Combining a mobile deep neural network and a recurrent layer for violence detection in videos

Paolo Contardo[1,2], Selene Tomassini[1], Nicola Falcionelli[1], Aldo Franco Dragoni[1] and Paolo Sernani[3,*]

[1]Information Engineering Department, Università Politecnica delle Marche, Via Brecce Bianche 12, 60131 Ancona, Italy
[2]Gabinetto Interregionale di Polizia Scientifica per le Marche e l'Abruzzo, Via Gervasoni 19, 60129 Ancona, Italy
[3]Department of Law, University of Macerata, Piaggia dell'Università 2, 62100 Macerata, Italy

Abstract

Several techniques for the automatic detection of violent scenes in videos and security footage appeared in recent years, for example with the goal of unburdening authorities from the need of analyzing hours of Closed-Circuit TeleVision (CCTV) clips. In this regard, Deep Learning-based techniques such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) emerged as effective for violence detection. Nevertheless, most of such techniques require significant computational and memory resources to run the automatic detection of violence. Thus, we propose the combination of an established CNN, MobileNetV2, designed for the use in mobile and embedded devices with a recurrent layer to extract the spatio-temporal features in the security videos. A lightweight model can run in embedded devices, in a edge computing fashion, for example to allow processing the videos near the camera recording them, to preserve privacy. Specifically, we exploit transfer learning, as we use a pre-trained version of MobileNetV2, and we propose two different models combining it with a Bidirectional Long Short-Term Memory (Bi-LSTM) and a Convolutional LSTM (ConvLSTM). The paper presents accuracy tests of the two models on the AIRTLab dataset and a comparison with more complex models developed in our previous work, in order to evaluate the drop of accuracy necessary to use a model compatible with limited resources. The network composed of MobileNetV2 and the ConvLSTM scores a 94.1% accuracy, against the 96.1% of a model based on a more complex 3D CNN.

Keywords

Violence Detection, Convolutional Neural Network, Long Short-Term Memory, Action Recognition, MobileNetV2, Law Enforcement, Crime Investigation, Deep Learning

1. Introduction

Closed-Circuit TeleVision (CCTV) emerged as one of the mainstream crime prevention techniques [1], providing abundant and precise information for security and law enforcement applications [2, 3]. In fact, Artificial Intelligence (AI) methodologies, especially those based on Deep Learning, are demonstrating their effectiveness in applications that take advantages of CCTV footage, such as weapon detection [4, 5], face recognition [6, 7], and accident detection [8]. With the goal of unburdening authorities from the need of manually analyzing hours of CCTV videos and allowing them to take decisions in short time [9], many techniques to automatically detect violence in videos emerged in the scientific literature. In this regard, the first studies focused on the use of flow descriptors and hand-crafted features (see, for example, [10, 11]). However, Deep Learning-based techniques demonstrated better accuracy in violence detection, proposing to use Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) for such task [12]. These techniques are capable of modeling the spatio-temporal information included in the CCTV footage, i.e., features that represent the motion information contained in a sequence of frames, in addition to the spatial information contained in a single frame.

In our previous work [13], we tested 13 different Deep Neural Networks (DNNs) for the task of violence detection in videos. Specifically, we compared a pre-trained 3D CNN, C3D [14], combined with a Support Vector Machine (SVM) classifier, with C3D combined with fully connected layers, with a trained-from-scratch Convolutional Long Short-Term Memory (ConvLSTM) [15] plus fully connected layers, with other ten networks based on time distributed pre-trained 2D CNNs combined with Bidirectional LSTM (Bi-LSTM) [16] (5 networks) and ConvLSTM (5 networks). The C3D-based models got the best accuracy results in detecting violence on different datasets, taking advantage of the 3D architecture capable of mod-

RTA-CSIT 2023: 5th International Conference Recent Trends and Applications In Computer Science And Information Technology, April 26–27, 2023, Tirana, Albania
*Corresponding author.
p.contardo@pm.univpm.it (P. Contardo); s.tomassini@pm.univpm.it (S. Tomassini); n.falcionelli@staff.univpm.it (N. Falcionelli); a.f.dragoni@staff.univpm.it (A. F. Dragoni); paolo.sernani@unimc.it (P. Sernani)
0000-0002-5605-4783 (P. Contardo); 0000-0002-1087-7004 (S. Tomassini); 0000-0002-1312-6310 (N. Falcionelli); 0000-0002-3013-3424 (A. F. Dragoni); 0000-0001-7614-7154 (P. Sernani)

eling the spatio-temporal features of the videos as well as of the transfer learning. Nevertheless, 3D CNNs require computational and storage resources which are usually not compatible with mobile and embedded devices [17] i.e., for edge computing.

To tackle such issue, in this paper we propose two models based on the combination of a CNN specifically designed for mobile devices i.e., MobileNetV2 [18], with a recurrent layer to extract the temporal information and fully connected layers for the classification of the videos into violent or not. Specifically, in one model we used the Bi-LSTM as the recurrent layers, whereas in the other we used the ConvLSTM. To understand its effectiveness and evaluate any potential drop of accuracy, we test the proposed networks on the AIRTLab dataset [19], comparing the results with those obtained in our previous work. As such, this paper contributes to the state of the art in violence detection with:

- The proposal of using MobileNetV2, pre-trained on the Imagenet dataset [20], by time distributing it over the frames of the security videos to be classified into violent or not, in combination with a recurrent module to model the temporal information in addition to the spatial information of the videos.
- The comparison of the proposed networks with our previous tested models [13] to evaluate the drop of accuracy necessary to use a network tailored for mobile and embedded devices i.e., MobileNetV2.

The rest of this paper is organized as follows. Section 2 provides a literature review about Deep Learning techniques applied in the violence detection task. Section 3 describes the proposed networks, providing the necessary background and detailing the structure of the used dataset. Section 4 discusses the experimental evaluation and presents the main findings. Finally, Section 5 draws the conclusions of this study.

2. Related Works

Several violence detection techniques based on Deep Neural Networks and, specifically, Recurrent Neural Networks (such as LSTM, Bi-LSTM, ConvLSTM) and CNNs demonstrated their effectiveness [12]. For example, Sudhakaran and Lanz. [21] combined the spatial features computed by 2D CNNs on the frames of the videos, with a ConvLSTM, to extract the temporal features as well. They got 94.5% accuracy on the Crowd Violence dataset [10] and 97.1% on the Hockey Fight dataset [22]. Li et al. [23] proposed a 3D CNN composed of 10 layers, adding dense and transitional layers after the convolutional layers. They got 97.2% accuracy on the Crowd Violence dataset, and 98.3% accuracy on the Hockey Fight dataset. Accattoli et al. [24] and Ullah et al. [25] also based their work on a 3D CNN, but, instead of training it from scratch, they applied transfer learning. Accattoli et al. added a SVM to the CNN, getting 99.2% accuracy on the Hockey Fight and 98.5% accuracy on the Crowd Violence. Instead, Ullah et al. implemented a end-to-end neural network by adding fully connected layers to the 3D CNN, getting 98% accuracy on the Crowd Violence and 96% accuracy on the Hockey Fight. Sernani et al. [13] compared 13 different Deep Neural Networks on the Hockey Fight, Crowd Violence and AIRTLab datasets. Specifically, they tested a pre-trained 3D CNN (C3D) combined with a SVM, C3D with fully connected layers, a ConvLSTM combined with fully connected layers, 5 time-distributed pre-trained 2D CNNs combined with the Bi-LSTM and the same 2D CNNs combined with a ConvLSTM. They got the best results with the two C3D-based networks, with 96.1% accuracy on the AIRTLab dataset, 97.86% accuracy on the Hockey Fight, and 99.6% accuracy on the Crowd Violence. Freire-Obregón et al. [26] used an Inflated 3D ConvNet to extract the spatio-temporal features on the output of two person trackers to perform context-free violence detection, i.e., the violence detection applied to the subjects in the videos only, discarding any background or context information. They combined such feature extractor with different classifiers, getting the best results with the Linear Regression, with 99.45% accuracy on the Crowd Violence dataset, 99.43% on the Hockey Fight, and 97.54% on the AIRTLab.

Whereas the aforementioned techniques demonstrated effective in the task of automatically detecting violence in different video databases, they are all high demanding for computational and storage resources, making them inadequate to run in mobile and embedded devices i.e., for edge computing. In our previous work [13], we demonstrated that pre-trained 2D CNNs, time distributed on the frames of the security videos and combined with Bi-LSTM, achieve a lower accuracy than 3D CNNs. For example, VGG16 [27] combined with a Bi-LSTM, achieved 94.92% accuracy on the AIRTLab dataset, 95.47% on the Hockey Fight, and 97.39% on the Crowd Violence. Nevertheless, such accuracy in detecting violence might be still acceptable, to get a compromise to run violence detection at the edge to avoid data transmission and preserve privacy. Therefore, given such results and the need for models capable of running violence detection at the edge, differently from the listed works, we propose to "time-distribute" MobileNetV2 [18], a 2D CNN specifically designed for mobile devices, on the frames of the security videos. We combine it with a recurrent layer and fully connected layers to perform the violence classification and test two different versions, one based on the Bi-LSTM and one on the ConvLSTM.

In addition to the search for the best accuracy, the sci-

entific literature concerning the use of Deep Learning techniques for the automatic detection of violent scenes includes other studies. For example, Ciampi et al. [28] tested some of the aforementioned techniques, such as 3D CNNs and ConvLSTM, on a novel dataset, the Bus Violence, to study the behavior of the violence detection methodologies based on Deep Learning when the background and context information significantly varies. Silva et al. [29] proposed the use of a federated learning approach to distribute the learning process across different devices, preserving privacy, with a server combining the locally trained model into a global model. However, instead on relying on videos or on video portions, the applied 2D CNNs to single frames, achieving the best results with MobileNet (99.4% accuracy on the AIRTLab dataset). Yang et al. [30] proposed a multimodal approach (Multimodal Contrastive Learning – MCL) to use both video and audio for the automatic detection of violence. They got 84.03% average precision on the XD-Violence dataset [31], against the 83.19% of using the video only and the 76.07% of using the audio only.

3. Materials and Methods

As explained in Sections 1 and 2, many studies about the use of Deep Neural Networks for the violence detection in videos proposed complex architectures, such as 3D CNNs, requiring computational and memory resources that are usually not compatible with mobile and embedded devices. To this end, we propose the use of MobileNetV2, time-distributed over 16-frames chunks of the videos, combined with a recurrent layer to model the temporal information of the sequence of frames, in addition to the spatial information. In the following, we provide some background about MobileNetV2, the LSTM architecture, and the ConvLSTM architecture (3.1). Then, we present the proposed neural networks (3.2) and describe the dataset used for the tests (3.3).

3.1. Background: MobileNetV2, LSTM, and ConvLSTM

In the original definition of LeCun and Bengio [32], a unit of a layer in a CNN receives inputs from a set of units in the local receptive field, via a convolution operation with kernels composed of shared weights. In MobileNetV2 [18] this concept is extended to cope with the limited computational resources of mobile and embedded devices. Instead of the traditional convolution operation of CNNs, MobileNetV2 decomposes convolutional layers into two separate layers:

- The depthwise convolution layer that applies a separate filter to each input channel.
- The pointwise convolution layer applied to the output of the depthwise convolution layer using a 1x1 convolution.

In addition, in MobileNetV2, linear bottlenecks and residual connections follow the convolution. Specifically, linear bottlenecks use a linear activation function instead of a non-linear activation function, reducing the computational cost of the network.

As a traditional CNN, MobileNetV2 models the spatial information of images i.e., the frames of the videos. Therefore, we added a recurrent layer to the output of MobileNetV2 to model the temporal information available in the videos, using a Bi-LSTM and a ConvLSTM. In the original LSTM architecture [33], a hidden unit is composed by a self-recurrent cell, called memory cell, whose input/output is regulated by three multiplicative gates i.e., the input gate, the output gate, and the forget gate [34]. Specifically, the output h_t at time point t of a LSTM hidden unit is given by the following equations [34]:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$

$$h_t = o_t \tanh(c_t) \quad (5)$$

where i_t, f_t, o_t, and c_t are the activation vectors of the input gate, forget gate, output gate, and memory cell at time point t, σ is the sigmoid function, b denotes the bias of each gate/cell, and W are diagonal weight matrices.

In the original formulation, a LSTM processes input data in ascending temporal order. However, the recognition of a pattern might be more effective with the use of future context as well. To this end, Bidirectional RNNs [35] and, specifically, Bidirectional LSTMs [16] have been proposed. The basic idea of such models is to present the training sequences both forwards and backwards, using two separate recurrent nets, which are connected to the same output layer. As such, we based one of our models on the Bi-LSTM, as the videos are processed once recorded, taking advantage of both previous and future context.

For the ConvLSTM, we use the formulation of Shi et al. [15], who extended the LSTM architecture by adding convolutional structures to state transition. As Shi et al. explained, the LSTM architecture is adequate to extract temporal features, but includes too much redundancy for spatial features. In this regard, they proposed to add convolutional structures in the transitions between the input gate and the memory cell, and in the self-recurrency of the memory cell, regulated by the forget gate. Therefore, in a ConvLSTM, the output of a hidden unit is regulated

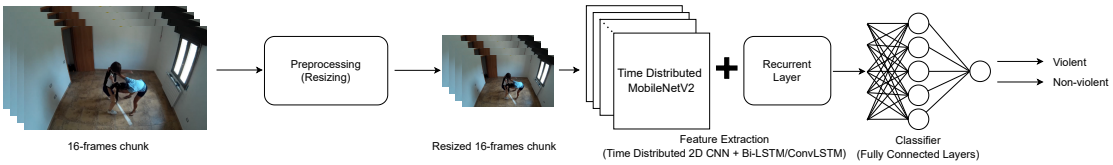


Figure 1: The schematic of the proposed models. They process sequences composed of 16 frames (16-frames chunks) resized to 224 x 224 pixels. To apply MobileNetV2 to the videos (i.e., a 3D input), given that it is a 2D CNN, the network is time distributed over the 16 frames of the security video chunks used in this study. To extract the temporal features of the videos in addition to the spatial features extracted by MobileNetV2, the time-distributed CNN is followed by a recurrent layer (a Bi-LSTM or a ConvLSTM). Finally, the fully connected layers perform the classification of the videos into violent or not.

by the following equations:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} c_{t-1} + b_i) \quad (6)$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} c_{t-1} + b_f) \quad (7)$$

$$x_t = f_t c_{t-1} + i_t \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (8)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} c_t + b_o) \quad (9)$$

$$h_t = o_t \tanh(c_t) \quad (10)$$

where the activations of input gate, forget gate, output gate, and memory cell (i_t, f_t, o_t, and c_t), as well as input and output (x_t, h_t) are 3D tensors. As such, we used the ConvLSTM in the second of our proposed models.

3.2. Proposed Classification Architecture

As depicted in the schematic in Figure 1, to classify the videos into violent or not, we propose two Deep Learning-based classifiers based on MobileNetV2, pre-trained on the Imagenet dataset [20], followed by a recurrent layer and fully connected layer. The weights of MobileNetV2 are freezed on the Imagenet training. Instead, the Bi-LSTM layer or the ConvLSTM layer and the fully connected layers are trained from scratch on the AIRTLab dataset, as explained in Section 4 (Subsection 4.1). Given that in our previous work we run the classification over 16-frames chunks of the videos, in this work we use the same videos split into 16 frames chunks, in order to allow a fair comparison between the classifiers. The video of the AIRTLab dataset are resized at 224 x 224 pixels, as this is the input shape in the original MobileNetV2 implementation.

Table 1 includes the layers composing the first proposed model. MobileNetV2, with its 2,257,984 freezed weights, is time distributed over the 16 frames used as the input. The Bi-LSTM is composed of 128 hidden units, followed by a 0.5 dropout to limit the overfitting, a fully connected layer with 128 ReLU neurons, another 0.5 dropout and a fully connected sigmoid neuron for the final classification.

Table 1
The first proposed classification model. The Bi-LSTM and two fully connected layers were added to MobileNetV2, pre-trained on ImageNet. MobileNetV2 was time distributed in order to be applied to a 3D input i.e., the clips of the AIRTLab dataset.

Layer	Architecture	Output Shape	Params #
Time Distr. MobileNetV2	-	(16, 7, 7, 1280)	2257984
Time Distr. Flatten	-	(16, 62720)	0
Bi-LSTM	128 units	(256)	64357376
Dropout	0.5 rate	(256)	0
Dense	128 units, ReLU	(128)	32896
Dropout	0.5 rate	(128)	0
Dense	1 units, Sigmoid	(1)	129

Table 2
The second proposed classification model. The ConvLSTM and two fully connected layers were added to MobileNetV2, pre-trained on ImageNet. MobileNetV2 was time distributed in order to be applied to a 3D input i.e., the clips of the AIRTLab dataset.

Layer	Architecture	Output Shape	Params #
Time Distr. MobileNetV2	-	(16, 7, 7, 1280)	2257984
ConvLSTM	64 3x3 filters, tanh	(5, 5, 64)	3096832
Flatten	-	(1600)	0
Dropout	0.5 rate	(1600)	0
Dense	256 units, ReLu	(256)	409856
Dropout	0.5 rate	(256)	0
Dense	1 unit, Sigmoid	(1)	257

Table 2 lists the layers composing the second proposed model. A ConvLSTM composed of 64 3 $\times$ 3 filters with the tanh activation function follows the time-distributed MobileNetV2. The network is completed by a 0.5 dropout, a fully connected layer with 256 ReLU neurons, another 0.5 dropout and a fully connected sigmoid neuron to perform the final classification into violent or not.

The Bi-LSTM-based model has a total of 66,648,385 parameters. The weights of MobileNetV2 are freezed, which means that the total number of trainable parameters is 64,390,401 (corresponding to the 128 hidden units of the Bi-LSTM, the 128 ReLU neurons of the first fully connected layer, and the sigmoid neuron of the last layer). Instead, in the ConvLSTM-based model there are 5,764,929 parameters (3,506,945 are trainable, corresponding to the

64 filters of the ConvLSTM layer, the 256 ReLU neurons of the first fully connected layer, and the final sigmoid neuron for the classification). Therefore, the model based on the ConvLSTM requires less memory than the model based on the Bi-LSTM, being more adequate for the use in mobile and embedded devices.

3.3. Used Dataset

To test the performance of the proposed classifiers and compare them to our previous work, we run accuracy tests on the AIRTLab dataset. It contains 350 videos (MP4 files with H.264 codec, mean length of 5.63 seconds). The frame rate is 30 fps and the frame resolution is 1920 x 1080 pixels. The dataset includes 230 violent videos and 120 non-violent videos. The 230 violent videos represent 115 violent actions recorded from two different cameras placed into two different spots. Similarly, the 120 non-violent videos represent 60 non-violent actions, recorded from two different cameras placed into two different spots. All the videos were taken inside the same room. One camera was placed in the top left corner in front of the room door. The second camera was in the top right corner on the door side.

A group of non-professional actors played the violent and non-violent actions. The number of actors varied from 2 to 4 per video. In the violent videos, the actors simulated actions frequent in scuffles, such as punches, kicks, beating with canes, slapping, gun shots, and stabbing. In the non-violent videos, the actors simulated actions which can result in false positives due to the similarity with violent actions (for example for the presence of fast movements). Specifically, the non-violent videos contains actions such as exulting, hugging, gesticulating, and clapping and giving high fives.

4. Results and Discussion

We tested the two proposed models with the same protocol used in our previous work [13] i.e., by measuring the classification results over the AIRTLab dataset. The objective is to compare the accuracy performance of the classifiers based on a 2D CNN designed for mobile and embedded devices with those of classifiers requiring more resources. Therefore, in the following subsections, we describe the experimental protocol (4.1), discuss the results (4.2), and present the limitations of our evaluation (4.3).

4.1. Experimental Protocol and Evaluation Metrics

Whereas MobileNetV2 was pre-trained on Imagenet and its weights were freezed, the Bi-LSTM and ConvLSTM layers, together with the fully connected layers, needed to be trained from scratch. Therefore, to run the training and test on the AIRTLab dataset, we applied a stratified shuffle split cross-validation scheme. To this end, we repeated a randomized 80-20 split 5 times, using the 80% of the data as the training set, and the 20% as the test set, preserving the percentage of samples from each class, in each split. The data splits were the same for both the proposed models and for the models of our previous work, to implement a fair comparison. Given that the inputs for the models are sequences composed of 16 frames and the videos in the dataset include a total of 3537 of such sequences, 2829 samples (i.e., 16-frames chunks) were used for training, and 708 for testing, in each split. The 12.5% of the training data i.e., the 10% of the entire dataset, was used as validation data.

Table 3
Number of training epochs in each split (S1-S5) of the stratified shuffle split cross validation scheme.

	S1	S2	S3	S4	S5	Mean
MobileNetV2 + Bi-LSTM	20	18	23	19	32	22.40 $\pm$ 5.68
MobileNetV2 + ConvLSTM	11	20	22	19	17	17.80 $\pm$ 4.21

Both the proposed models used the Binary Cross-Entropy loss function, minimized with the Adam optimizer. We early stopped the training after 5 epochs without any improvement on the minimum validation loss, restoring the weights corresponding to the best epoch. To this end, Table 3 lists the number of training epochs in each split of the stratified shuffle split cross validation scheme, for each model. The average number of training epochs was 22.4 ($\pm$ 5.68) for the model using the Bi-LSTM layer, and 17.8 ($\pm$ 4.21) for the model based on the ConvLSTM. The batch size was 8 for both neural networks.

The tests ran on Google Colab Pro with the GPU runtime (the GPU used for the tests was a Nvidia A100 SXM4 with 40 GB of RAM) and extended RAM (83.5 GB), using Keras 2.11.0, TensorFlow 2.11.0, and Scikit-learn 1.2.1.

Labeling as negative the 16-frames chunks of the non-violent videos and as positive the chunks of the violent videos, we computed the following metrics over the test set in each split of the stratified shuffle split cross validation scheme:

- Sensitivity (True Positive Rate – TPR) i.e., the portion of positives that are correctly identified (over all the available positives).
- Specificity (True Negative Rate – TNR) i.e., the portion of negatives that are correctly identified (over all the available negatives).
- Accuracy i.e., the portion of samples that are correctly identified (over all the available samples).
- F_1 score i.e., the armonic mean of precision (the

Table 4
The results of the model composed of MobileNetV2 and the Bi-LSTM, computed for each split of the stratified shuffle-split cross validation scheme.

	Split 1	Split 2	Split 3	Split 4	Split 5
Sensitivity	100.00%	90.97%	97.48%	95.59%	97.69%
Specificity	00.00%	87.93%	62.50%	76.72%	82.76%
Accuracy	67.23%	89.97%	86.02%	89.41%	92.80%
F_1 score	80.41%	92.42%	90.36%	92.39%	94.80%

Table 5
The results of the model composed of MobileNetV2 and the ConvLSTM, computed for each split of the stratified shuffle-split cross validation scheme.

	Split 1	Split 2	Split 3	Split 4	Split 5
Sensitivity	94.54%	95.80%	97.90%	94.96%	96.22%
Specificity	88.36%	92.24%	85.34%	93.10%	93.10%
Accuracy	92.51%	94.63%	93.79%	94.35%	95.20%
F_1 score	94.44%	96.00%	95.49%	95.76%	96.42%

ratio between the positives correctly identified and all the identified positives) and sensitivity.

These metrics can be formulated in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) according to the following equations:

$$sensitivity = \frac{TP}{TP + FN} \tag{11}$$

$$specificity = \frac{TN}{TN + FP} \tag{12}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$F_1\ score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{14}$$

Moreover, in each split, we computed the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC), showing the TPR against the False Positive Rate (FPR) when the classification threshold varies, to understand the diagnostic capability of each model.

4.2. Results

Table 4 lists the metrics obtained by the model composed of MobileNetV2 and the Bi-LSTM over the five splits of the cross-validation performed on the AIRTLab dataset. The metrics significantly vary across the splits, showing a poor generalization capability. For example, in split 1, all the 708 samples of the test set are labeled as violent, causing 232 false positives. As such, the sensitivity is 100% whereas the specificity is 0%. The split where most of the negatives are correctly identified is the number 2: here, 204 negatives out of 232 are correctly classified (specificity 87.93%). In the same split, 433 violent chunks out of 476 are correctly classified. As such, the accuracy is 92.8%.

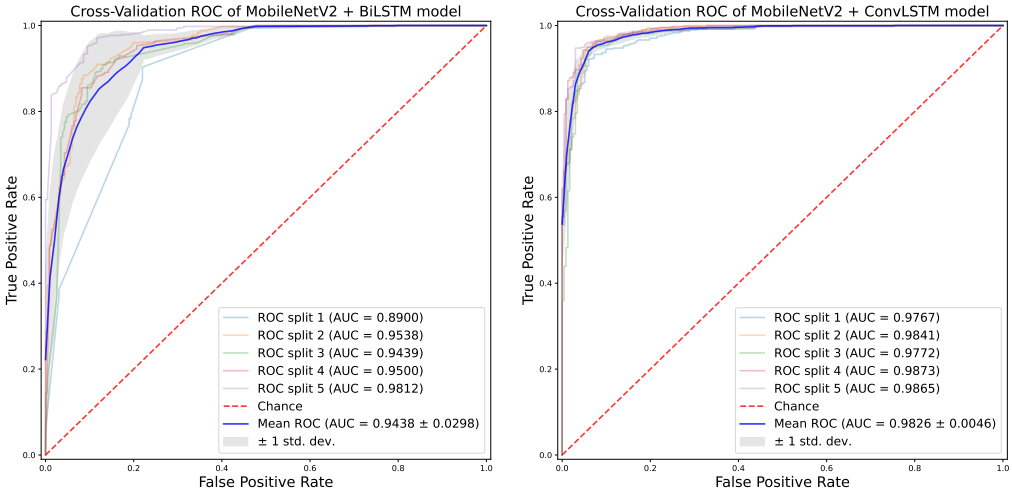
Instead, the model based on MobileNetV2 and the ConvLSTM exhibits a better generalization capability than the previous one, as showed in Table 5. The sensitivity is greater than 94% across all the splits, and the lowest specificity is in split 3 (85.34%). The best split is the number 5, where the F_1 score is 96.42%.

The difference in the generalization capability of the two proposed models is highlighted by the ROC curves in Figure 2. In fact, the model using the Bi-LSTM as the recurrent layer scores an average AUC equal to 94.38% ($\pm$ 2.98%), whereas the model using the ConvLSTM gets 98.26% ($\pm$ 0.46%). This behavior might be due to the different number of trainable parameters of the two models. In the Bi-LSTM-based model there are 64,390,401 trainable parameters. Instead, in the ConvLSTM-based model, the number of trainable parameters is 3,506,945. As such, the Bi-LSTM-based model might be oversized for the violence detection task on the AIRTLab dataset, struggling to converge to an acceptable classification performance. Therefore, the ConvLSTM-based model, that is the lightest in terms of required resources between the two proposed in this work, exhibits a better performance in terms of classification accuracy and generalization capability.

Table 6 compares the performance of the two models proposed in this paper with those based on C3D tested in our previous work. Even if lighter in terms of required computational resources, the model based on MobileNetV2 and the ConvLSTM gets an average AUC of 98%, against the 99% of the C3D-based models. The average accuracy and F_1 score of the ConvLSTM-based model are 94.1% (pm 0.91%) and 95.62% (pm 0.67%) being only around 2% lower than the C3D + SVM model of our previous work. Therefore, limited resources as those of mobile or embedded devices might justify the use of the MobileNetV2 combined with the ConvLSTM, as the decrease in the accuracy metrics is limited.

4.3. Limitations

The results of the research described in this paper are promising, but include some limitations. In fact, we focused on the accuracy of two models based on MobileNetV2, which is designed for mobile or embedded devices. Nevertheless, we ran our comparative tests in the cloud, using a GPU. Whereas the decrease in accuracy is limited and justifies the use of the best between the proposed models, tests on real mobile or embedded devices i.e., at the edge, are needed to get more general conclusions. Morevoer, our tests are based on a dataset of videos where the violence is simulated by actors. Tests on videos from real surveillance cameras are needed to



(a) (b)

Figure 2: ROC curve and AUC for the MobileNetV2 + Bi-LSTM (a), and MobileNetV2 + ConvLSTM (b) models.

Table 6
A comparison of the average values of the metrics for the two proposed models, based on MobileNetV2, with the metrics computed for the models of our previous work, based on C3D, over the five splits of the stratified shuffle split.

	Sensitivity	Specificity	Accuracy	F_1 score	AUC
MobileNetV2 + Bi-LSTM	96.34 ± 3.03%	61.98 ± 32.14%	85.08 ± 9.18%	90.08 ± 5.04%	94.38 ± 2.98%
MobileNetV2 + ConvLSTM	95.88 ± 1.17%	90.43 ± 3.09%	94.10 ± 0.91%	95.62 ± 0.67%	98.26 ± 0.46%
C3D + SVM	97.06 ± 0.80%	94.14 ± 1.51%	96.10 ± 0.71%	97.10 ± 0.53%	99.30 ± 0.23%
C3D + FC	97.82 ± 0.69%	91.12 ± 2.03%	95.62 ± 0.42%	96.78 ± 0.30%	98.94 ± 0.31%

confirm the accuracy results.

In addition, we collected the metrics on 16-frames chunks taken from the short videos of the AIRTLab dataset (the average length is 5.6 seconds), to make this work comparable with our previous research. Whereas most of the related literature performs tests on short videos, the accuracy on full length, real videos should be evaluated. Indeed, using the short chunks of frames taken from long videos as in our study might result in too many false positives. Thus, results on the chunks should be merged together with a proper strategy to maximize the accuracy on full length videos. To this end, a simple solution is labeling a part of a long video as violent only when a fixed number of consecutive 16-frames chunks are labeled as violent.

5. Conclusions

To be used in real applications, Artificial Intelligence and Deep Learning-based techniques need to take into account real time performances and be capable of running in mobile and embedded devices, in a edge computing fashion. In fact, an intelligent answer preserves its importance only if given in time, as remarked in [36]. Hence, in this paper, we proposed two Deep Neural Networks for the classification of videos into violent or not. Both networks are based on MobileNetV2, a CNN specifically designed for mobile and embedded devices. Such CNN is responsible for the extraction of the spatial features in the videos. We combined MobileNetV2 with a recurrent layer for the extraction of the temporal features as well. One of the two proposed models uses a Bi-LSTM layer as the recurrent module. Instead, the other uses a ConvLSTM.

We ran comparative tests on the AIRTLab dataset. The model using the ConvLSTM, the lightest in terms of required computational and memory resources between the two proposed in this paper, got the best accuracy, with an average AUC equal to 98.26% (± 0.46%). Compared to the models of our previous work, based on a 3D CNN, the decrease of performance in terms of AUC is around 1%, and 2% in terms of classification accuracy over the splits of the AIRTLab dataset. Such results encourage the use of mobile models for embedded devices. For example, this

might be useful to process data directly near the camera that is recording the security video and, thus, preserve the privacy while addressing public security.

Future works will address the identified limitations. In particular, tests on real mobile or embedded devices need to be performed to get more conclusive and general results.

Acknowledgments

The presented research has been part of the Memorandum of Understanding between the Università Politecnica delle Marche, Centro "CARMELO" and the Ministero dell'Interno, Dipartimento di Pubblica Sicurezza, Direzione Centrale Anticrimine della Polizia di Stato.

References

[1] E. L. Piza, B. C. Welsh, D. P. Farrington, A. L. Thomas, Cctv surveillance for crime prevention, Criminology & Public Policy 18 (2019) 135–159. doi:10.1111/1745-9133.12419.

[2] P. Contardo, P. Sernani, N. Falcionelli, A. F. Dragoni, Deep learning for law enforcement: A survey about three application domains, in: 4th International Conference on Recent Trends and Applications in Computer Science and Information Technology, volume 2872 of *CEUR Workshop Proceedings*, 2021, pp. 36–45. URL: http://ceur-ws.org/Vol-2872/paper06.pdf.

[3] Z. Xu, C. Hu, L. Mei, Video structured description technology based intelligence analysis of surveillance videos for public security applications, Multimedia Tools and Applications 75 (2016) 12155–12172. doi:10.1007/s11042-015-3112-5.

[4] P. Yadav, N. Gupta, P. K. Sharma, A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods, Expert Systems with Applications 212 (2023) 118698. doi:10.1016/j.eswa.2022.118698.

[5] D. Berardini, A. Galdelli, A. Mancini, P. Zingaretti, Benchmarking of dual-step neural networks for detection of dangerous weapons on edge devices, in: 2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), 2022, pp. 1–6. doi:10.1109/MESA55290.2022.10004469.

[6] P. Contardo, E. Di Lorenzo, N. Falcionelli, A. F. Dragoni, P. Sernani, Analyzing the impact of police mugshots in face verification for crime investigations, in: 2022 IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE), 2022, pp. 236–241. doi:10.1109/MetroXRAINE54828.2022.9967671.

[7] P. Contardo, P. Sernani, S. Tomassini, N. Falcionelli, M. Martarelli, P. Castellini, A. F. Dragoni, FRMDB: Face recognition using multiple points of view, Sensors 23 (2023). doi:10.3390/s23041939.

[8] K. B. Lee, H. S. Shin, An application of a deep learning algorithm for automatic detection of unexpected accidents under bad cctv monitoring conditions in tunnels, in: 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019, pp. 7–11. doi:10.1109/Deep-ML.2019.00010.

[9] A. Castillo, S. Tabik, F. Pérez, R. Olmos, F. Herrera, Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning, Neurocomputing 330 (2019) 151–161. doi:10.1016/j.neucom.2018.10.076.

[10] 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, 2012, pp. 1–6. doi:10.1109/CVPRW.2012.6239348.

[11] Y. Gao, H. Liu, X. Sun, C. Wang, Y. Liu, Violence detection using oriented violent flows, Image and Vision Computing 48-49 (2016) 37–41. doi:10.1016/j.imavis.2016.01.006.

[12] F. U. M. Ullah, M. S. Obaidat, A. Ullah, K. Muhammad, M. Hijji, S. W. Baik, A comprehensive review on vision-based violence detection in surveillance videos, ACM Comput. Surv. 55 (2023). doi:10.1145/3561971.

[13] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, A. F. Dragoni, Deep learning for automatic violence detection: Tests on the airtlab dataset, IEEE Access 9 (2021) 160580–160595. doi:10.1109/ACCESS.2021.3131315.

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4489–4497. doi:10.1109/ICCV.2015.510.

[15] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, W.-c. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, 2015, p. 802–810.

[16] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Networks 18 (2005) 602–610. doi:10.1016/j.neunet.2005.06.042.

[17] W. Niu, M. Sun, Z. Li, J.-A. Chen, J. Guan, X. Shen, Y. Wang, S. Liu, X. Lin, B. Ren, RT3D: Achieving real-time execution of 3D convolutional neural networks on mobile devices, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2021) 9179–9187. doi:10.1609/aaai.v35i10.17108.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520. doi:10.1109/CVPR.2018.00474.

[19] M. Bianculli, N. Falcionelli, P. Sernani, S. Tomassini, P. Contardo, M. Lombardi, A. F. Dragoni, A dataset for automatic violence detection in videos, Data in Brief 33 (2020) 106587. doi:10.1016/j.dib.2020.106587.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[21] 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), 2017, pp. 1–6. doi:10.1109/AVSS.2017.8078468.

[22] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, R. Sukthankar, Violence detection in video using computer vision techniques, in: P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, W. Kropatsch (Eds.), Computer Analysis of Images and Patterns, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 332–339. doi:10.1007/978-3-642-23678-5_39.

[23] J. Li, X. Jiang, T. Sun, K. Xu, Efficient violence detection using 3D convolutional neural networks, in: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–8. doi:10.1109/AVSS.2019.8909883.

[24] S. Accattoli, P. Sernani, N. Falcionelli, D. N. Mekuria, A. F. Dragoni, Violence detection in videos by combining 3D convolutional neural networks and support vector machines, Applied Artificial Intelligence 34 (2020) 329–344. doi:10.1080/08839514.2020.1723876.

[25] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, S. W. Baik, Violence detection using spatiotemporal features with 3D convolutional neural network, Sensors 19 (2019) 2472. doi:10.3390/s19112472.

[26] D. Freire-Obregón, P. Barra, M. Castrillón-Santana, M. D. Marsico, Inflated 3d convnet context analysis for violence detection, Machine Vision and Applications 33 (2022) 1–13. doi:10.1007/s00138-021-01264-9.

[27] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2015). URL: https://arxiv.org/abs/1409.1556.

[28] L. Ciampi, P. Foszner, N. Messina, M. Staniszewski, C. Gennaro, F. Falchi, G. Serao, M. Cogiel, D. Golba, A. Szczęsna, G. Amato, Bus violence: An open benchmark for video violence detection on public transport, Sensors 22 (2022). doi:10.3390/s22218345.

[29] V. E. D. S. Silva, T. B. Lacerda, P. B. Miranda, A. C. Nascimento, A. P. C. Furtado, Federated learning for physical violence detection in videos, in: 2022 International Joint Conference on Neural Networks (IJCNN), 2022, pp. 1–8. doi:10.1109/IJCNN55064.2022.9892150.

[30] L. Yang, Z. Wu, J. Hong, J. Long, MCL: A contrastive learning method for multimodal data fusion in violence detection, IEEE Signal Processing Letters (2022) 1–5. doi:10.1109/LSP.2022.3227818.

[31] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, Z. Yang, Not only look, but also listen: Learning multimodal violence detection under weak supervision, in: Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 322–339. doi:10.1007/978-3-030-58577-8_20.

[32] Y. LeCun, Y. Bengio, Convolutional Networks for Images, Speech, and Time Series, MIT Press, Cambridge, MA, USA, 1998, p. 255–258.

[33] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[34] A. Graves, N. Jaitly, A. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 2013, pp. 273–278. doi:10.1109/ASRU.2013.6707742.

[35] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (1997) 2673–2681. doi:10.1109/78.650093.

[36] A. F. Dragoni, P. Sernani, D. Calvaresi, When rationality entered time and became real agent in a cyber-society, in: Proceedings of the 3rd International Conference on Recent Trends and Applications in Computer Science and Information Technology, volume 2280 of *CEUR Workshop Proceedings*, 2018, pp. 167–171. URL: http://ceur-ws.org/Vol-2280/paper-24.pdf.