Acta Polytechnica **65**(3):306–319, 2025
https://doi.org/10.14311/AP.2025.65.0306

Published by the Czech Technical University in Prague

TEMPORAL FUSION STRATEGY FOR VIOLENCE DETECTION: UTILISING CONVOLUTIONAL AND LSTM NEURAL NETWORKS FOR SURVEILLANCE VIDEOS

Khaled Merit[a,*], Mohammed Beladgham[a], Abdelmalik Taleb-Ahmed[b]

[a] *Tahri Mohammed University of Bechar, Department of Electrical Engineering, Laboratory of TIT, Street of Independence, Road of Kenadsa, B.P 417, 08000 Bechar, Algeria*

[b] *University of Valenciennes, UMR CNRS 8520, Laboratory of IEMN DOAE, F-59313 Valenciennes, France*

[*] corresponding author: merit.khaled@univ-bechar.dz

Abstract. In the latest intelligent cities, there is a pursuit for the utmost degree of automation and integration of services. One of the major challenges in the surveillance industry is the need to automate real-time video analysis to identify critical cases. This paper introduces sophisticated models using Convolutional Neural Networks (CNN), specifically MobileNet V3, VGG16, and InceptionV3 networks, as well as networks using LSTM and feedforward networks. These models are designed to accurately categorise videos into two completely separate classes, namely: ("Non-Violence" and "Violence"). The RLVS database is used for this classification task. Various data representations are used by Temporal Fusion approaches. The highest attained outcome was an Accuracy of 91.03 %, and an F1-score of 90.90 %, which is superior to the results obtained in similar research performed on the same database for achieving the goal of recognising actions that are violent in Surveillance Videos.

Keywords: Deep learning, efficient violence detection, temporal fusion, LSTM, automated video surveillance, intelligent cities, video recognition.

1. Introduction

The modernisation of society and urban technologies tend to result in so-called intellectual cities, which are characterised by the capacity to implement communication technologies [1] to provide the highest level of integration and automation of services. Developers develop numerous solutions for monitoring environments and utilising sensors and cameras, among other tools, to obtain reports, support decision-making, and take action. This high level of monitoring generates a large amount of data, a phenomenon known as big data [2], which opens space for so-called data science to seek to generate value by an intelligent analysis of the data [3].

In the search for the greatest possible automation of tasks, intelligent analyses of data are carried out using Machine Learning techniques [4], using a large amount of data to identify patterns, predict phenomena, etc. Surveillance cameras, particularly in the security sector, generate massive amounts of images per second for monitoring. Normally, humans oversee these data, a task that becomes extremely challenging due to the large volume of data generated in real-time by security videos. The solution for this is the practical use of Deep Learning (DL) approaches [5] for the automatic processing of videos, as in [6, 7], to identify, for example, risk situations and/or incidents, such as traffic accidents, fire, robbery, violence and indecent assault.

There is a need for a more efficient form of analysis in the surveillance camera monitoring system. Human analysis can be inaccurate and not fast enough to generate immediate action. It is proposed, then, a system that automatically analyses a video, can identify a violent situation, and, in a much more agile way, supports decision-making and action. Environments that require strong monitoring, such as public roads, shopping malls, metro stations, and penitentiaries, can greatly benefit from this application. Once it identifies a violent incident, it can immediately activate responsible entities, thereby preventing escalations of these incidents.

For the problem of violence detection in videos, [8] presents a solution based on neural networks, both convolutional and recurrent, with results superior to those in [9, 10], which use only convolutional networks. This suggests that the best approach to the task in question is the joint use of recurrent and convolutional networks. This paper contributes by proposing the use of Convolutional LSTM (ConvLSTM), a model that combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs). It also looks at how to use and rate video representation methods based on Temporal Fusion (TF), which worked better than previous research. The structure of the present document is as follows: The introduction is presented in Section 1. Section 2 presents works related to the context; Section 3 explains the materials and methods used; and Section 4 presents and analyses the results obtained. Section 5 presents the conclusions and future work prospects.

Dataset	No. Total of samples	Frames resolution (height × width)	Non-violent samples (NV) Frames per second (fps)	Non-violent samples (NV) No. of Samples	Violent samples (V) Frames per second (fps)	Violent samples (V) No. of Samples	Size
RLVS	2 000	variable (224–1 080) × (224–1 920)	29.5	1 000	29.5	1 000	1.9 GB
Hokey Fight	1 000	288 × 360	25	500	25	500	223.1 MB
Violent Flows	246	240 × 320	25	123	25	123	166.8 MB
Action Movies	73 127	576 × 720 480 × 720	29.9	100	25	100	769.9 MB

TABLE 1. Summary of experimental datasets (RLVS, HF, VF, and AM).

2. Related work

Previous researchers have employed various methods, including manual feature extraction, deep learning, and classic computer vision methodology, to recognise violence in videos. Research indicates that deep learning approaches yield higher accuracy in violence recognition compared to manual methods.

Deep learning techniques, notably CNNs, have demonstrated success in research related to computational vision, as indicated by [11]. For example, [7, 12] demonstrated the widespread application of CNNs, which are based on the human vision system, to image classification and content identification problems.

Numerous studies using CNNs for the task of detecting violence have been conducted. In [13], the construction of convolutional networks for the binary classification of violent or non-violent incidents is presented, while in [10], the authors propose solutions based on Transfer Learning (TL) from models, also from CNNs, trained with the ImageNet database [14]. Other works also present the combination of other models with convolutional neural networks, as in the case of [9], which presents a solution based on Hough Forest and CNNs.

Recurrent neural networks show a good performance when dealing with temporal series [15]. Since videos have temporal characteristics between their frames, we can exploit this ability to increase the performance of the models. This process can be found in [8], which uses convolutional networks and LSTM to classify violent videos.

Therefore, a deep learning model based on CNNs and RNNs (Recurrent Neural Networks) of the LSTM type is proposed, resulting in a ConvLSTM network for the exploration of both spatial and temporal characteristics in videos to contribute to the task proposed in the automatic classification of videos for the detection of violent incidents. This work differs from other works be using the techniques mentioned and also making use of TL for the composition of the proposed models, in addition to applying techniques of video representation based on Temporal Fusion. It is not present in other works that tackle the same task.

3. Materials and methods

In this section, we describe the methods and materials used to perform the experiments conducted for the violence detection task. The subsections are divided into: Datasets (experimental data), which presents and evaluates the database used; preprocessing, which informs how machine learning will be used in combination with the database presented; Temporal Fusion (TF), which addresses the techniques for representing the input data; methodology, which presents the proposed machine learning models for performing the task in question; and performance evaluation, which presents the methods used to evaluate the suggested models.

3.1. Datasets

Researchers use multiple test datasets to assess the effectiveness of various methods. These datasets commonly include Hockey Fighting (HF), Violent Flows (VF), Action Movies (AM), and Real-Life Violence Situations (RLVS), which are widely used in existing approaches. Table 1 provides additional information about each of the datasets.

(1.) Hockey Fight (HF) dataset: Bermejo et al. [16] present the dataset. Its purpose is to measure the effectiveness of violence detection systems in uncrowded scenes. The collection comprises 1 000 movies, 500 of which depict violent scenes, and the other 500 depict non-violent scenes. The videos are recorded from National Hockey League (NHL) games. The minimum, mean, and maximum number of frames are, respectively, 40, 41, and 49. Additionally, the video clips have a resolution of 288 × 360 pixels. Figure 1 depicts both violent and non-violent scenarios in its first and second rows from this dataset.

(2.) Violent Flows (VF) dataset: Hassner et al. [17] compiled the dataset. Its purpose is to evaluate the effectiveness of violence detection techniques in crowded environments. The collection has 246 videos, all of which have a resolution of 240 × 320 pixels. The maximum, mean, and minimum number of frames are 161, 89, and 26, respectively. Figure 2 displays the violent and non-violent events of the Violent Flows database in its first and second rows, respectively.

(3.) Action Movies (AM) dataset: Bermejo et al. [16] compiled this dataset. Similar to the Hockey Fights dataset, this collection also includes scenes with low

(a). Violent scenes.

(b). Non-violent scenes.

Figure 1. The HF dataset's representative frames.

(a). Violent scenes.

(b). Non-violent scenes.

Figure 2. The VF dataset's representative frames.

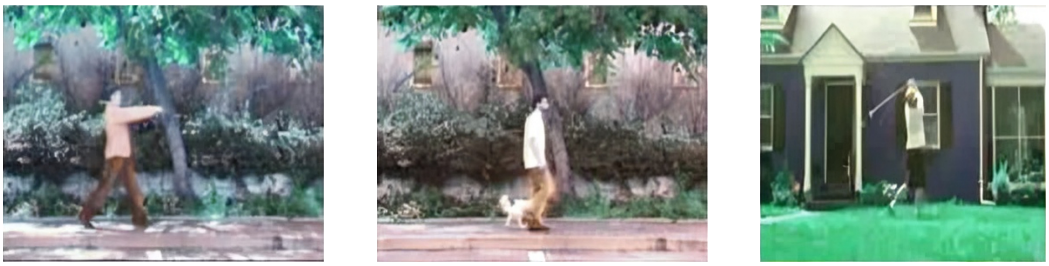
population density (uncrowded scenes). However, it offers a diverse range of scenes and resolutions. The collection comprises 200 videos, consisting of 100 scenes depicting violence and 100 scenes without violence. Action films provide violent videos, while public scenes provide non-violent ones. The maximum, mean, and minimum number of frames are 60, 49, and 42, respectively. The mean resolution of the dataset is 515×720 pixels. The Figure 3 displays both violent and non-violent scenarios from the dataset in its top and bottom rows.

Although both the Action Movies (AM) and Hockey Fight (HF) datasets were introduced by Bermejo et al. [16], they are independent and do not share any overlapping video samples. The AM dataset consists of scenes from action movies and public scenes, while the HF dataset is recorded from National Hockey League (NHL) games. These datasets were designed for different contexts (movies vs. sports), ensuring that their content is distinct and non-overlapping.

The variation in the number of images (frames) across datasets is due to differences in video duration, frame rates, and the nature of the scenes captured. For example, the Hockey Fight dataset contains shorter clips with fewer frames, while the Action Movies dataset includes longer scenes with more frames. To address this variability, we normalised the number of frames per video during preprocessing by extracting 2 frames per second for all videos. This ensures a consistent temporal representation across datasets and minimises the impact of frame count variability on the model performance.

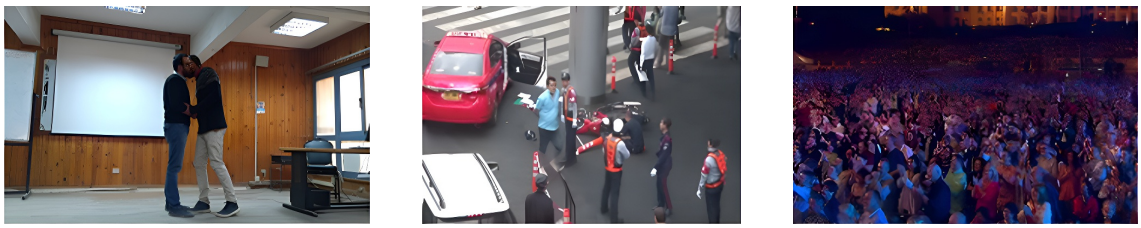
In this paper, the Real-Life Violence Situations Dataset (RLVS-2019) database was used [18], which consists of 2 000 examples of short videos, with an average duration of 5.4 s and average sampling of 29.5 fps,

(A). Violent scenes.



(B). Non-violent scenes.

FIGURE 3. The AM dataset's representative frames.



(A). Violent scenes.

(B). Non-violent scenes.

FIGURE 4. The RLVS-2019 dataset's representative frames.

separated into two categories: "Violence" and "Non-Violence". We divide the examples evenly between the classes, ensuring a balanced basis and minimising the risk of overfitting. Examples of the "Violence" class are samples of videos captured by security cameras, film excerpts, and real videos found on YouTube. The "Non-violence" class also includes real videos and film excerpts depicting everyday situations, such as sports practice, conversations, openings, walking, and etc. Figure 4 illustrates, using frame sequences, a sample from each class in the database. The dataset is available for free, online on [19].

When analysing the frames individually, there is a certain difficulty in understanding the context of the scene in question, which would make it difficult to detect violence from static images. In the third frame of the example of the "Violence" class, for example, if analysed individually, there is no way to state whether or not it represents a violent scene; however, when analysing the frames in sequence, the characteristics that differentiate each of the classes are more evident. Thus, for the video classification task regarding the two classes mentioned, the use of deep learning models for the analysis of temporally dependent frames is potentially a solution to this problem.

3.2. PREPROCESSING

The database presented will be the one used to carry out supervised machine learning with the task of binary classification between the classes "Violence" and "Non-Violence". We perform this task in two stages: the training phase and the validation phase. We use the K-fold cross-validation technique for this, with K equal to 10. We divide the database into 10 equal

parts, also known as splits, and use each split as a validation set and the others as a training set at each iteration. This is repeated 10 times, that is, until all splits have been used as a validation set.

During the training phase of each iteration, the examples of the training set (the result of the union of all subsets except the validation set) will be presented to the neural networks, whose architectures will be presented in Subsection 3.4, in a total of 50 epochs, so that there are adjustments to their trainable parameters, thus generating intelligent models. To ensure the generalisation power of the models, in the validation phase of each iteration, examples of the validation set, not used in the training phase, will be presented to the models, and their outputs obtained will be compared to the desired outputs for each example to evaluate their performance according to the metrics presented in Subsection 3.6. After all training and validation rounds have been performed, the performance metrics will be considered using the arithmetic method and the standard deviation of the performance obtained in the cross-validation.

3.3. Temporal fusion

The use of videos as predictive attributes in machine learning models presents a significant challenge. The main concern is to maintain the temporal characteristics in these data without the number of frames used, making the training of these models unfeasible.

An approach to the representation of videos is TF, which, according to [20], presents several forms of association between the frames of a video. Figure 5 illustrates these different forms of data representation.

(1.) Late Fusion: Consists of the use of the first and last frames of each example to analyse the beginning and end of the action represented in each video. This approach first processes each of the frames individually, then processes the outputs in a related manner, as illustrated in Figure 5a. The technique has this name, Late Fusion (LF), because it deals with the fusion, after processing, of the characteristics of the frames with the greatest temporal distance from each other, that is, that delimit the beginning and end of a scene.

(2.) Early Fusion: Consists of processing by a model of concatenated frames of each video. Figure 5b illustrates this technique of representing the input data. This technique is named Early Fusion (EF) because data fusion occurs before any frame undergoes processing.

(3.) Slow Fusion: This technique involves separate model processing of the video frames, followed by related processing of the outputs, as shown in Figure 5c. This technique, known as Slow Fusion (SF), associates several frames temporally after undergoing specific processing.

The authors also present the single-frame technique in [20], which selects only one frame from each video as its representative. This work treats the problem as an image classification task and does not address the technique. This work exclusively used techniques for video representation. The slow fusion and early fusion approaches require the selection of video frames to accurately represent the subject. The number of frames chosen for these approaches can be neither too large, to the point of making training unfeasible, or too small, to the point that there are significant losses in the temporal characteristics of the video. We decided to use 2 frames per second, as suggested in [20], for videos with an average duration of 10 frames per video, given that the videos in the database average at about 5 seconds. We applied this number of frames to all base videos, resulting in a variation in frames per second sampled while maintaining a fixed number of frames per video. We extracted the frames from videos with duration times that differed from the average, evenly spaced over time.

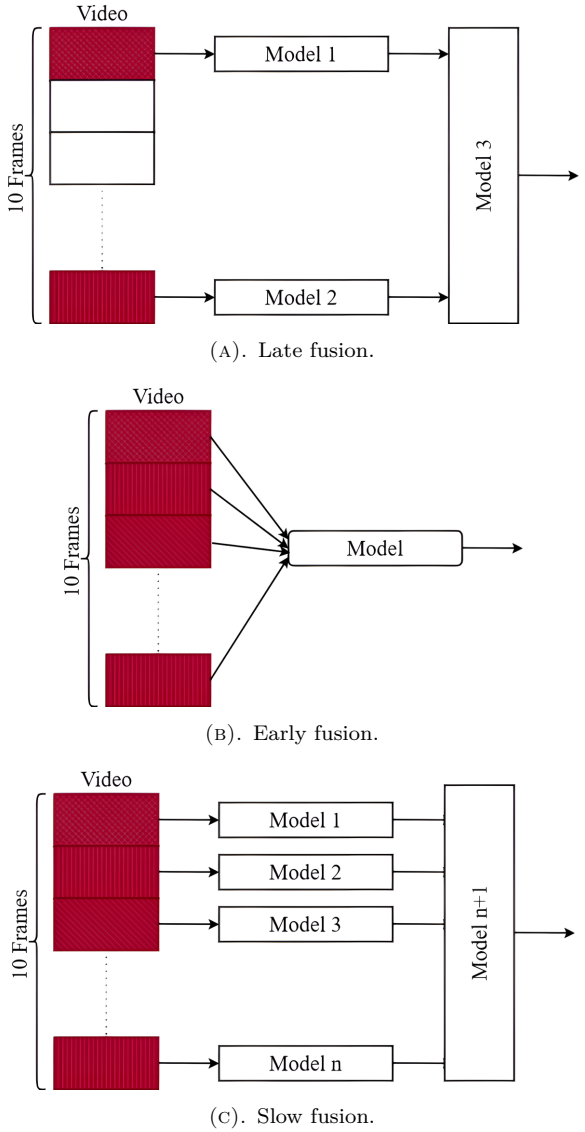
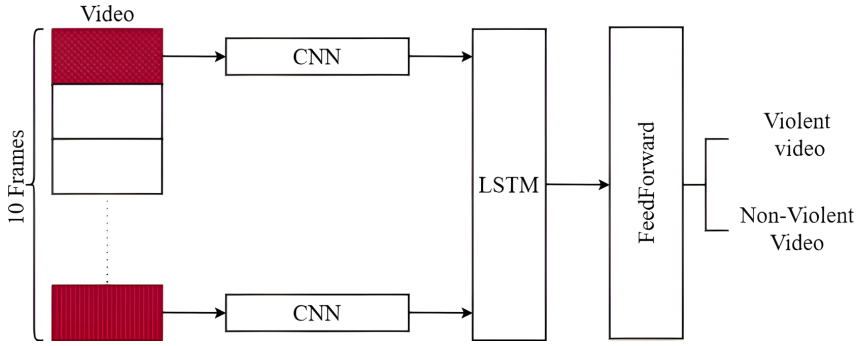
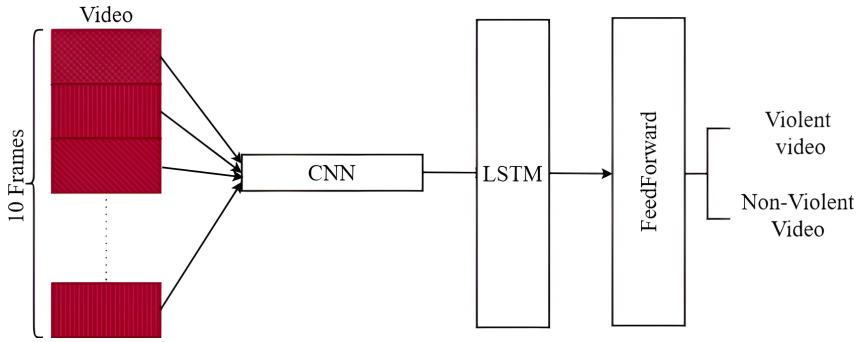


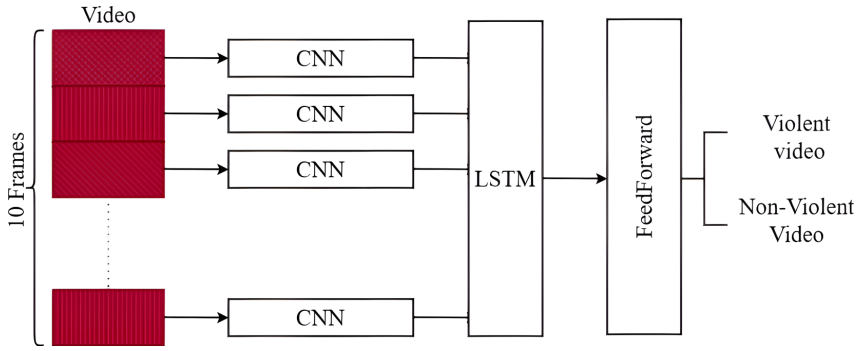
FIGURE 5. General architecture of temporal fusion techniques for the representation of a video through the association of frames.



(A). Based on LF.



(B). Based on EF.



(C). SF-based.

FIGURE 6. Presentation of architectural proposals.

3.4. METHODOLOGY

3.4.1. PROPOSED MODEL

Subsection 3.3 demonstrated how the videos will be presented to the machine learning models that, as will be seen below, are based on deep learning methods. In [20], the authors, in addition to proposing temporal fusion techniques to support the representation of data in video classification tasks, suggest that the models used are based on Convolutional Neural Networks, as a way of extracting characteristics of the situations found in the videos, followed by feedforward layers of perceptron neurons, which perform pattern recognition from the characteristics extracted in the previous layers. This approach is widely used in several applications of DL, especially for computational vision tasks. However, videos possess a characteristic that static images lack: temporal dependence. An appropriate technique for extracting this type of characteristic can better explore the temporal sequence between frames. Therefore, we use LSTM frames for this purpose, as they represent the state of the art in extracting temporal characteristics from signals.

The technique relies on ConvLSTMs, where the LSTM layers analyse the temporal features in their inputs, and the convolutional layers then extract these spatial features. Thus, the strategy is to present the input data to the convolutional layers so that the extraction of spatial characteristics of the frames occurs, followed by an LSTM layer that extracts temporal characteristics from the data resulting from the convolutional layers, and finally, to perform a classification of the patterns through the use of feedforward layers of perceptron neurons. Figure 6 provides a block-by-block overview of the ConvLSTM architecture for the models we build, segmented by the temporal fusion techniques we present.

Figure 6 allows us to verify the flow of various neural network types, adhering to the CNN-LSTM-

feedforward sequence, and the submission of video frames to the models using temporal fusion technique. We performed several empirical tests for each of the architectures presented in this figure to determine the parameters and hyperparameters of the neural networks. We use the convolutional layers of the canonical networks, namely MobileNet V3, InceptionV3, and VGG16, for the convolutional neural network blocks. We apply the transfer learning technique, which uses the weights derived from training with the ImageNet database. This approach does not use the Fine-Tuning (FN) technique and only adjusts the weights of the final layers added to the models. Additionally, we built a simple neural network using the parameters and hyperparameters depicted in Figure 7 and trained it using the RLVS-2019 database.

In Figure 7, the parameters for the convolutional layers are the number of convolution numbers, kernel size, and activation function. For the pooling layers, the parameter informed represents the dimension of the pooling matrix, and the parameter informed for the batch normalisation layers represents the momentum. We used the "Adam" optimizer for all models.

We are going to test the networks shown as convolutional blocks for all of the architectures in Figure 6, along with the output block that has the last layers of the LSTM and feedforward types. These final layers were empirically defined, as illustrated in Figure 8, in which the parameters defined for the feedforward layers are the number of perceptron neurons and activation function; for the LSTM layers, the reported parameter represents the number of LSTM neurons; and for the dropout layers, it represents the normalisation rate.

Therefore, for each of the temporal fusion techniques used, 4 models will be tested based on the junctions of the 4 convolutional blocks mentioned with the output block, whose layers are presented in Figure 8.

3.5. Simulation parameters

We implement our models in TensorFlow, and all experiments are carried out on a machine with an Intel(R) Core(TM) i7-12800H CPU @ 4.0 GHz, 32 GB of memory, 8 GB Nvidia Quadro RTX A2000, and Windows 11 Pro as an operating system.

Table 2 presents the simulation parameters for the proposed research work. Every parameter has been carefully chosen with its optimal values to ensure the most favourable results of the study. We specifically targeted these parameters for performance evaluation, focusing on accuracy and computational efficiency. Furthermore, we have depicted the operating system details and necessary simulation specifications with appropriate values. Table 2 comprises two columns, one for the parameter name and the other for the corresponding parameter value.

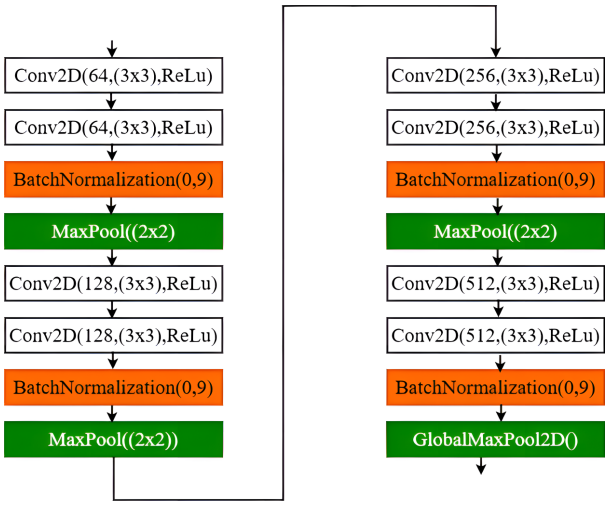


Figure 7. A neural network of simple architecture is built.

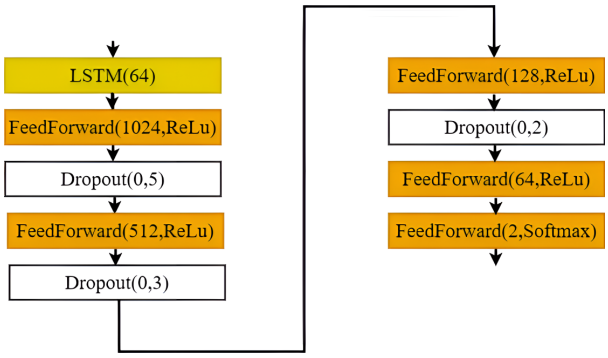


Figure 8. Final layers of the proposed models that will be used after the convolutional layers.

3.6. Performance evaluation

The results obtained will be evaluated according to the Accuracy and F1-score performance metrics. As shown in Equations (1)–(4):

(1.) Accuracy: The accuracy is given by:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \qquad (1)$$

where,

- A true positive (TP) occurs when the model accurately predicts violent activity and the video indeed contains violent content.
- A true negative (TN) occurs when we correctly predict that a video is non-violent, and it is indeed non-violent.
- A false positive (FP) occurs when the model predicts violent action but labels the video as non-violent.
- A false negative (FN) occurs when the model predicts a video to be violent while labelling the video as non-violent.

This metric represents the proportion of correct classifications among the total number of classifi-

Parameter	Value
Simulation software	Python
Libraries	OpenCV, Matplotlib, NumPy, PyLab, SciPy, Time
Implementation environment	TensorFlow, CUDA, Keras
Dataset	RLVS-2019
The operating system	Windows 11 Pro 64-bit (x64)
CPU	Intel(R) Core (TM) i7-12800H CPU @ 5.0 GHz
GPU	Nvidia Quadro RTX A2000 GPU 8 GB up tp 24 GB
RAM	32 GB
CUDA	9.0.176
TensorFlow	2.3.0

TABLE 2. Configuration of the experimental environment.

Convolutional block	Accuracy	Accuracy standard deviation	F1-score	Standard deviation of F1-score	Training time [s]	Average execution time [ms]
Simple	74.02 %	1.02	73.41 %	0.99	551.0	54.19
MobileNet V3	80.24 %	0.25	80.33 %	0.22	251.0	25.03
InceptionV3	76.41 %	0.68	76.18 %	0.73	451.0	39.69
VGG16	88.80 %	0.19	88.51 %	0.21	556.0	52.25

TABLE 3. Results obtained by late-fusion-based models.

cations carried out, bringing an intuitive notion of the model's performance and, in this case, reliability because it is a model trained from a balanced database.

(2.) F1-score: However, we also use The F1-score as a harmonic mean of precision and recall, which balances the trade-off between these two metrics. Precision measures the accuracy of positive predictions (i.e. how many predicted violent videos are actually violent), while recall measures the ability to identify all positive instances (i.e. how many actual violent videos are correctly identified). The F1-score is particularly useful in imbalanced datasets, as it provides a single metric that accounts for both false positives and false negatives. In our case, since the dataset is balanced, we also rely on accuracy as a primary metric. However, the F1-score offers a more robust evaluation of the model performance, especially in tasks where both false positives and false negatives are critical, as indicated by:

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4}$$

Using the K-fold technique for model validation, we present the accuracy and F1-average scores, along with their respective standard deviations, once we have completed all cross-validation iterations. We also present the temporal metrics, including the training time in seconds and the average execution time in milliseconds, to assess the computational performance of the models.

4. Results and discussion

This section shows the results obtained from the experiments performed according to the proposed strategy for video classification to identify instances of violence. The training and validation of all models were based on K-fold cross-validation, which divided the base into 10 test splits. For the models that used TL, only the output block had its neural weights adjusted, while the models that used a simple network had all their layers trained. These subsections showcase the validation phase results, categorising them into Late Fusion, Early Fusion, and Slow Fusion. These subsections evaluate and present the performance of the proposed models for each temporal fusion technique. Finally, they draw comparisons with related works by comparing the performance of the proposed models' to the state-of-the-art.

4.1. Late fusion

Table 3 presents the results obtained by the models based on the late fusion technique, in which the first and last frame of each video of the base are presented individually for the convolutional blocks listed in the table, followed by the output block that contains the LSTM and feedforward layers.

The results proved to be reliable due to the fact that the accuracy and F1-score showed similar values, indicating that the training of the models was conducted properly.

Convolutional block	Accuracy	Accuracy standard deviation	F1-score	Standard deviation of F1-score	Training time [s]	Average execution time [ms]
Simple	71.72 %	1.21	70.91 %	1.11	526.0	48.13
MobileNet V3	85.31 %	0.91	84.93 %	0.90	231.0	28.12
InceptionV3	72.25 %	0.15	71.41 %	0.21	480.0	45.60
VGG16	84.13 %	0.32	83.22 %	0.42	521.0	50.81

Table 4. Results obtained by early fusion-based models.

Convolutional block	Accuracy	Accuracy standard deviation	F1-score	Standard deviation of F1-score	Training time [s]	Average execution time [ms]
Simple	72.34 %	0.26	70.15 %	0.34	651.0	61.33
MobileNet V3	91.03 %	0.68	90.90 %	0.62	291.0	29.93
InceptionV3	76.67 %	0.22	76.01 %	0.22	501.0	49.85
VGG16	89.09 %	1.21	88.72 %	1.00	631.00	56.48

Table 5. Results obtained by slow fusion-based models.

For this approach, the VGG16 convolutional block model showed better results regarding accuracy and F1-score, surpassing the InceptionV3 convolutional block model by about 12 % and the simple convolutional block model by 14 %. The result greater than 88 % accuracy and F1-score, which is competitive with state-of-the-art models, still suggest possibilities for improving the result by the other Temporal Fusion techniques, since Late Fusion is the technique that least contributes to the extraction of temporal characteristics as only the initial and final frames of each of the examples are used.

The MobileNet V3 convolutional block model outperformed all others in terms of training and execution time, as predicted by its more simplified architecture. However, the metrics for the performance of the MobileNet-based model were lower than those of the VGG16-based model.

4.2. Early fusion

Table 4 illustrates the performance of the models using the convolutional blocks from the table and the early fusion method, which combines 10 frames of each video. With the exception of the MobileNet V3 convolutional block model, all other models showed a performance loss when compared to their counterparts using the late fusion technique. In general, this means that joining data sets together before extracting spatial features might make the convolutional block lose some temporal features, which would make the LSTM layer task harder and, in turn, make the models less effective.

As in the late fusion approach, the MobileNet-based model was more efficient in the temporal evaluation, thus being the model with the best performance for both evaluations: performance (evaluated by accuracy and F1-score) and computational performance (evaluated by the training and execution times of the models).

4.3. Slow fusion

The slow fusion technique, which used 10 frames of each video as input for 10 identical convolution blocks, formed the basis of the final models tested. Table 5 displays the results of testing four different compositions for the convolution blocks based on the methods used.

It is observed that all models with convolutional blocks in which TL was used obtained the best performance for this approach, which was expected since slow fusion presents the highest amount of extraction of individual spatial characteristics for the analysed frames, which contributes to the best performance of the LSTM layer in the task of extracting temporal characteristics in its input data from the convolutional blocks. This is why the model that did the best overall in the study used the slow fusion method. That model was the one with the MobileNet V3 convolutional block, which achieved an accuracy of 91.03 % and F1-score of 90.90 %, as shown in Figures 9 and 10. In addition, the MobileNet-based model remained the model with the shortest training and execution time, as it is a lighter model than the others, as shown in Figure 11.

4.4. Comparison with related works

In [18], the authors, in addition to organising the RLVS database, propose several models for the task of classifying videos to detect scenes of violence. A related work that presents a solution for the same task and, preferably, uses the same database should serve as the most appropriate comparison for the presented work. In this case, the use of the same database for the comparison of results is even more important

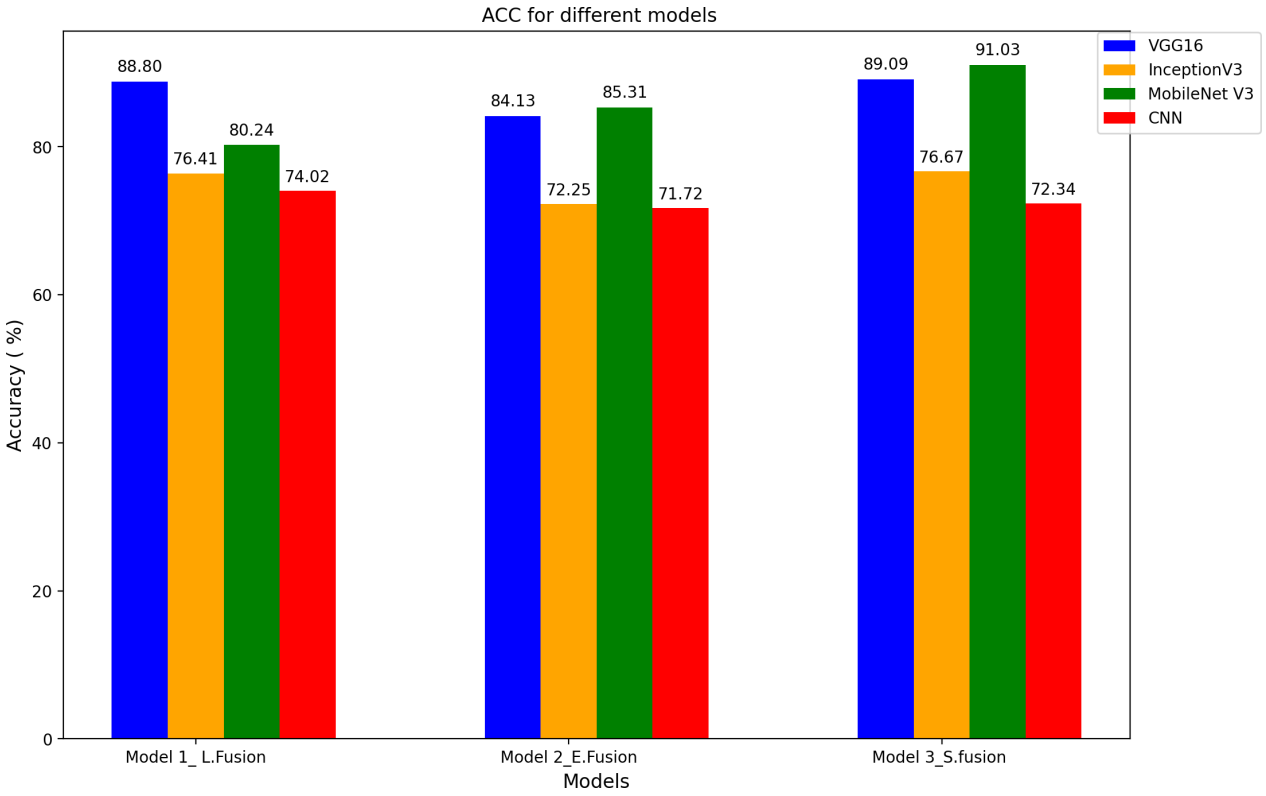


FIGURE 9. Accuracy scores of VD models for the RLVS dataset.

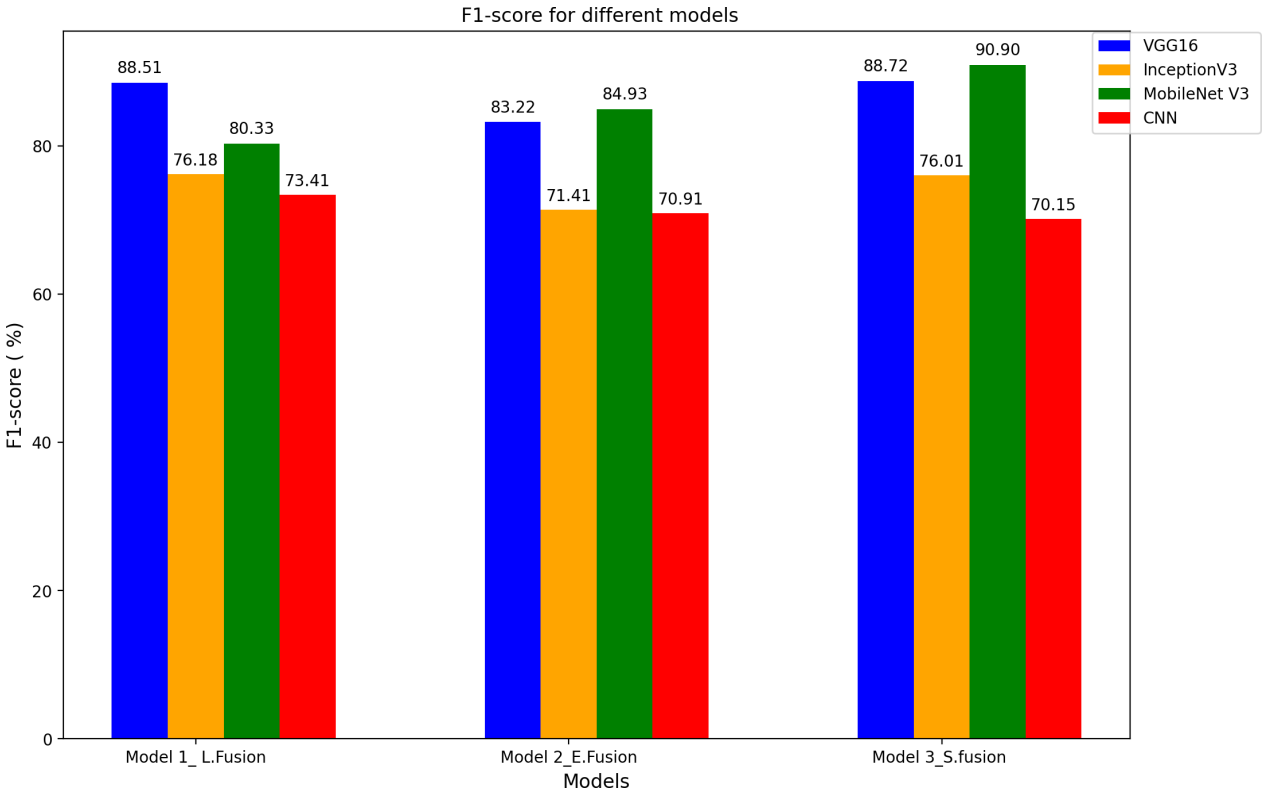
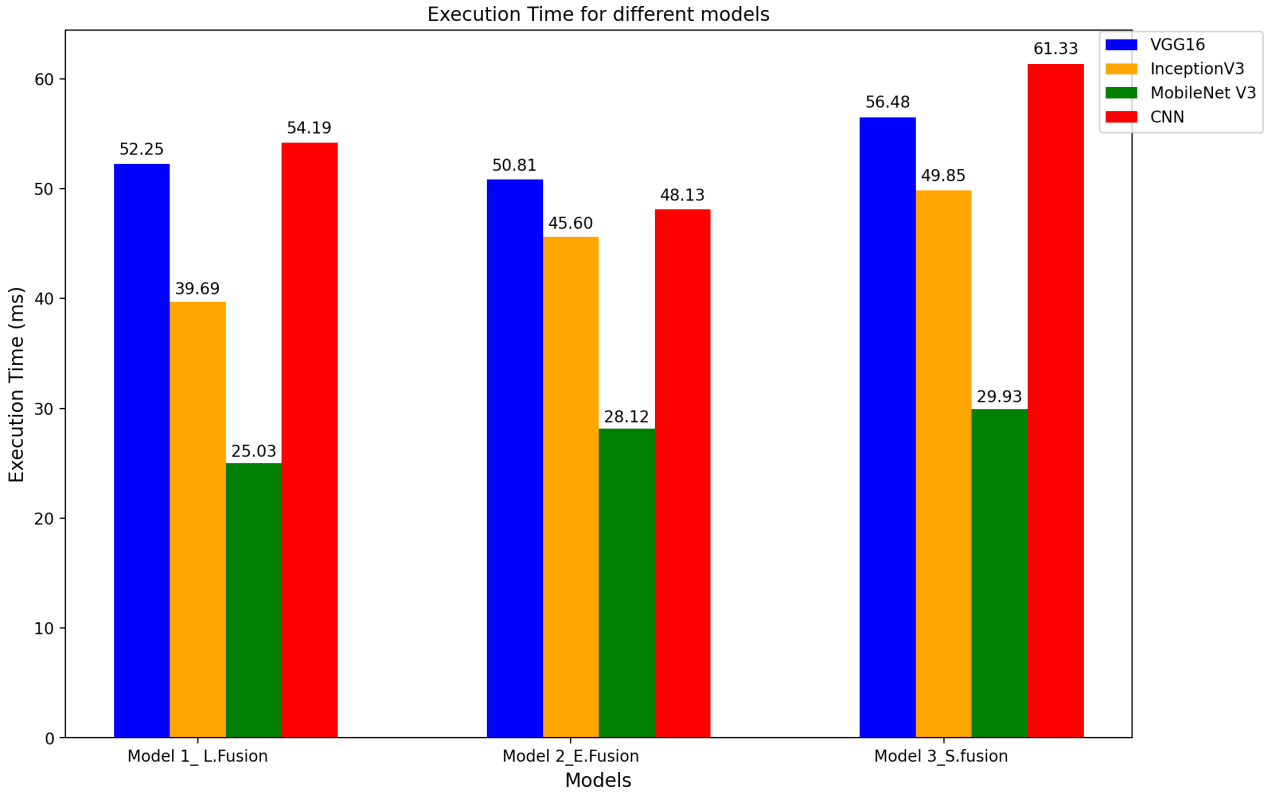
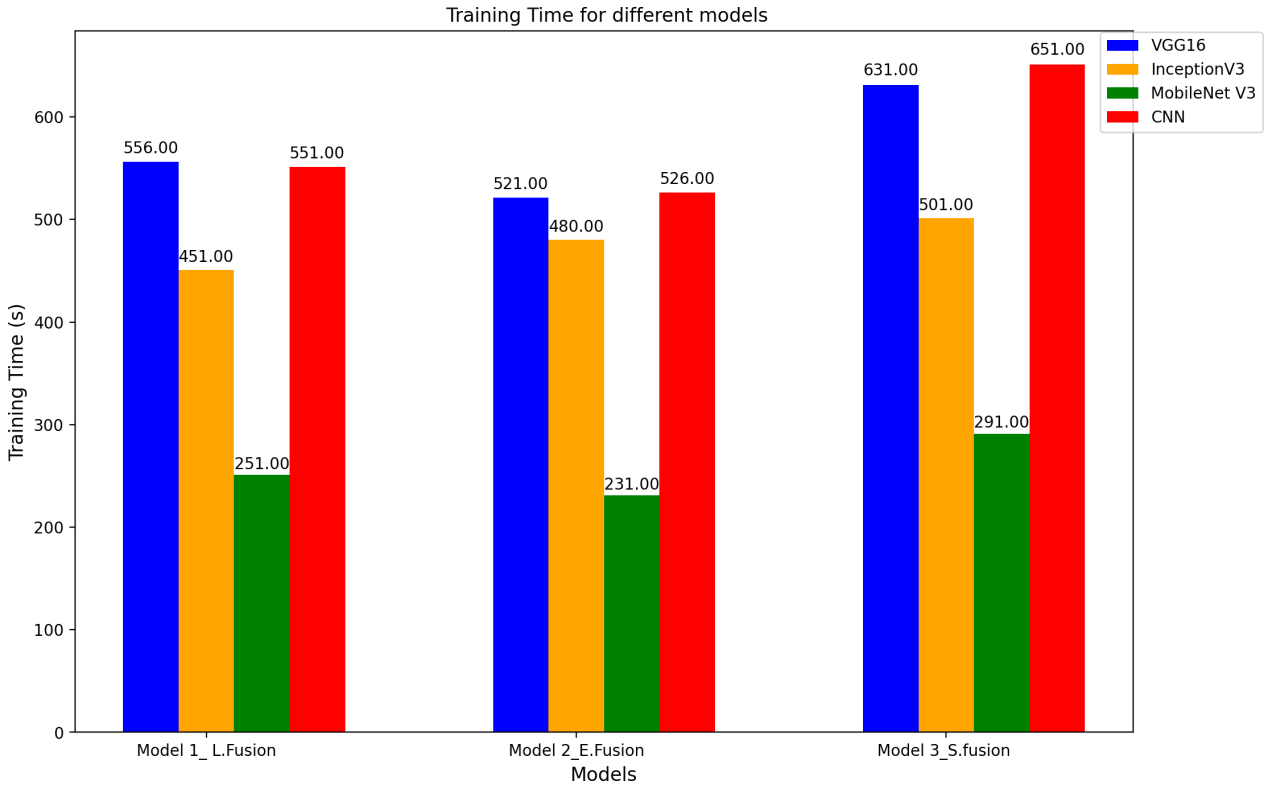


FIGURE 10. F1-score of VD models for the RLVS dataset.



(A). Execution time.



(B). Training time.

FIGURE 11. Computational time performance analysis for each model.

Models	Accuracy
VGG16-LSTM [18]	88.20 %
Late Fusion – MobileNet V3-LSTM (our)	88.80 %
Slow Fusion – MobileNet V3-LSTM (our)*	91.03 %

TABLE 6. Comparison of the results obtained with those found in [18].

Methods	Year	Deep Learning Features	Dataset	Accuracy
Soliman et al. [18]	2019	VGG16-LSTM	RLVS	88.20 %
Traor'e and Akhloufi [21]	2020	2D CNN-BiGRU	RLVS	90.25 %
Bertasius et al. [22]	2021	TimeSformer	RLVS	79.00 %
AlDahoul et al. [23]	2021	CNN-LSTM-IOT	RLVS	73.35 %
Romas et al. [24]	2022	U-Net – MobileNet V2 – LSTM	RLVS	82.03 %
Proposed method	2024	Slow Fusion – MobileNet V3 – LSTM	RLVS	91.03 %

TABLE 7. The Accuracy score of our proposed model along with the state-of-the-art methods for RLVS datasets.

because the main databases used in other articles, including references cited in Section 2, are studies referring to a considerably smaller database, such as The Hockey Dataset [16] with 1000 images; Action Movie Dataset [16] with 200 images; and Violent-Flow Dataset [17] with 246 images. The use of these small databases, or with relatively similar images (as in the case of the Hockey Fight dataset), can imply biased results. Conversely, [18] proposes an RLVS base that incorporates images from these bases along with new examples. Thus, the most appropriate RLVS base is considered a benchmark for the task in question.

Therefore, Table 6 presents the best results of [18] and this work, which proposes models based on Deep Learning for the task in question. The model that represents this work is based on the Slow Fusion approach with MobileNet V3 convolutional blocks, as it presented the best results regarding the analysed metrics. The model with better metrics presented in [18], in turn, for the RLVS database was based on the VGG16 architecture using fine-tuning techniques followed by LSTM layers.

Table 7 shows the accuracy-based performance increase between the state-of-the-art works for the task in question and the proposed model in this work. The proposed model outperformed previous works in the same field, demonstrating its superior performance. This proves that temporal fusion methods, especially slow fusion, work for tasks such as data representation techniques for video analysis, as they make the results of the models shown for the short video violence detection task more competitive.

Many related works, such as [8–10, 13], present solutions for the same task as this work with performance close to 100 %; however, using the Action Movies database [16], which presents movie scenes, also divided into two classes ("Non-Violence" and "Violence" behaviors), there are only 100 examples for each class. This lack of data generates difficulty in generalising the models, as can be seen in [18] when training and validating a model from the Action Movies dataset base and then validating with the RLVS base. The Action Movies dataset validation yielded an accuracy of approximately 0.99, whereas the RLVS base validation yielded an accuracy of 0.75. Therefore, we chose the RLVS database to conduct the presented experiments because it contains examples from the Action Movies dataset, making it more suitable for carrying out the task.

After evaluating the proposed approach, we proceeded to develop an application with the capability to identify instances of violent behaviour in videos. Initially, the user chooses the specific video to identify the violent behavior. The video processing program divides the video into individual frames and evaluates each frame to determine if it depicts violent actions. We present the results in a video format to facilitate user observation. Figure 12, highlighted in red, shows the detection of a violent behaviour, whereas Figure 13, highlighted in green, shows no violent behaviour observed.

Based on empirical evidence, we have discovered that violence analysis is now operating at a high level, but it cannot accurately determine the severity of violent behaviour. Furthermore, if the films are excessively long, the analysis process can be time-consuming. To overcome these challenges, our forthcoming research endeavors will focus on refining existing methodologies and developing innovative techniques to enhance both the precision of violence severity assessment and the efficiency of analysing lengthy film content.

5. Conclusions and future work

The work presented the testing and validation of several deep learning models, based on ConvLSTM, for the task of detecting violence in videos. We used different temporal fusion methods to separate the models. The Slow Fusion (SF) method, which used the MobileNet V3 convolutional block, showed the

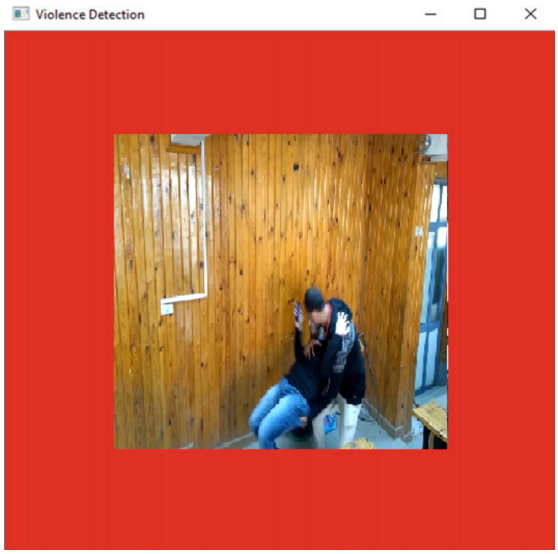


Figure 12. A violent video is depicted in the application.

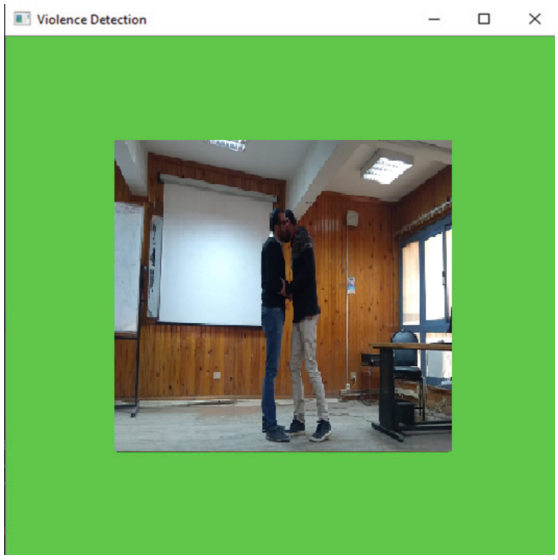


Figure 13. A non-violent video is depicted in the application.

best results, with 91.03 % accuracy and a 90.90 % F1-score. This result is superior to state-of-the-art models that used the RLVS database, also used in this work, for training and validation. In addition to better performance in classification, the result of using the MobileNet V3 model with slow fusion makes the solution more viable for platforms with lower computational power, as it is a lightweight model, which is an advantage for surveillance applications. The fact that the MobileNet V3 convolutional block models have short training and execution times compared to other models in different temporal fusion tests demonstrates their efficiency.

Future work will focus on how to enhance the applicability and robustness of the proposed model. First, we plan to test the model on diverse datasets, such as HMDB51, UCF101, Kinetics, and other relevant databases, such as Surveillance Fight Detection (SFD), and RWF-2000. This will evaluate its generalisability across different contexts, including surveillance, sports, and crowded public events. Second, we aim to extend the classification task to include multiple categories of violent behaviors (e.g., fighting, shooting, aggressive gestures), which will improve the model's practical utility in real-world applications. In addition, we will address the current limitations, including computational complexity, potential bias from the exclusive use of the RLVS dataset, and the inability to assess the severity of violent actions. These improvements will involve optimizing the model for real-time processing, incorporating additional datasets to reduce bias, and developing methods to evaluate the severity of detected violence. Finally, we will explore advanced techniques, such as 3D Convolutional Neural Networks (3DCNN) and frequency-domain representations, to further improve the model's performance in extracting spatial and temporal features.

Acknowledgements

This work is supported by a research project on the design and implementation of a surveillance system based on biometric systems for the detection and recognition of individuals and abnormal behaviors No. A25N01UN080120180002.

References

[1] J. S. Gracias, G. S. Parnell, E. Specking, et al. Smart cities – A structured literature review. *Smart Cities* **6**(4):1719–1743, 2023. https://doi.org/10.3390/smartcities6040080

[2] I. A. T. Hashem, V. Chang, N. B. Anuar, et al. The role of big data in smart city. *International Journal of Information Management* **36**(5):748–758, 2016. https://doi.org/10.1016/j.ijinfomgt.2016.05.002

[3] D. M. Blei, P. Smyth. Science and data science. *Proceedings of the National Academy of Sciences of the United States of America* **114**(33):8689–8692, 2017. https://doi.org/10.1073/pnas.1702076114

[4] F. A. Temel, O. C. Yolcu, N. G. Turan. Artificial intelligence and machine learning approaches in composting process: A review. *Bioresource Technology* **370**:128539, 2023. https://doi.org/10.1016/j.biortech.2022.128539

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

[6] L. Calderoni, D. Maio, S. Rovis. Deploying a network of smart cameras for traffic monitoring on a "city kernel". *Expert Systems with Applications* **41**(2):502–507, 2014. https://doi.org/10.1016/j.eswa.2013.07.076

[7] K. Muhammad, J. Ahmad, I. Mehmood, et al. Convolutional neural networks based fire detection in surveillance videos. *IEEE Access* **6**:18174–18183, 2018. https://doi.org/10.1109/access.2018.2812835

[8] S. Sudhakaran, O. Lanz. Learning to detect violent videos using convolutional long short-term memory. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6. IEEE, 2017. https://doi.org/10.1109/avss.2017.8078468

[9] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, G. Bueno. Fight recognition in video using hough forests and 2D convolutional neural network. *IEEE Transactions on Image Processing* **27**(10):4787–4797, 2018. https://doi.org/10.1109/tip.2018.2845742

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

[11] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience* **2018**(1):7068349, 2018. https://doi.org/10.1155/2018/7068349

[12] A. B. Sargano, X. Wang, P. Angelov, Z. Habib. Human action recognition using transfer learning with deep representations. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 463–469. IEEE, 2017. https://doi.org/10.1109/ijcnn.2017.7965890

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

[14] J. Deng, W. Dong, R. Socher, et al. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009. https://doi.org/10.1109/cvprw.2009.5206848

[15] T. Guo, Z. Xu, X. Yao, et al. Robust online time series prediction with recurrent neural networks. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 816–825. IEEE, 2016. https://doi.org/10.1109/dsaa.2016.92

[16] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, R. Sukthankar. Violence detection in video using computer vision techniques. In *Computer Analysis of Images and Patterns*, pp. 332–339. Springer, 2011. https://doi.org/10.1007/978-3-642-23678-5_39

[17] T. Hassner, Y. Itcher, 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. IEEE, 2012. https://doi.org/10.1109/cvprw.2012.6239348

[18] M. M. Soliman, M. H. Kamal, M. A. E.-M. Nashed, et al. Violence recognition from videos using deep learning techniques. In *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 80–85. IEEE, 2019. https://doi.org/10.1109/icicis46948.2019.9014714

[19] M. Elesawy, M. Hussein, M. A. E. Massih. Real life violence situations dataset, 2019. [2024-02-13]. https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset/

[20] A. Karpathy, G. Toderici, S. Shetty, et al. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732. 2014. https://doi.org/10.1109/cvpr.2014.223

[21] A. Traoré, M. A. Akhloufi. 2D bidirectional gated recurrent unit convolutional neural networks for end-to-end violence detection in videos. In *International Conference on Image Analysis and Recognition*, pp. 152–160. Springer, 2020. https://doi.org/10.1007/978-3-030-50347-5_14

[22] G. Bertasius, H. Wang, L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 813–824. 2021. https://doi.org/10.48550/arXiv.2102.05095

[23] N. AlDahoul, H. A. Karim, R. Datta, et al. Convolutional neural network – long short term memory based IOT node for violence detection. In *2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, pp. 1–6. IEEE, 2021. https://doi.org/10.1109/iicaiet51634.2021.9573691

[24] R. Vijeikis, V. Raudonis, G. Dervinis. Efficient violence detection in surveillance. *Sensors* **22**(6):2216, 2022. https://doi.org/10.3390/s22062216