subset of human action detection, with violent actions considered anomalous due to their rarity or unexpectedness in social contexts.

2.2 Basic violence detection steps by means of artificial intelligence

This Section outlines the steps involved in training and operating a violence detection algorithm in real-time, as shown in Fig. 1, divided into *Training* and *Real life situation*. The *Training* phase includes using a labelled dataframe (violence and non-violence videos), where videos are typically cropped to ensure uniform frame count and image size. The data is then preprocessed into the format required by the detection algorithm.

Violence detection algorithms vary but generally involve extracting features (spatial, temporal, etc.) from videos and training an algorithm to learn from these features. Some algorithms process this in two stages (feature extraction followed by learning), others combine both processes, and some use two separate algorithms for feature extraction and learning. Once features are extracted, they are passed to a classifier that determines if the scene is violent. This requires testing various hyperparameter combinations and metrics to find the optimal structure [6].

In real-time detection, a continuous image or video stream is processed. The image undergoes the same preprocessing as the training data, ensuring compatibility with the trained algorithm. Since real-time analysis can be computationally

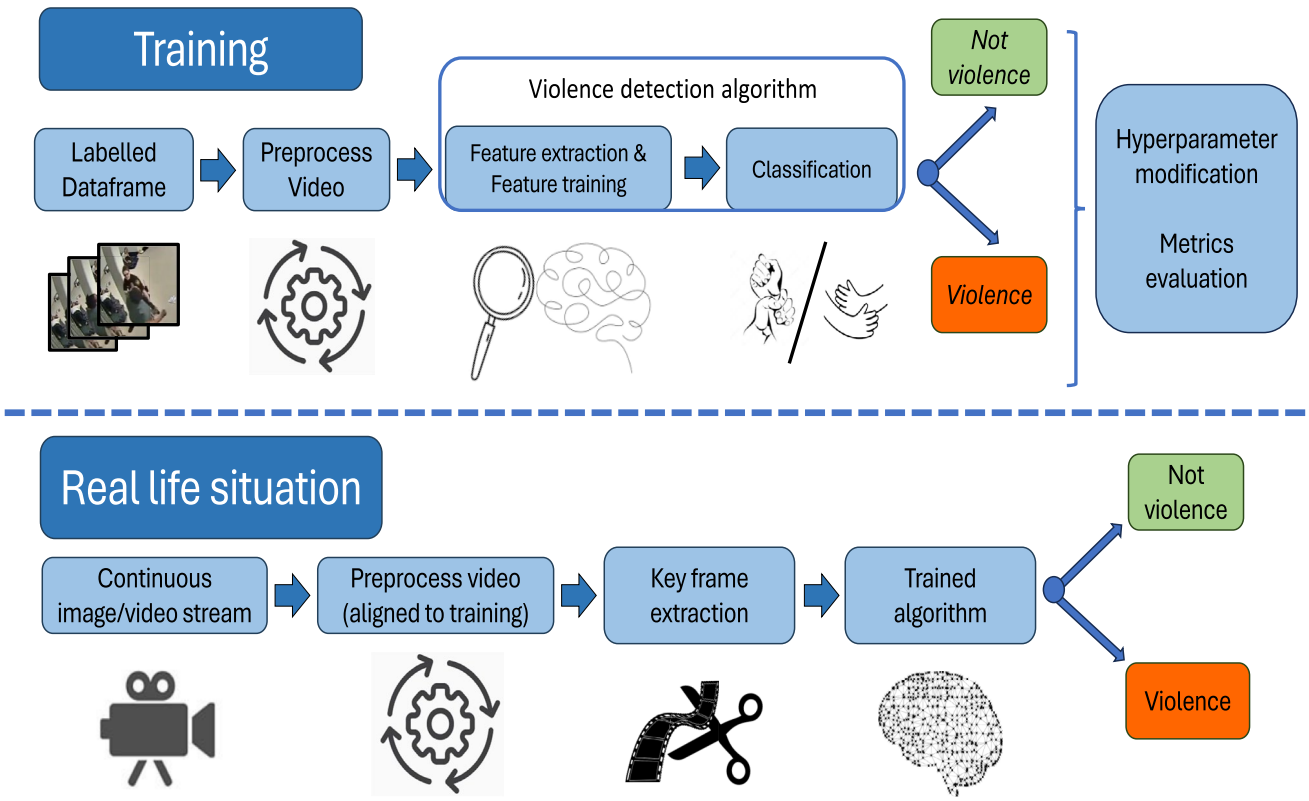

Fig. 1 Basic steps in the implementation of video violence detection algorithms

and economically intensive, lighter techniques can be used to analyze only those frames likely containing violence, although extracting specific frames is optional and not always included in state-of-the-art approaches. Finally, potential violent frames are classified by the trained algorithm as violent or non-violent [8].

2.3 Violence video algorithm types and datasets

The detection of violence, being an anomalous action, requires an optimal dataset for algorithm training. Numerous datasets have emerged, with commonly used ones including *Hockey Fights* [16], *Action Movies* [16], *Violent Flow* [17], *RWF-2000* [18], and *RLVS* [19].

Violence detection algorithms are generally categorized into *Traditional Methods*, using manual feature extraction and traditional Machine Learning, and *Deep Learning Methods* [20–22]. The main types include CNN, LSTM, Manual feature, Skeleton-based, Transformer, and Audio-based methods.

- **CNN**: CNNs extract spatial features from video frames for classification [23]. Advanced versions, like CNN-3D and CNN-4D, extract both temporal and spatial features. Pre-trained CNNs, such as ResNet18, improve accuracy [24, 25].
- **LSTM**: LSTMs capture temporal features and patterns in violence detection. While individual LSTMs are less common, CNN-LSTM combinations are frequently used [21].
- **Manual Feature**: This method involves manual extraction of features such as *Local Binary Pattern (LBP)* and *Fuzzy Histogram of Optical Flow* for training and classification, utilizing AdaBoost and decision trees [26].
- **Skeleton-based**: These methods focus on detecting body positions in videos to infer violence, either through deep learning or manual methods. Techniques like CNN and SVM classifiers or Skeleton Points Interaction Learning (SPIL) are used [27, 28].
- **Transformer**: Transformer-based architectures, like *ViT*, are used to analyze video patches for violence detection. Generative adversarial networks (GANs) also assist in extracting motion features [29].
- **Audio-based**: Audio feature extraction helps in detecting violence, with methods such as Extreme Learning Machine (ELM) used for training, and combining audio with video features for enhanced analysis [30].

In summary, various algorithms are used in violence detection, often combining different approaches, particularly CNN and LSTM, to leverage their strengths [31].

2.4 Violence detection using CNN and LSTM combination

The combination of CNN and LSTM for violence detection is the most widely used and effective architecture in recent research, achieving excellent performance compared to other methods. CNN extracts spatial features from video frames, which are then fed to the LSTM to capture temporal patterns [32, 33]. Within this combination, pre-trained CNNs are more commonly used than non-pre-trained ones.

Tables 1 and 2 summarize papers using CNN-LSTM combinations (or other RNNs like GRUs) [31]. The *CNN* column lists CNNs used, with 68% of the papers employing pre-trained CNNs. The *LSTM* column shows that 50% use LSTMs, 33% Bi-LSTMs, and 5.5% GRU [33–35]. Bi-LSTMs process sequences in both directions, improving performance [36], and Conv-LSTMs combine convolutional and LSTM capabilities [37, 38]. Only one paper compares LSTM vs Bi-LSTM [39], showing minor improvements with Bi-LSTM.

The *Classification* column indicates the use of fully connected layers (FCL) in most studies, with the SoftMax activation function typically applied. Only Jahlan et al. [37] use classical Machine Learning classifiers. The *Train P.T. Layers* column shows that only two papers train layers with violence datasets, with Sharma et al. [34] unfreezing the last four layers for this purpose.

The following columns describe LSTM configurations (layers, neurons) and dense layers (typical values: 2-4). However, no clear trend emerges regarding optimal layer combinations.

Table 2 includes columns for *P.T. Uses F.C.L.*, indicating whether fully connected layers are used in transfer learning, with 6 of 10 papers omitting these layers. The *Learning Rate* column shows values between 10^{-2} and 10^{-4}.

The *Hockey Fight*, *Action Movies*, *Violent Flow*, *RWF-2000*, and *RLVS* columns display the test accuracy of the algorithms on these datasets, which are the most commonly used for violence detection [31]. High accuracies are observed for *Hockey Fight* and *Action Movies*, while real-world datasets like *RWF-2000* and *RLVS* show lower accuracy due to varied lighting and security camera angles.

2.5 Challenges

This Section aims to establish the challenges and motivations of this work on the basis of the shortcomings analysed in the state of the art. As has been set out in this Section 2, multiple approaches have been explored to address the existence of physical aggression in societies worldwide, with real-time detection of violence being the ultimate barrier to victim

Table 1 Violence detection articles which use CNN and LSTM combination. Part 1

Cite	CNN	LSTM	Classification	Train P.T. layers	LSTM layers	LSTM neurons	Dense layers	Dense neurons
Vijeikis et al. [33]	MobileNet V2	LSTM	F.C.L. (N.S.)	N	1	128	2	32,2
Halder and Chatterjee [36]	Own CNN	Bi-LSTM	F.C.L (N.S.)	No P.T.	N.S.	N.S.	N.S.	N.S.
Aarthy and Nithya [40]	PT VGG-16 (ImageNet)	LSTM	F.C.L (Sigmoid)	N	1	50	4	64,64,64,2
Mugunga et al. [22]	PT VGG-16 (ImageNet)	Bi-Conv-LSTM	F.C.L. (SoftMax)	N	3	256	4	1000,256,10,2
Jahlan and Elrefaei [37]	PT MNAS CNN	Conv-LSTM	Random Forest, SVM, K nearest neighbour	N	N.S.	N.S.	N F.C.L	N F.C.L
Traoré and Akhloufi [41]	Two PT EfficienNet-B0 (ImageNet)	Bi-LSTM	F.C.L. (Sigmoid)	N	N.S.	N.S.	3	N.S.
Asad et al. [35]	Two PT VGG-16 (ImageNet) + Wide Dense Residual Blocks (WDRB)	LSTM	F.C.L. (SoftMax)	N	1	512	1	2
Gupta and Ali [39]	PT VGG-16 (ImageNet)	LSTM/Bi-LSTM	F.C.L (N.S.)	N	1	64	N.S.	N.S.
Ullah et al. [20]	Darknet + Residual Optical Flow	M-LSTM	F.C.L (SoftMax)	No P.T.	N.S.	N.S.	N.S.	N.S.
Traoré and Akhloufi [42]	PT VGG16 (INRA)	Bi-GRU	F.C.L. (SoftMax)	N	N.S.	N.S.	3	512,256,2
Islam et al. [43]	Own CNN	LSTM	F.C.L. (SoftMax)	No P.T.	1	128	3	256,16,2
Sharma et al. [34]	PT Xception (ImageNet)	LSTM	F.C.L (SoftMax)	Y (4 last layers)	1	512	3	128,32,2
Talha et al. [44]	CNN	LSTM	F.C.L	No P.T.	N.S.	N.S.	N.S.	N.S.
Mumtaz et al. [32]	PT VGG-19 (ImageNet) + extra CNN layers	Bi-LSTM	F.C.L (SoftMax)	Y (N.S.)	2	128,64	N.S.	N.S.
Srivastava et al. [45]	PT: VGG16/ VGG19/ InceptionV3...	LSTM	F.C.L (SoftMax)	N	1	512	3	1024,512,2
Contardo et al. [38]	PT MobileNetV2 (ImageNet)	Bi-LSTM/ ConvLSTM	F.C.L. (SoftMax)	N	1/1	128/64	2	128/256

Table 2 Violence detection articles which use CNN and LSTM combination. Part 2

Cite	P.T. Uses F.C.L.	Learning rate	Hockey fight	Action movies	Violent flow	RWF-2000	RLVS
Vijeikis et al. [33]	No P.T.	N.S.	99.5	96.1	–	82.0	–
Halder and Chatterjee [36]	No P.T.	10^{-2}	99.3	100.0	98.6	–	–
Aarthy and Nithya [40]	Y	10^{-3}	99.1	–	–	–	–
Mugunga et al. [22]	N	10^{-4}	99.1	100.0	98.4	92.40	–
Jahlan and Elrefaei [37]	No F.C.L	N.S.	99.0	96.0	100.0	–	–
Traoré and Akhloufi [41]	N	10^{-3}	99.0	–	93.8	–	96.8
Asad et al. [35]	N	10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}	98.8	99.0	97.1	–	–
Gupta and Ali [39]	Y	N.S.	97.6/98.8	–	92.2/95.1	–	–
Ullah et al. [20]	No P.T.	10^{-4}	98.0	–	98.2	–	–
Traoré and Akhloufi [42]	N	10^{-4}	98.0	–	96.0	–	90.3
Islam et al. [43]	No P.T.	10^{-4}	97.0	100.0	–	–	–
Sharma et al. [34]	N.S.	N.S.	96.6	98.3	–	–	–
Talha et al. [44]	N	10^{-2}	94.9	92.2	77.3	–	–
Mumtaz et al. [32]	Y	10^{-2}	91.29	–	90.5	90.5	–
Srivastava et al. [45]	Y	10^{-2}	–	–	–	–	–
Contardo et al. [38]	N	N.S.	–	–	–	–	–

protection. The detection of violence using artificial intelligence is composed of multiple stages, with studies having been carried out in recent years in which single and multi-algorithm architectures of a very varied nature (CNN, RNN, Transformers...) have been implemented. Finally, the most widely used architecture in the recent state of the art, both individually and as a combination of algorithms, is the use of CNN and LSTM.

However, none of the articles reviewed that use CNN in combination with some form of RNN, as far as we are aware, test the improvement of the use of RNN over the use of CNN alone. In adittion, only one of the selected articles contains a comparison between LSTM and Bi-LSTM [39]. While it is claimed that the use of Bi-LSTM improves over LSTM, the percentage improvement in the selected datasets does not exceed 4%. Furthermore, the analysis of the selected articles that use a combination of CNN with some form of RNN does not allow us to clearly extract architectures or hyperparameters that are clearly better than others. This is due to: the multiple steps that make up a violence detection algorithm, the few dataframes in common between papers to be able to compare their results and the use of dataframes that do not contain scenes of real aggressions, which results in obtaining accuracies that are too high to make evaluations between different architectures.

Based on these shortcomings in the state of the art, to achieve the following objectives three architectures are designed and created, combining the use of the pre-trained VGG-19 network on the ImageNet dataset along with handcrafted features, LSTM layers, and Bi-LSTM layers. These architectures are intended to evaluate the potential improvements in video violence detection.

- **Evaluate the effectiveness of model combinations:** Investigate whether violence prediction can be achieved through individual frames without temporal relationships between them.
- **Analyze the performance of different architectures:** Compare the performance of combining VGG-19 with LSTM layers versus combining it with Bi-LSTM layers, observing that the improvement when using Bi-LSTM over LSTM does not exceed 4%.
- **Compare VGG-19 and VGG-16:** Assess whether the VGG-19 network can achieve better results than the VGG-16 network in video violence detection, evaluating its effectiveness and potential advantages.

3 Methodology

This Section consists of the explanation of the model architectures that have been decided to develop for this work, based on the Section 2.5 challenges adressed in this work. As discussed in Section 2, the use of pre-trained CNNs outperforms the use of untrained CNNs, where the VGG-16 CNN has been widely used. It has been decided to study the use of pre-trained VGG-16 and VGG-19 CNN. Although VGG-19 is less commonly used and often reports worse results in the literature, in principle its larger number of convolutional layers should provide a higher capacity for extracting complex patterns. Therefore, both architectures will be tested to evaluate their performance in this work. Like many other pretrained CNN-based algorithms such as ResNet, MobileNet, or VGG-16, VGG-19 is trained on a large dataset of images such as ImageNet. These algorithms are thus used, as previously discussed, for spatial feature extraction rather than directly for frame-by-frame violence detection, which would be difficult to classify, as a frame within a violent scene might be considered non-violent without the temporal context it resides in. Nevertheless, this study will propose two methods to verify the effectiveness of VGG (VGG-16 or VGG.19) compared to using VGG in conjunction with RNN layers (specifically, LSTM and Bi-LSTM).:

- The first method consists of analyzing the number of frames that VGG detects as violent and non-violent. It should be noted that, although the datasets used are efficiently trimmed, there may be a small number of frames in each video that may not be considered violent if the violent scene has not yet begun or has already ended.
- The second method consists of establishing a limit on the number of frames detected as violent, beyond which the video is considered violent, in order to jointly analyze a video after the prediction made by VGG-19 frame by frame. This method thus involves combining pretrained VGG with a manual method, to make it comparable to the other architectures in which the video is evaluated as violent or non-violent using LSTM layers and Bi-LSTM layers.

Regarding the proposed architectures based on the combined use of CNN and Bi-LSTM/LSTM layers, two architectures will be used: VGG pretrained together with LSTM layers and VGG pretrained in conjunction with the use of Bi-LSTM layers.

Section 3.1 outlines the architecture of VGG-16 and VGG-19. Section 3.3 will expose the architecture of the violence detection architecture using Pre-trained VGG with minimum violent frame number. Section 3.4 will discuss the architecture created for violence detection using Pre-trained VGG with Bi-LSTM/LSTM layers.

3.1 VGG-16 and VGG-19 architecture

This Section presents the justification for the choice of the pre-trained VGG-16 and VGG-19 architectures as CNNs

to be used for the design and implementation of the architectures in this work [32]. The difference between VGG-16 and VGG-19 is that VGG-16 has 13 convolutional layers, 5 pooling layers and 3 fully connected layers, while VGG-19 has 16 convolutional layers, 5 pooling layers and 3 fully connected layers. While a larger number of convolutional layers means an increase in the number of parameters, which affects computational cost and training time, it also means a greater ability to understand complex structures and patterns.

In Fig. 2 the structure of VGG-19 has been depicted. It can be seen how it expects to receive an array of shape (224,224,3), which indicates 224 pixels wide and high for each RGB colour channel. It then goes through a series of convolutional blocks, which are made up of consecutive convolutional layers; the convolutional layers use filters or kernels consisting of a series of weights that are applied on the array of pixels that is the image from which the network learns increasingly complex patterns. Between each convolutional block there are interspersed Max Poool Layers that reduce the dimension of the pixel matrix that is the image, selecting a maximum value from each region of the image. Once the image has passed through all the convolutional blocks, it is passed to three densely connected layers that act as a classifier. In Fig. 2 it can be seen how the last fully connected layer generates an array with the form (1, 1, 1000), this is because when VGG-19 is trained on the image dataset *ImageNet*, it must classify between a total of 1000 classes. Once the output of the last layer is generated, the SoftMax function is applied to normalise the generated output, ensuring that the sum of probabilities is equal to 1.

Given the use of pre-trained VGG-16 and VGG-19 in ImageNet for violence detection, it becomes necessary to alter the last fully connected layer, which originally contains 1000 neurons, to one with 2 neurons while retaining the weights of the pre-trained network. This adjustment is required due to violence detection being a binary problem (violence vs. non-violence).

3.2 LSTM and Bi-LSTM architecture

Long Short-Term Memory (LSTM) networks are a class of recurrent neural networks (RNNs) designed to model sequential data while alleviating the vanishing gradient problem present in standard RNNs. An LSTM unit incorporates a memory cell regulated by input, forget, and output gates, which control the flow of information over time. This gating mechanism enables the network to capture long-range temporal dependencies, making LSTMs well suited for tasks involving temporal dynamics, such as video analysis [21].

In video violence detection, LSTM layers are commonly used to model temporal relationships between consecutive frames. Typically, each video frame is first processed by a convolutional neural network to extract spatial features, and the resulting sequence of feature vectors is then provided as input to the LSTM. This allows the model to learn temporal patterns that may characterize violent or non-violent behavior across time [6].

Bidirectional LSTM (Bi-LSTM) networks extend this formulation by processing the input sequence in both forward and backward directions. By exploiting information from past and future frames simultaneously, Bi-LSTMs

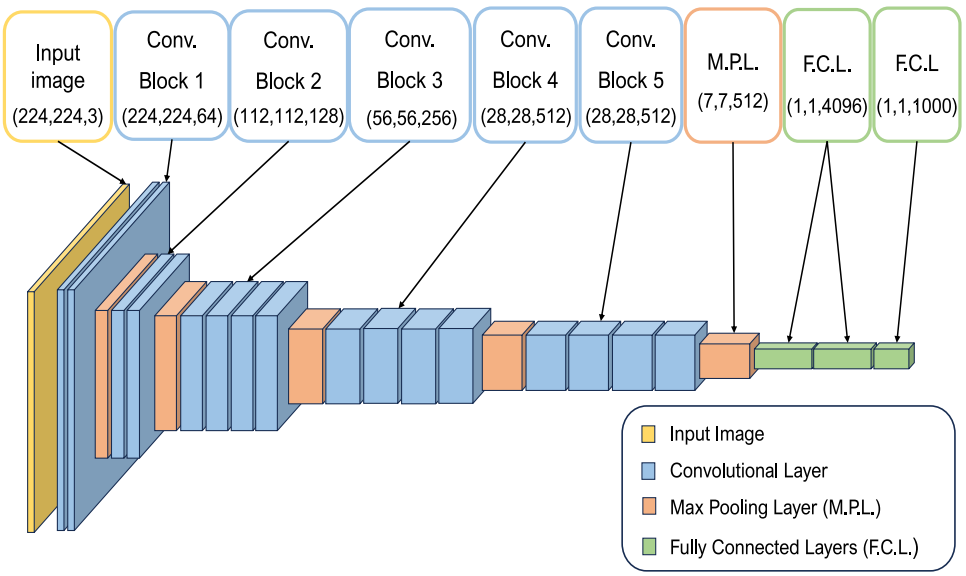

Fig. 2 VGG-19 network architecture

can capture more contextual temporal dependencies when the entire video sequence is available. This bidirectional processing has been reported to improve performance in several video understanding tasks, although it also increases computational complexity and memory requirements.

In this work, both LSTM and Bi-LSTM architectures are considered in order to analyze their respective contributions to video violence detection. Their performance is evaluated under the same experimental conditions and using the same CNN-based feature extractors, allowing a controlled comparison of the impact of unidirectional and bidirectional temporal modeling on classification accuracy.

3.3 Violence detection model using pre-trained VGG with minimum violent frame number

As stated at the beginning of Section 3, in order to compare the accuracy of the prediction solely using pre-trained VGG (VGG-16 or VGG-19), we will examine the number of frames in the test videos predicted as violent and non-violent. Since these results are analyzed frame by frame rather than the video as a whole, as done by the architectures combining VGG with LSTM or VGG with Bi-LSTM, presented in the upcoming sections, this section presents an architecture that predicts frame by frame whether the image is violent or not, and based on the number of frames predicted as violent, the entire video is considered violent or not. This architecture is illustrated in Fig. 3.

Several values will be established as the minimum number of frames detected as violent to consider the video as violent, in order to observe how they vary for each dataframe. It is also worth noting that, to the best of our knowledge, in the recent state of the art, no violence detection model has been developed that combines a CNN with a Manual process, as proposed in this section. Thus, it represents an interesting model to investigate on its own.

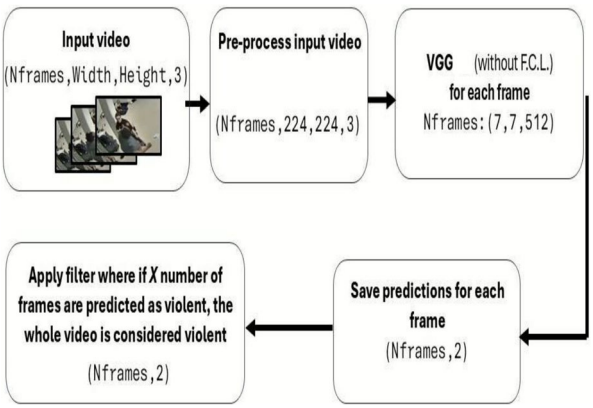

Fig. 3 Violence detection model using pre-trained VGG with minimum violent frame number

3.4 Violence detection model architecture combining pre-trained VGG and Bi-LSTM/LSTM layers

This section outlines the proposed architecture for the combination of pre-trained VGG (VGG-16 or VGG-19) and Bi-LSTM/LSTM layers. This way of performing violence detection in combination using CNN and LSTM consists of extracting the spatial characteristics of the frames of a video on the one hand, and once all the spatial characteristics have been obtained, they are concatenated into a single element (an array) and introduced into the LSTM layers, which then connect with the densely connected layers. This is represented in Fig. 4

First, the video is divided into N_{frames} and resized to the size (224, 224) which is supported by VGG. Then the frames are introduced to VGG one by one. As the spatial features of each frame are calculated by VGG, they are grouped together and added to a single element as they leave the last convolutional layer in the format (7,7,512), resulting in a structure with the form: (N_{frames},7,7,512). Once the pooled features are grouped, a layer *Global Average Pooling 2D* is applied which reduces the dimensionality to the format (N_{frames}, 512) and is then introduced to two Bi-LSTM/LSTM layers and three densely connected layers with Sigmoid activation.

This architecture is processed in two distinct parts, rather than a single layered structure of consecutive neural networks. This means that if it is desired to train some layers of the pre-trained CNN to adjust it to the detection of violence, as is the case in this work, VGG is trained on one side and the Bi-LSTM/LSTM layers together with the dense layers on the other. There is no evidence in the state of the art analysed whether this combination results in lower accuracy because the CNN and (Bi-)LSTM are not trained together as a single architecture. The Bi-LSTM layered architecture, has been used in a conference paper presented as part of this line of research [46].

3.5 Evaluation metrics

The evaluation metrics used in this work are consistent with those commonly adopted in previous studies on video violence detection. They are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

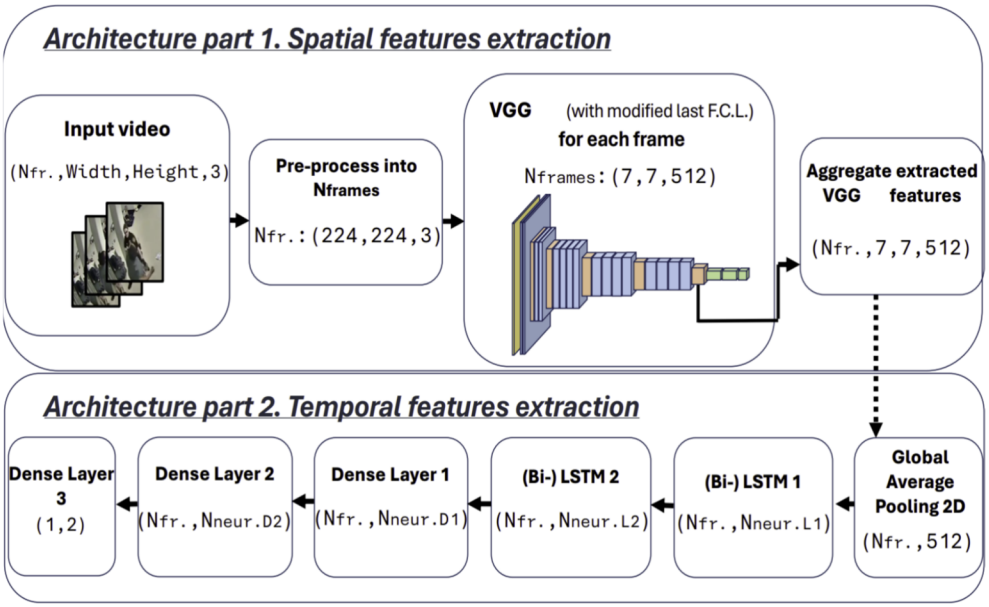

Fig. 4 Proposed video violence detection architecture using pretrained VGG combined with (Bi-)LSTM layers via spatial feature concatenation

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5}$$

The area under the ROC curve (AUC) is also reported as a threshold-independent performance metric, widely used in the literature to summarize the trade-off between true positive and false positive rates.

4 Results

This Section is going to present the results obtained during the training and testing phase of the architecture presented in Section 3. All computations are done on a server running an Intel(R) Core(TM) i9-10940X CPU with 188 Gigabytes of RAM and an NVIDIA GeForce RTX 3090 GPU with 24 GigaBytes of memory.

4.1 Datasets

The datasets used to train and evaluate the proposed models are described in this section. According to recent surveys on video violence detection [31], the most frequently used benchmarks include the *Hockey Fights* dataset, the *Violent Flow* dataset, the *Action Movies* dataset, and the *Real-World Fight 2000 (RWF-2000)* dataset. The use of well-established benchmarks is essential to ensure meaningful comparison with existing state-of-the-art approaches.

In this work, three datasets are selected: *Hockey Fights*, *Violent Flow*, and *RWF-2000*. This choice is motivated by the need to evaluate the proposed methods under diverse and complementary scenarios, while maintaining comparability with a large number of previous studies. The *Action Movies* dataset, although widely used, is excluded because it mainly contains staged scenes with controlled lighting and camera conditions, which differ significantly from real-world surveillance scenarios.

The *Hockey Fights* dataset [16] consists of video clips extracted from National Hockey League (NHL) games. Although the violent events are real, the videos are typically well illuminated and often include close-up views, which makes the classification task comparatively easier. Its extensive use in the literature nevertheless makes it a valuable benchmark for comparison. The *Violent Flow* dataset [17] contains real-world crowd scenes collected from online sources, presenting higher variability in camera motion, background clutter, and crowd density. Finally, the *RWF-2000* dataset comprises surveillance videos captured by security cameras and represents the most challenging scenario due to its realistic conditions, larger scale, and higher intra-class variability.

All three datasets are balanced, containing the same number of violent and non-violent videos. For each dataset, the official data partitioning is adopted, using 70% of the videos for training, 10% for validation, and 20% for testing. This split is consistent with the protocol followed in many previous works and enables direct and fair comparison with state-of-the-art methods.

Although cross-validation is a common strategy to estimate model generalization, it is not adopted in this work in

Table 3 Selected datasets information for training and testing the models

Name	Year	Video number	Median frame number	Frame rate (FPS)
Hockey fights	2011	1000	50	20-30
Violent Flow/Crowd	2012	246	107	25
Real World Fight-2000 (RWF-2000)	2021	2000	150	30

order to remain consistent with the official train/validation/test splits provided by each dataset, which are widely used in previous studies. This choice ensures fair comparability with the state of the art. Moreover, the experimental setup already involves an extensive exploration of architectures and hyperparameter configurations across three datasets, and applying *k*-fold cross-validation would substantially increase the computational cost. Since the goal of this work is to analyze relative performance trends under consistent conditions rather than to optimize a single model configuration, the adopted protocol is considered appropriate.

The *Violent Flow* dataset is the only one among the selected benchmarks with variable video lengths. Since the proposed architectures require a fixed number of frames, videos shorter than the median length (107 frames) are padded by repeating the last frame, while longer sequences are temporally trimmed. This choice is motivated by the fact that violent actions are typically brief and temporally localized events, whose discriminative cues occur within a short time window. Replicating intermediate frames or uniformly stretching the sequence could distort the temporal dynamics of the action and blur the distinction between violence and visually similar interactions such as physical contact or hugging. By duplicating only the final frame, the original temporal evolution of the action is preserved and no artificial motion patterns are introduced, making this strategy a minimally intrusive way to enforce fixed-length inputs while maintaining the integrity of the temporal information required by the model (Table 3).

4.2 VGG-16 and VGG-19 results

This Section is divided into two subsections. Section 4.2.1 will present the results obtained from training the last densely connected layer of pre-trained VGG-19 to classify frames from videos as violent or non-violent. Section 4.2.2 will present the results of the testing process of VGG-19, detecting frames from the testing datasets as violent or non-violent.

4.2.1 Pre-trained VGG-16 and VGG-19 training

This section describes the training procedure applied to the pre-trained VGG-16 and VGG-19 networks. Both models are initialized with weights learned on the *ImageNet* dataset. The original final fully connected layer with 1000 neurons is replaced by a new layer with two neurons, corresponding to the binary classification task (violent vs. non-violent).

Table 4 Training results of VGG-16 and VGG-19 on selected violence datasets using one frame per video

Dataset	CNN	Training videos	Learning rate	Epochs	Validation accuracy
Hockey fights	VGG-16	700	0.001	10	0.96
	VGG-19	700	0.001	10	0.97
RWF-2000	VGG-16	1400	0.001	10	0.68
	VGG-19	1400	0.001	10	0.70
Violent flow	VGG-16	172	0.001	10	0.90
	VGG-19	172	0.001	10	0.93

During training, all convolutional layers are frozen and only the newly added classification layer is optimized. A learning rate of 10^{-3} is used together with the Adam optimizer and the binary cross-entropy loss function. Training is carried out for 10 epochs, which is sufficient for convergence when fine-tuning a single fully connected layer.

In order to reduce overfitting and prevent the model from learning scene-specific or video-specific biases, only one frame per video is used during training. This design choice avoids the strong redundancy that arises when multiple highly correlated frames from the same video are included, which was observed to cause rapid overfitting and poor generalization when all frames were used. By sampling a single representative frame per video, the model is encouraged to learn more general visual cues associated with violent and non-violent content, rather than memorizing background or contextual patterns tied to a specific clip.

All selected frames are resized to (224, 224, 3) before being processed by the network. Since VGG operates on individual images, the training set is constructed as an array of shape $(N_{\text{videos}}, 224, 224, 3)$, where each element corresponds to one frame extracted from a different video. Labels are encoded as one-hot vectors of size 2, with class values indicating violent or non-violent content.

It is important to note that, although the videos in the considered datasets are temporally trimmed so that most frames correspond to violent actions, some frames at the beginning or end of a clip may not strictly depict violence. Using only one frame per video helps mitigate the impact of such temporal noise and reduces the risk of bias introduced by repetitive visual patterns.

Table 4 summarizes the training configuration and validation accuracy obtained for VGG-16 and VGG-19 on the selected datasets. The reported validation accuracy corresponds to the final training epoch. For the Hockey Fights dataset, convergence is typically reached after approximately 5 epochs, whereas RWF-2000 requires around 10

epochs due to its higher variability. Violent Flow converges rapidly, usually within the first 3 epochs. These observations further indicate that extending training beyond 10 epochs does not provide additional benefits when only the final classification layer is optimized.

4.2.2 Pre-trained VGG-16 and VGG-19 testing

In this Section, the pre-trained VGG-16 and VGG-19 test process is explained. The test process consists of making predictions with VGG-16 and VGG-19 using the 20% of the videos not used during the training process (neither as training, nor as validation), where there will be four possible situations: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FP). These four possibilities are the classic ones of any binary classification problem. True Positive indicates that the model predicts that the violent image is violent. True Negative indicates that the model predicts that the non-violent image is non-violent. False Positive indicates that the model predicts that the non-violent image is violent. False Negative indicates that the model predicts that the violent image is non-violent. The goal, of course, is to maximise both True Positive and True Negative, minimising False Positive and False Negative. A False Negative is worse than a False Positive, since if an aggression is occurring and the model does not detect it as such, the victim would not be rescued.

Numerous metrics can be obtained from the confusion matrices, which are used to evaluate the results obtained depending on how the terms are grouped. The most widely used metric is the Accuracy [31], but there are also others such as: precision, recall, F1 Score and Specificity. Also widely used is the AUC metric, which is a measure of the discrimination capacity of a binary classification model, representing the probability that the model correctly classifies pairs of positive and negative observations by varying the decision thresholds. The results obtained for the three selected datasets are included in Table 5.

The results obtained are very promising for the Hockey Fights and Violent Flow datasets, considering that the models are limited to frame-by-frame prediction without incorporating temporal relationships. Using accuracy as the primary evaluation metric, VGG-16 achieved 92%, 64%, and 86% on the Hockey Fights, RWF-2000, and Violent Flow datasets, respectively, while VGG-19 obtained 94%, 64%, and 92% on the same datasets. Although both CNN deliver comparable performance, VGG-19 consistently surpasses VGG-16, particularly on the Violent Flow dataset. The substantially lower results on RWF-2000 can be explained by the challenges of achieving high validation accuracy on such a large and diverse dataset, as well as the inherent characteristics of the videos themselves (e.g., low luminosity in night scenes, distant perspectives, and variability in aggressor count or aggression type). In view of these results, VGG-19 is selected as the preferred architecture for the subsequent stages of this work.

4.3 VGG-19 with minimum violent frame number

This section evaluates the effectiveness of a violence detection strategy based on a pre-trained VGG-19 model combined with a simple frame aggregation rule. Unlike CNN–RNN architectures, this approach does not rely on learning temporal dependencies. Instead, it aggregates frame-level predictions into a single video-level decision by applying a threshold on the number of frames classified as violent.

A video is considered violent if the number of frames predicted as violent exceeds a predefined threshold, referred to as the *minimum violent frame number*. This parameter is not a learnable model weight but a decision threshold that controls the sensitivity of the aggregation process. Consequently, no additional model training is performed at this stage.

Rather than training, a *calibration phase* is carried out on the training split in order to analyse the effect of different threshold values. During this phase, several candidate thresholds are evaluated and their corresponding classification performance is computed. This process does not involve gradient-based optimization or parameter updates; it only serves to characterize how the aggregation rule behaves under different operating points. The purpose of this calibration is to identify representative thresholds and to study how the temporal distribution of violent frames influences classification performance.

Table 5 Testing Pre-trained VGG metrics on selected dataframes

Dataset	CNN	Test video number	Accuracy	Precision	Recall	F1 score	Specificity	AUC
Hockey fights	VGG-16	200	0.92	0.97	0.91	0.94	0.95	0.98
	VGG-19	200	0.94	0.93	0.95	0.94	0.93	0.92
RWF-2000	VGG-16	400	0.64	0.65	0.55	0.61	0.72	0.64
	VGG-19	400	0.64	0.65	0.63	0.64	0.66	0.71
Violent flow	VGG-16	50	0.86	0.81	0.92	0.87	0.79	0.90
	VGG-19	50	0.92	0.87	0.99	0.93	0.86	0.97

Table 6 Minimum frame number to consider a video as violent and the percentage of frames it represents with respect to the total number of frames of the videos of each dataset

Minimum frame number to consider a video as violent	Hockey fights (%)	Violent flow (%)	RWF-2000 (%)
10	25.0	9.3	6.7
20	50.0	18.7	13.3
40	100.0	37.4	26.7
80	–	74.8	53.3
120	–	–	80.0

The selected threshold values are expressed both as absolute numbers of frames and as percentages of the total number of frames per video, allowing a consistent comparison across datasets with different video lengths. Table 6 reports the evaluated thresholds and their relative proportions for each dataset.

For the Hockey Fights and RWF-2000 datasets, good performance is achieved when a relatively small number of frames is sufficient to classify a video as violent. This indicates that violent events in these datasets are often temporally concentrated, so increasing the threshold excessively may suppress true positives. In contrast, the Violent Flow dataset benefits from higher threshold values, reflecting the longer and more homogeneous violent segments typically present in crowd scenes. These observations highlight that the optimal threshold is strongly dataset-dependent and closely related to the temporal distribution of violent content.

After selecting representative thresholds during the calibration phase, the final performance is evaluated on the test split. For completeness, Table 7 reports both the accuracy obtained on the training split during calibration and the accuracy achieved on the test set. The former serves only as a reference to illustrate the behaviour of the aggregation rule on the data used for threshold selection, while the latter reflects the generalization performance on unseen videos.

Overall, these results indicate that violence detection can, in certain scenarios, be effectively addressed using frame-level visual cues combined with a simple aggregation rule, without explicitly modeling temporal dependencies. While temporal modeling remains important for more complex and unconstrained datasets such as RWF-2000, the proposed approach demonstrates that carefully calibrated frame-based aggregation can provide a competitive and computationally efficient alternative when sufficient discriminative information is present at the frame level.

4.4 Pre-Trained VGG-19 + LSTM results

This section presents the results obtained during the training and testing phases of the architecture that combines a pre-trained VGG-19 network with LSTM layers, as illustrated in Fig. 4. Section 4.4.1 describes the training procedure, while Section 4.4.2 reports the evaluation results on the test sets.

4.4.1 LSTM and dense layers training

At this stage, only the VGG-19 network has been trained. The next step consists of training the second part of the architecture shown in Fig. 4, which includes two LSTM layers followed by three fully connected layers. To this end, the spatial features extracted by the trained VGG-19 network are first stored, resulting in an array of shape $(N_{videos} \times N_{frames}, 7, 7, 512)$.

As described in Section 4.2.1, during the training of VGG-19 all frames from all videos are grouped together. However, in order to enable temporal modeling, it is necessary to recover the correspondence between frames and their respective videos so that the LSTM can learn temporal dependencies. Therefore, the extracted features are reorganized into an array of shape $(N_{videos}, N_{frames}, 7, 7, 512)$, corresponding to the output of the fifth convolutional block followed by the Max Pooling layer.

Since LSTM layers expect input data in the form $(\text{Batch Size}, \text{Time Steps}, \text{Features})$, a Global Average Pooling 2D layer is applied to reduce the spatial dimensions, producing an input tensor of shape $(N_{videos}, N_{frames}, 512)$. This tensor is then fed into the first LSTM layer. The corresponding training labels are encoded as vectors of shape $(N_{videos}, 2)$, where [1, 0] represents a non-violent video and [0, 1] represents a violent one.

During training, several hyperparameters are explored, including the learning rate, the number of neurons in each LSTM layer, and the number of neurons in each fully connected layer. The tested values follow those commonly reported in the literature (see Table 1), namely learning rates of 10^{-2}, 10^{-3}, and 10^{-4}, and layer sizes of 64, 128, 256, 512, and 1024 neurons.

The hyperparameter search allows the two LSTM layers and the fully connected layers to have different numbers of

Table 7 Metrics for the best VGG-19 with minimum violent frame number model for each selected dataset

Dataset	Minimum frame number	Train accuracy	Test accuracy	Precision	Recall	F1 score	Specificity
Hockey fights	10	0.97	0.95	0.92	0.98	0.95	0.91
RWF 2000	10	0.80	0.76	0.74	0.89	0.81	0.60
Violent flow	80	0.97	0.96	0.93	1.00	0.96	0.92

Table 8 Training results of the best three models with LSTM layers hyperparameters combinations for the three selected datasets

Dataset	Model	Validation accuracy	Learning rate	LSTM 1	LSTM 2	F.C.L 1	F.C.L 2
Hockey fights	1	0.96	0.0001	256	128	64	64
	2	0.96	0.01	512	512	128	64
	3	0.96	0.001	64	1024	512	64
RWF-2000	1	0.87	0.001	1024	1024	256	512
	2	0.87	0.0001	512	64	128	128
	3	0.87	0.001	128	512	128	256
Violent flow	1	0.96	0.01	64	512	64	128
	2	0.96	0.01	128	64	128	64
	3	0.96	0.0001	512	512	128	1024

Table 9 Testing results of the best three models with LSTM layers hyperparameters combinations for the three selected datasets

Dataset	Model	Test accuracy	Learning rate	LSTM 1	LSTM 2	F.C.L 1	F.C.L 2
	1	0.94	0.0001	256	128	64	64
	2	0.96	0.01	512	512	128	64
Hockey fights	3	0.96	0.001	64	1024	512	64
	1	0.72	0.001	1024	1024	256	512
	2	0.71	0.0001	512	64	128	128
RWF-2000	3	0.71	0.001	128	512	128	256
	1	0.86	0.01	64	512	64	128
	2	0.86	0.01	128	64	128	64
Violent flow	3	0.86	0.0001	512	512	128	1024

neurons, enabling an analysis of how increasing, decreasing, or maintaining the same dimensionality affects performance. Each dataset is trained for 150 epochs, which is three times the number of epochs used for training VGG-19 alone, reflecting the increased complexity of the model. As in the previous stage, the Adam optimizer and the binary cross-entropy loss function are employed.

Given the large hyperparameter search space (1875 possible combinations), Keras-Tuner is used to manage the exploration process. A total of 200 configurations (approximately 10% of the full search space) are evaluated.

Since the goal during training is to maximize the validation accuracy, the three best-performing models identified by Keras-Tuner are retained for each dataset. This strategy is adopted because models achieving similar validation accuracy may generalize differently to unseen test data. It is worth noting that, based on the training results, no clear or consistent relationship can be identified between performance and the number of neurons in the LSTM layers, the number of neurons in the fully connected layers, or their relative proportions.

Table 8 summarizes the hyperparameter configurations of the three best models for each dataset. These models extract temporal information from the spatial features produced by the fifth convolutional block of VGG-19 using two LSTM layers followed by three fully connected layers, the last of which contains two neurons corresponding to the binary classification task. The column *Model* indicates the ranking assigned by Keras-Tuner, where 1 corresponds to the best-performing configuration.

4.4.2 LSTM and dense layers testing

This section presents the results obtained during the testing phase of the three best models selected for each dataset. These models combine spatial features extracted by the pre-trained VGG-19 network with LSTM layers for temporal modeling and fully connected layers for final classification.

Since accuracy is the most commonly reported evaluation metric in the literature on violence detection [31], it is used as the primary criterion to select the best-performing model among the three candidates identified by Keras-Tuner. For each dataset, three tables are provided: (i) one reporting the test accuracy and corresponding hyperparameter configurations, (ii) one presenting the confusion matrix of the selected best model, and (iii) one summarizing the main evaluation metrics of that model.

Table 9 reports the test accuracy achieved by the three candidate models for each dataset. In cases where multiple models achieve the same accuracy, the model ranked higher by Keras-Tuner is selected.

For the Hockey Fights dataset, the second model achieves the highest test accuracy of 96%. For the RWF-2000 dataset, the best-performing model reaches an accuracy of 72%. Finally, for the Violent Flow dataset, the highest accuracy obtained is 86%, achieved by the first-ranked model.

Table 10 Testing metrics of the best model with LSTM layers hyperparameters combination for the three selected datasets

Dataset	Test video number	Model number	Accuracy	Precision	Recall	F1 score	Specificity	AUC
Hockey fight	200	2	0.96	0.96	0.96	0.96	0.96	0.97
RWF-2000	400	1	0.72	0.70	0.77	0.74	0.68	0.79
Violent flow	50	1	0.86	0.85	0.88	0.86	0.84	0.93

Table 11 Training results of the best three models with Bi-LSTM layers hyperparameters combinations for the three selected datasets [46]

Dataset	Model	Validation accuracy	Learning rate	Bi-LSTM 1	Bi-LSTM 2	F.C.L 1	F.C.L 2
Hockey fights	1	0.97	0.01	64	64	64	512
	2	0.97	0.01	512	64	1024	512
	3	0.97	0.01	64	256	64	256
RWF-2000	1	0.87	0.01	512	64	256	64
	2	0.87	0.001	512	256	64	256
	3	0.87	0.001	128	128	64	128
Violent flow	1	0.96	0.001	1024	512	64	256
	2	0.96	0.01	256	256	256	128
	3	0.96	0.01	128	512	128	256

Overall, the best results are obtained on the Hockey Fights dataset, followed by Violent Flow and finally RWF-2000. This ordering is consistent with differences in video length, scene variability, and visual complexity across datasets. In particular, the larger size and higher diversity of the RWF-2000 dataset make convergence toward high accuracy more challenging.

Table 10 presents the main evaluation metrics derived from the confusion matrices of the selected best models, including precision, recall, F1-score, specificity, and AUC.

4.5 Pre-Trained VGG-19 + Bi-LSTM results

This section presents the results obtained during the training and testing phases of the violence detection model based on a pre-trained VGG-19 network combined with Bidirectional LSTM (Bi-LSTM) layers.

4.5.1 Bi-LSTM and dense layers training

The training procedure follows exactly the same methodology described in Section 4.4.1, with the only difference being the replacement of the LSTM layers by Bi-LSTM layers. Therefore, spatial features extracted by the pre-trained VGG-19 network are reorganized and used as input to two stacked Bi-LSTM layers, followed by three fully connected layers for binary classification.

Table 11 presents the hyperparameter configurations of the three best-performing models obtained for each dataset when using Bi-LSTM layers. These results were originally reported in our previous work [46] and are included here for completeness and comparison.

As in the LSTM-based approach, the models are selected according to their validation accuracy during training. For each dataset, the three best-performing configurations are retained. All reported models achieve identical validation accuracy within each dataset, indicating that different architectural configurations can reach similar performance levels when trained under the same conditions.

4.5.2 Bi-LSTM and dense layers testing

The testing phase is conducted following the same procedure described in Section 4.4.2, but using the models that incorporate Bi-LSTM layers instead of standard LSTM units. The evaluation is performed on the held-out test sets for each dataset.

Table 12 reports the test accuracy obtained by the three candidate models for each dataset. In cases where multiple models achieve identical accuracy values, the model ranked highest by Keras-Tuner during the validation stage is selected.

For the Hockey Fights dataset, the third model achieves the best test performance, reaching an accuracy of 97%. In the case of the RWF-2000 dataset, the highest accuracy obtained is 73%, corresponding to the first-ranked model. Finally, for the Violent Flow dataset, the best-performing model reaches an accuracy of 90%, also corresponding to the first-ranked configuration.

Table 13 summarizes the main evaluation metrics of the best-performing Bi-LSTM model for each dataset.

4.6 Hyperparameter result analysis for LSTM and Bi-LSTM combinations

As exposed in Sections 4.4.1 and 4.5.1, models based on the combination of pre-trained VGG-19 with LSTM layers, and Bi-LSTM layers have been trained. In both cases,

Table 12 Testing results of the best three models with Bi-LSTM layers hyperparameters combinations for the three selected datasets

Dataset	Model	Test accuracy	Learning rate	Bi-LSTM 1	Bi-LSTM 2	F.C.L 1	F.C.L 2
Hockey fights	1	0.93	0.01	64	64	64	512
	2	0.95	0.01	512	64	1024	512
	3	0.97	0.01	64	256	64	256
RWF-2000	1	0.73	0.001	1024	1024	256	512
	2	0.73	0.0001	512	64	128	128
	3	0.68	0.001	128	512	128	256
Violent flow	1	0.90	0.01	64	512	64	128
	2	0.82	0.01	128	64	128	64
	3	0.88	0.0001	512	512	128	1024

Table 13 Testing metrics of the best model with Bi-LSTM layers hyperparameters combinations for the three selected datasets

Dataset	Test video number	Model number	Accuracy	Precision	Recall	F1 Score	Specificity	AUC
Hockey fight	200	3	0.97	0.95	0.98	0.97	0.95	0.98
RWF-2000	400	1	0.73	0.74	0.72	0.73	0.75	0.79
Violent flow	50	1	0.90	0.86	0.97	0.91	0.84	0.96

Keras-Tuner was used to obtain the best model combination out of 200 possible hyperparameter combinations. The aim was to observe if certain hyperparameter values resulted in a notable improvement in violence detection results.

For each model separately, the results of the hyperparameter combinations obtained for the three selected datasets were combined, the mean of the obtained validation accuracy results was calculated, and it was determined if any of the hyperparameter values showed a significant deviation from the rest of the values. However, no value stood out notably to affirm that its use clearly resulted in better accuracy. Furthermore, for each model, an analysis was conducted to determine if increasing, keeping the same number, or decreasing the number of neurons between the LSTM or Bi-LSTM layers and the dense layers resulted in a significant improvement in the results. However, none of the options stood out notably. Specific analysis of values for the first and second LSTM layers and dense layers was not performed, as even though approximately 10% of the total possible combinations were covered, there were not enough tested combinations to yield reliable results.

In conclusion, significant finding is the lack of a consistent improvement in accuracy across different configurations of hyperparameters, such as neuron count in LSTM/Bi-LSTM layers and dense layers. Despite testing various combinations, no clear pattern emerged that consistently enhanced performance. This observation highlights that extensive hyperparameter tuning does not always translate to better accuracy, potentially simplifying future model optimization efforts for violence detection. This insight contributes a nuanced understanding to the state of the art, suggesting that high performance may be achievable with a streamlined approach to parameter selection.

4.7 Comparison of results between models

In this work, several baseline models are defined in order to enable a meaningful performance comparison. These baselines include frame-wise VGG-16 and VGG-19 classifiers, as well as simple aggregation-based variants, which provide reference points for evaluating the impact of temporal modeling. While different hyperparameter configurations are explored, the primary goal is not only to analyze sensitivity to parameter choices, but also to assess whether the introduction of LSTM or Bi-LSTM layers yields consistent performance gains over simpler baseline approaches.

This section compares the performance of the different model variants proposed in this work, including frame-wise VGG-19 prediction, VGG-19 combined with a manual aggregation rule, and VGG-19 combined with LSTM and Bi-LSTM layers. To improve clarity and avoid dispersing the main findings across multiple tables, Table 14 summarizes the best-performing configuration for each dataset and model family. Beyond the numerical comparison, the following discussion provides an analysis of the underlying factors that help explain the observed performance differences across models and datasets.

As discussed in previous sections, comparisons involving frame-wise VGG-19 predictions must be interpreted with caution, since this approach operates at the image level rather than directly at the video level. Nevertheless, for the Hockey Fights and Violent Flow datasets, frame-based inference achieves competitive performance when compared with architectures that explicitly model temporal dependencies.

When introducing a simple manual aggregation strategy based on a minimum number of violent frames, the model produces a single video-level decision and improves upon pure frame-wise classification. Interestingly, this approach

Table 14 Summary of best test accuracy results obtained by the proposed model variants across datasets

Dataset	Model variant	Type	Best setting	Test accuracy	Key takeaway
Hockey fights	VGG-16 (frame-wise)	Frame-only	Frame classification	0.92	Slightly lower performance than VGG-19
	VGG-19 (frame-wise)	Frame-only	Frame classification	0.94	Strong performance without temporal modeling
	VGG-16 + minimum violent frames	Manual logic	Min frames = 10	0.93	Comparable but slightly lower than VGG-19
	VGG-19 + minimum violent frames	Manual logic	Min frames = 10	0.95	Simple aggregation improves video-level decision
	VGG-19 + LSTM	Temporal	Best Keras-Tuner model	0.96	Temporal modeling provides limited gain
	VGG-19 + Bi-LSTM	Temporal	Best Keras-Tuner model	0.97	Best overall result on this dataset
Violent flow	VGG-16 (frame-wise)	Frame-only	Frame classification	0.90	Slight degradation with shallower backbone
	VGG-19 (frame-wise)	Frame-only	Frame classification	0.92	High discriminative power from visual cues
	VGG-16 + minimum violent frames	Manual logic	Min frames = 80	0.93	Competitive but below VGG-19
	VGG-19 + minimum violent frames	Manual logic	Min frames = 80	0.96	Outperforms temporal models
	VGG-19 + LSTM	Temporal	Best Keras-Tuner model	0.86	Lower performance than manual aggregation
	VGG-19 + Bi-LSTM	Temporal	Best Keras-Tuner model	0.90	Moderate improvement over LSTM
RWF 2000	VGG-16 (frame-wise)	Frame-only	Frame classification	0.61	Slightly lower robustness than VGG-19
	VGG-19 (frame-wise)	Frame-only	Frame classification	0.64	Strongly affected by dataset complexity
	VGG-16 + minimum violent frames	Manual logic	Min frames = 10	0.74	Competitive but limited by dataset complexity
	VGG-19 + minimum violent frames	Manual logic	Min frames = 10	0.76	Best performance among proposed variants
	VGG-19 + LSTM	Temporal	Best Keras-Tuner model	0.72	Temporal modeling partially helps
	VGG-19 + Bi-LSTM	Temporal	Best Keras-Tuner model	0.73	Slight improvement over LSTM

achieves better performance than the LSTM- and Bi-LSTM-based models on the RWF-2000 and Violent Flow datasets, exceeding them by up to 6% in the latter case. This suggests that, under certain conditions, aggregating strong spatial predictions can be more effective than learning temporal dynamics from limited or noisy data.

These results indicate that violence detection can, in some scenarios, achieve competitive performance using frame-level visual cues alone, without explicitly modeling temporal relationships, even though violent actions are inherently temporal phenomena. This observation highlights the importance of dataset characteristics—such as viewpoint, illumination conditions, motion patterns, and scene variability—in determining the practical benefit of temporal modeling.

Finally, the comparison confirms that architectures using Bi-LSTM layers consistently outperform their LSTM counterparts, although the observed gains remain modest. For the Hockey Fights and RWF-2000 datasets, the improvement is approximately 1%, while for Violent Flow it reaches around 4%. This limited margin suggests that, although bidirectional temporal modeling can provide performance benefits, its added complexity may not always be justified depending on computational constraints and application requirements. Overall, these findings contribute to the ongoing discussion on the cost–benefit trade-off of advanced recurrent architectures in video-based violence detection.

Although Tables 1 and 2 list several approaches that also combine VGG-16 or VGG-19 with LSTM-based temporal modeling, these methods are not architecturally equivalent to the models proposed in this work. Most existing approaches introduce additional convolutional blocks, feature fusion stages, optical-flow branches, or customized temporal modules, and are typically evaluated under heterogeneous training protocols. In contrast, the proposed approach deliberately adopts a simplified and controlled

Table 15 Hockey fights dataset state of art and proposed models accuracy

Cite	CNN	Combination	Hockey fights
Vijeikis et al. [33]	MobileNet V2	LSTM	99.5
Halder and Chatterjee [36]	Own CNN	Bi-LSTM	99.27
Mugunga et al. [22]	PT VGG-16 (ImageNet)	Bi-Conv-LSTM	99.1
Aarthy and Nithya [40]	PT VGG-16 (ImageNet)	LSTM	99.1
Jahlan and Elrefaei [37]	PT MNAS CNN	Conv-LSTM	99
Traoré and Akhloufi [41]	Two PT EfficienNet-B0 (ImageNet)	Bi-LSTM	99
Gupta and Ali [39]	PT VGG-16 (ImageNet)	LSTM/Bi-LSTM	97.6/98.8
Asad et al. [35]	Two PT VGG-16 (ImageNet) + Wide Dense Residual Blocks (WDRB)	LSTM	98.8
Ullah et al. [20]	Darknet + Residual Optical Flow	M-LSTM	98
Traoré and Akhloufi [42]	PT VGG16 (INRA)	Bi-GRU	98
Proposed model	PT VGG19 (ImageNet)	Bi-LSTM	97
Islam et al. [43]	Own CNN	LSTM	97
Sharma et al. [34]	PT Xception (ImageNet)	LSTM	96.55
Proposed model	PT VGG19 (ImageNet)	LSTM	96
Proposed model	PT VGG19 (ImageNet)	Manual Feature	95
Talha et al. [44]	CNN	LSTM	94.9
Proposed model	PT VGG19 (ImageNet) (frame by frame)	N	94
Mumtaz et al. [32]	PT VGG-19 (ImageNet) + extra CNN layers	Bi-LSTM	91.29

design, where a pretrained VGG backbone is used strictly as a feature extractor and temporal modeling is applied in a standardized manner. This allows a direct and fair comparison between frame-level inference, LSTM, and Bi-LSTM variants under identical experimental conditions. Consequently, the objective is not to outperform all previously reported architectures, but to analyze the actual contribution of temporal modeling and architectural complexity when the backbone, training strategy, and evaluation protocol are kept fixed.

Table 16 Violent flow dataset state of art and proposed models accuracy

Cite	CNN	LSTM	Violent flow
Gupta and Ali [39]	PT VGG-16 (ImageNet)	LSTM/Bi-LSTM	92.2/95.1
Jahlan and Elrefaei [37]	PT MNAS CNN	Conv-LSTM	100
Halder and Chatterjee [36]	Own CNN	Bi-LSTM	98.64
Mugunga et al. [22]	PT VGG-16 (ImageNet)	Bi-Conv-LSTM	98.4
Ullah et al. [20]	Darknet + Residual Optical Flow	M-LSTM	98.21
Asad et al. [35]	Two PT VGG-16 (ImageNet) + Wide Dense Residual Blocks (WDRB)	LSTM	97.1
Proposed model	PT VGG19 (ImageNet)	Manual Feature	96
Traoré and Akhloufi [42]	PT VGG16 (INRA)	Bi-GRU	95.5
Traoré and Akhloufi [41]	Two PT EfficienNet-B0 (ImageNet)	Bi-LSTM	93.75
Proposed model	PT VGG19 (ImageNet) (frame by frame)	N	92
Mumtaz et al. [32]	PT VGG-19 (ImageNet) + extra CNN layers	Bi-LSTM	90.47
Proposed model	PT VGG19 (ImageNet)	Bi-LSTM	90
Proposed model	PT VGG19 (ImageNet)	LSTM	86
Talha et al. [44]	CNN	LSTM	77.31

Table 17 RWF-2000 dataset state of art and proposed models accuracy

Cite	CNN	LSTM	RWF-2000
Mugunga et al. [22]	PT VGG-16 (ImageNet)	Bi-Conv-LSTM	92.4
Mumtaz et al. [32]	PT VGG-19 (ImageNet) + extra CNN layers	Bi-LSTM	90.47
Vijeikis et al. [33]	MobileNet V2	LSTM	82
Proposed model	PT VGG19 (ImageNet)	Manual Feature	76
Proposed model	PT VGG19 (ImageNet)	Bi-LSTM	73
Proposed model	PT VGG19 (ImageNet)	LSTM	72
Proposed model	PT VGG19 (ImageNet) (frame by frame)	N	64

Tables 15, 16, and 17 report a comparison between the proposed approaches and previously published methods that combine convolutional neural networks with recurrent architectures for video-based violence detection. The referenced works correspond to the most representative and recent studies employing CNN–LSTM-type pipelines, as summarized in Table 2. This comparison allows situating the proposed models within the broader state of the art under comparable experimental settings.

For the Hockey Fights dataset, all proposed models outperform the results reported by Mumtaz et al. [32], which also relies on a VGG-19 backbone. Several of the proposed configurations achieve competitive performance with respect to other state-of-the-art approaches, although the highest accuracies are still obtained by methods based on VGG-16 or more specialized architectures. This observation suggests that, while VGG-19-based solutions remain competitive, architectural choices and training strategies continue to play an important role in maximizing performance on this dataset.

Regarding the Violent Flow dataset, the results obtained by Mumtaz et al. [32] using a VGG-19 + Bi-LSTM configuration are very close to those achieved in this work. Notably, the proposed approach based on VGG-19 combined with a manual aggregation strategy attains higher accuracy than several methods relying on LSTM-based temporal modeling, including some approaches using pre-trained VGG-16. This further supports the idea that, under certain conditions, simple aggregation mechanisms applied to strong frame-level predictors can be competitive with more complex temporal architectures.

Finally, for the RWF-2000 dataset, none of the proposed models reach the performance levels reported by the best-performing methods in the literature. This dataset appears to be substantially more challenging, likely due to its higher variability in viewpoints, illumination conditions, background clutter, and real-world recording scenarios. These characteristics limit the effectiveness of standard architectures and make generalization more difficult.

Although the accuracy obtained on the RWF-2000 dataset is lower than that reported by some previous approaches using similar VGG–LSTM combinations, the proposed framework offers several complementary advantages. First, unlike many existing works that introduce additional convolutional blocks, optical-flow streams, or heavily customized architectures, the proposed models rely on a deliberately simple and standardized design. This allows a controlled analysis of the actual contribution of temporal modeling without introducing confounding architectural factors. Second, all variants are evaluated under identical training and evaluation protocols, enabling a fair comparison between frame-based, LSTM, and Bi-LSTM approaches. Third, the proposed models exhibit reduced architectural complexity and lower computational requirements, which is particularly relevant for real-time or resource-constrained deployment scenarios. Finally, the lower performance on RWF-2000 further highlights the intrinsic difficulty of this dataset, suggesting that future improvements may require richer supervision, domain adaptation strategies, or additional contextual information rather than increased model complexity alone.

In addition to accuracy, computational efficiency is a key factor for practical deployment in real-time surveillance systems. Frame-based CNN models exhibit low inference latency (on the order of 1.5–1.7 ms per frame in the experimental setup), since each image can be processed independently as it arrives. In contrast, LSTM and Bi-LSTM architectures introduce additional inference overhead of several milliseconds per video sequence, as they require the extraction and storage of feature maps from all frames before temporal modeling can be applied. This sequential processing increases both latency and GPU memory consumption, since intermediate representations must be retained until the full sequence is available. As a result, frame-wise approaches allow streaming evaluation without buffering large amounts of data, enabling deployment on devices with more limited memory resources. Therefore, although recurrent models may yield moderate accuracy improvements, their higher computational and memory requirements must be carefully weighed against their practical benefits in real-time scenarios.

These observations provide insight into the relationship between dataset characteristics, model complexity, and temporal modeling strategies, moving beyond a purely descriptive comparison of numerical results.

5 Conclusions and future work

Physical assaults represent a notable concern in our society, affecting individuals worldwide and directly influencing victims and their psychological well-being [1, 2]. The identification of violence in real-time videos serves as the ultimate barrier in protecting victims, where artificial intelligence has demonstrated excellent results in this task [31].

Firstly, a table listing articles using a combination of CNN and LSTM for violence detection in the recent state of the art has been compiled and analyzed. From this table, several deficiencies in the state-of-the-art have been identified, which this work aims to address. Firstly, there is a lack of understanding about the real contribution of using RNN combined with CNN, compared to using only CNN. Secondly, the actual improvement of using LSTM over Bi-LSTM remains unclear, with only one recent study having

made this comparison, showing a modest 4% improvement. Thirdly, there is no clear relationship between hyperparameter values or combination of these values and an increase in model accuracy. Fourthly, while it has been demonstrated that VGG-16 achieves excellent results in violence detection, the few recent studies using VGG-19 have not achieved better results, although theoretically a greater number of convolutional layers should lead to a better understanding of the scene in question.

Therefore, in this work, several models based on the well-known VGG-19 network with pre-trained weights from the ImageNet dataset have been developed. VGG-19 is used to extract spatial features frame by frame from violence dataset videos. The first model involves only a frame-by-frame analysis of detections made with pre-trained VGG-19, although it is assumed that temporal relationships between frames are lost, providing information on the effectiveness of action analysis through images. The second model incorporates VGG-19 with manual logic implementation, such that a video is not considered violent unless a certain number of frames are detected as violent by pre-trained VGG-19. The third and fourth models involve the combination of pre-trained VGG-19 with the use of LSTM and Bi-LSTM layers, respectively.

Upon comparing the results of the architecture based on pre-trained VGG-19 with manual feature use to architectures based on pre-trained VGG-19 with LSTM and Bi-LSTM layers, it has been observed that CNNs play a significant role in prediction compared to the improvement brought about by the use of LSTM and Bi-LSTM layers. This raises the question of how necessary the temporal relationship between frames is for violence detection in video, although it is clear that an action occurs over time and is not instantaneous. It also raises the question of whether combining CNNs and LSTMs in parallel rather than concatenated would yield better results, as the input to LSTM and Bi-LSTM layers would be the original video rather than the spatial features extracted by pre-trained VGG-19.

As results, it has been observed that even when establishing a wide range of hyperparameter values with learning rate and number of neurons in LSTM/Bi-LSTM layers and dense layers, there are no specific values that clearly increase the accuracy obtained. Neither does the increase, decrease, or same number of neurons notably affect the accuracy obtained.

The violence detection results obtained for the Hockey fights and Violent Flow dataset are promising, while those for the RWF-2000 dataset are less satisfactory due to its more varied and complex scenes. VGG-19 demonstrates strong frame-by-frame predictions with 94% and 92% accuracy for the Hockey fights and Violent Flow dataset, respectively. Predictions using a minimum number of detected violent frames surpass those combined with LSTM and Bi-LSTM on the RWF-2000 and Violent Flow dataset, achieving 76% and 96% accuracy, respectively, and are comparable within a range of 1-2% in the Hockey fights dataset with 95%. Furthermore, the use of pre-trained VGG-19 with LSTM and Bi-LSTM layers achieves 96% and 97% accuracy, respectively, for the Hockey fights dataset, 86% and 90% for the Violent Flow dataset, and 72% and 73% for the RWF-2000 dataset, indicating a marginal 1% improvement with the use of Bi-LSTM in two out of the three datasets.

The proposed models achieve competitive performance on the Hockey Fights and Violent Flow datasets when compared with several recent works based on pre-trained CNN architectures. In some cases, the obtained results are comparable to, or slightly higher than, those reported by approaches using VGG-19 or VGG-16 backbones; however, such comparisons must be interpreted with caution, as differences in data splits, preprocessing strategies, backbone configurations, and training protocols prevent strictly equivalent evaluations. Interestingly, despite the greater depth of VGG-19 and its higher representational capacity, the results suggest that increased architectural complexity does not necessarily lead to better performance in video violence detection. This observation indicates that simpler architectures can be equally effective when combined with appropriate feature aggregation strategies, and highlights the importance of carefully balancing model complexity and practical effectiveness when designing violence detection systems.

As future work, the integration of trustworthy artificial intelligence principles [47] is considered, since the current models do not provide explicit explanations for their predictions. In this context, explainability techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) may be employed to identify the image regions that most influence the decision process. In addition, to improve performance on challenging real-world datasets such as RWF-2000, future research may explore transfer learning from larger and more diverse datasets, as well as data augmentation strategies that simulate realistic camera artifacts (e.g., motion blur, illumination variations, or compression noise). Domain adaptation or domain generalization techniques could also be investigated to reduce the performance gap between controlled and unconstrained scenarios.

Acknowledgements This research has been supported by the project "European Network of AI Excellence Centres: Expanding the European AI lighthouse (dAIEdge)", Grant Agreement Number 101120726. Funded by the European Union, views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Muarifah A, Mashar R, Hashim IHM, Rofiah NH, Oktaviani F (2022) Aggression in adolescents: the role of mother-child attachment and self-esteem. Behav Sci 12(5)
2. Long D, Liu L, Xu M, Feng J, Chen J, He L (2021) Ambient population and surveillance cameras: the guardianship role in street robbers' crime location choice. Cities 115:103223
3. Nurisma SZ, Astuti B (2023) Peace sociodrama: a strategy to reduce junior high school aggression. Int J Soc Serv Res 3(5):1319–1324
4. Enaifoghe A, Dlelana M, Durokifa AA, Dlamini NP (2021) The prevalence of gender-based violence against women in South Africa: a call for action. Afr J Gend Soc Dev 10(1):117
5. Negre P, Alonso RS, González-Briones A, Prieto J, Rodríguez-González S (2024) Literature review of deep-learning-based detection of violence in video. Sensors 24(12):4016
6. Omarov B, Narynov S, Zhumanov Z, Gumar A, Khassanova M (2022) State-of-the-art violence detection techniques in video surveillance security systems: a systematic review. PeerJ Comput Sci 8:920
7. Vomfell L, Härdle WK, Lessmann S (2018) Improving crime count forecasts using Twitter and taxi data. Decis Support Syst 113:73–85
8. Ding D, Ma Z, Chen D, Chen Q, Liu Z, Zhu F (2021) Advances in video compression system using deep neural network: a review and case studies. Proc IEEE 109(9):1494–1520
9. Vosta S, Yow K-C (2022) A CNN-RNN combined structure for real-world violence detection in surveillance cameras. Appl Sci 12(3)
10. Alonso RS, Sittón-Candanedo I, Casado-Vara R, Prieto J, Corchado JM (2020) Deep reinforcement learning for the management of software-defined networks and network function virtualization in an edge-IoT architecture. Sustainability 12(14):5706
11. Dong Y, Jiang H, Liu Y, Yi Z (2024) Global wavelet-integrated residual frequency attention regularized network for hypersonic flight vehicle fault diagnosis with imbalanced data. Eng Appl Artif Intell 132:107968
12. Dong Y, Jiang H, Wang X, Li Z (2025) Entropy-oriented semi-supervised dynamic prototype contrastive learning for rotating machinery fault diagnosis. IEEE/ASME Trans Mechatron
13. Wang X, Jiang H, Zeng T, Dong Y (2025) An adaptive fused domain-cycling variational generative adversarial network for machine fault diagnosis under data scarcity. Inf Fusion, 103616
14. Iqbal I, Odesanmi GA, Wang J, Liu L (2021) Comparative investigation of learning algorithms for image classification with small dataset. Appl Artif Intell 35(10):697–716
15. Iqbal I, Shahzad G, Rafiq N, Mustafa G, Ma J (2020) Deep learning-based automated detection of human knee joint's synovial fluid from magnetic resonance images with transfer learning. IET Image Proc 14(10):1990–1998
16. Bermejo Nievas E, Deniz Suarez O, Bueno García G, Sukthankar R (2011) Violence detection in video using computer vision techniques. In: Computer analysis of images and patterns: 14th international conference, CAIP 2011, Seville, Spain, August 29-31, 2011, Proceedings, Part II 14. Springer, pp 332–339
17. Hassner T, Itcher Y, Kliper-Gross O (2012) Violent flows: real-time detection of violent crowd behavior. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp 1–6
18. Cheng M, Cai K, Li M (2021) RWF-2000: an open large scale video database for violence detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp 4183–4190
19. Soliman MM, Kamal MH, El-Massih Nashed MA, Mostafa YM, Chawky BS, Khattab D (2019) Violence recognition from videos using deep learning techniques. In: 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS). pp 80–85
20. Ullah FUM, Obaidat MS, Muhammad K, Ullah A, Baik SW, Cuzzolin F, Rodrigues JJ, Albuquerque VHC (2022) An intelligent system for complex violence pattern analysis and detection. Int J Intell Syst 37(12):10400–10422
21. Ullah FUM, Muhammad K, Haq IU, Khan N, Heidari AA, Baik SW, Albuquerque VHC (2021) AI-assisted edge vision for violence detection in IoT-based industrial surveillance networks. IEEE Trans Industr Inf 18(8):5359–5370
22. Mugunga I, Dong J, Rigall E, Guo S, Madessa AH, Nawaz HS (2021) A frame-based feature model for violence detection from surveillance cameras using ConvLSTM network. In: 2021 6th International Conference on Image, Vision and Computing (ICIVC). IEEE, pp 55–60
23. Qasim Gandapur M, Verdú E (2023) ConvGRU-CNN: spatio-temporal deep learning for real-world anomaly detection in video surveillance system
24. Tumer C, Isgor B, Koklu M (2025) Violence detection in video using hybrid models based on MobileNetV2
25. Akula V, Kavati I (2024) Human violence detection in videos using key frame identification and 3D CNN with convolutional block attention module. Circ Syst Signal Process 43(12):7924–7950
26. Jaiswal SG, Mohod SW (2021) Classification of violent videos using ensemble boosting machine learning approach with low level features
27. Truong M-T, Hoang V-D (2025) Skeleton-based multi-person action recognition towards real-world violence detection. Eng Appl Artif Intell 161:111987
28. Tran N, Nguyen H, Ly D, Ngo K, Nguyen HD (2025) Advancing violence detection with graph-based skeleton motion analysis. SN Comput Sci 6(6):1–18
29. Alshalawi A, Abdul W, Muhammad G (2025) Advanced detection of violence from video: performance evaluation of transformer and state of the art of convolution of neural network transformer. IEEE Access
30. Meng J, Tian H, Lin G, Hu J-F, Zheng W-S (2025) Audio-visual collaborative learning for weakly supervised video anomaly detection. IEEE Trans Multimed
31. Negre P, Alonso RS, Prieto J, Dang CN, Corchado JM (2024) Systematic mapping study on violence detection in video by means of trustworthy artificial intelligence. Available at SSRN 4757631

32. Mumtaz N, Ejaz N, Aladhadh S, Habib S, Lee MY (2022) Deep multi-scale features fusion for effective violence detection and control charts visualization. Sensors 22(23):9383
33. Vijeikis R, Raudonis V, Dervinis G (2022) Efficient violence detection in surveillance. Sensors 22(6):2216
34. Sharma S, Sudharsan B, Naraharisetti S, Trehan V, Jayavel K (2021) A fully integrated violence detection system using CNN and LSTM. Int J Electr Comput Eng (2088-8708) 11(4)
35. Asad M, Yang J, He J, Shamsolmoali P, He X (2021) Multi-frame feature-fusion-based model for violence detection. Vis Comput 37:1415–1431
36. Halder R, Chatterjee R (2020) CNN-BiLSTM model for violence detection in smart surveillance. SN Comput Sci 1(4):201
37. Jahlan HMB, Elrefaei LA (2021) Mobile neural architecture search network and convolutional long short-term memory-based deep features toward detecting violence from video. Arab J Sci Eng 46(9):8549–8563
38. Contardo P, Tomassini S, Falcionelli N, Dragoni AF, Sernani P (2023) Combining a mobile deep neural network and a recurrent layer for violence detection in videos
39. Gupta H, Ali ST (2022) Violence detection using deep learning techniques. In: 2022 International Conference on Emerging Techniques in Computational Intelligence (ICETCI). IEEE, pp 121–124
40. Aarthy K, Nithya AA (2022) Crowd violence detection in videos using deep learning architecture. In: 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). IEEE, pp 1–6
41. Traoré A, Akhloufi MA (2020) Violence detection in videos using deep recurrent and convolutional neural networks. In: 2020 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, pp 154–159
42. Traoré A, Akhloufi MA (2020) 2D bidirectional gated recurrent unit convolutional neural networks for end-to-end violence detection in videos. In: International conference on image analysis and recognition. Springer, pp 152–160
43. Islam MS, Hasan MM, Abdullah S, Akbar JUM, Arafat N, Murad SA (2021) A deep spatio-temporal network for vision-based sexual harassment detection. In: 2021 Emerging Technology in Computing, Communication and Electronics (ETCCE). IEEE, pp 1–6
44. Talha KR, Bandapadya K, Khan MM (2022) Violence detection using computer vision approaches. In: 2022 IEEE World AI IoT Congress (AIIoT). IEEE, pp 544–550
45. Srivastava A, Badal T, Saxena P, Vidyarthi A, Singh R (2022) UAV surveillance for violence detection and individual identification. Autom Softw Eng 29(1):28
46. Negre P, Alonso RS, Prieto J, Novais P, Corchado JM (2024) Violence detection in video models implementation using pre-trained VGG19 combined with manual logic, LSTM layers and Bi-LSTM layers. In: DCAI 2024. In Press
47. European Commission, High-Level Expert Group on AI (2019) Ethics guidelines for trustworthy AI. https://ec.europa.eu/digital-strategy/news-redirect/65479. Accessed 30 Oct 2024