

Human skeletons and change detection for efficient violence detection in surveillance videos

Guillermo Garcia-Cobo, Juan C. SanMiguel *

Video Processing and Understanding Lab, Universidad Autónoma de Madrid, Spain

ARTICLE INFO

Communicated by Nikos Paragios

Dataset link: <https://github.com/atmguille/Violence-Detection-With-Human-Skeletons>

MSC:

68T45

2000

Keywords:

Machine vision and scene understanding

Violence detection

Pose estimation

Deep Learning

ConvLSTM

ABSTRACT

In our constantly monitored world, surveillance cameras play a crucial role in curbing crime and violence in public spaces by serving as a deterrent. To enhance their effectiveness, there is a growing need for automated tools that can detect crimes in real time. In this paper, we propose a novel deep learning architecture that accurately and efficiently detects violent crimes in surveillance videos. We rely on what we believe are the most essential pieces of information to detect violence, namely: human bodies and their interaction. To this end, we employ human pose extractors and change detectors as the input of our proposal. Subsequently, we combine them using a novel method, which relies on additions instead of multiplications to guarantee the transmission of information even when one of the inputs provides a zero-valued signal; outperforming other combination alternatives of the literature. Finally, to account for both spatial and temporal information, we use a convolutional alternative of the standard LSTM, the ConvLSTM. The experiments performed on several benchmark datasets demonstrate the efficacy and efficiency of our proposal, achieving state-of-the-art results with much fewer trainable parameters. We release the code to replicate the proposed architecture at <https://github.com/atmguille/Violence-Detection-With-Human-Skeletons>.

1. Introduction

In the last years, urban violence and crime have posed an increasing threat to society. According to the *Global Study on Homicide* (United Nations Office on Drugs and Crime, 2019), elaborated by the United Nations (UN), almost half a million people were killed in homicides worldwide in 2017. This surpasses by far the 89 000 killed in armed conflicts and the 19 000 killed in terrorist attacks. Moreover, the report states that there is a clear uptrend in the numbers. One of the recurrent policies adopted by local authorities to reduce crime has been installing surveillance cameras in public places. In 2014, Jenkins (2016) estimated that there were 245 million surveillance cameras, while in 2019, the estimation (Philippou, 2019) was 770 million. These reports expect that the number will continue to increase, reaching 1 billion in the coming years (one for every 8 people).

These numbers have resulted in a large amount of video footage, making it impossible to accurately monitor in real time these videos by human operators. It is a fact that surveillance cameras provide evidence after crimes have been committed, but they are rarely used to prevent or stop criminal activities in real time. However, being able to do so would be highly desired, as it would dramatically reduce reaction times. In many cases, this is a key factor to prevent fatalities and/or reducing the suffering of victims.

Recent contributions of Artificial Intelligence (AI) to violence detection and other related fields have proven that it could be of great help to tackle this situation. Nevertheless, to be able to make a real impact on public safety, AI solutions must balance performance and efficiency. Being able to rapidly alert of crimes occurring is as important, if not more, as detecting them with the highest accuracy.

To this end, we propose a novel architecture for the detection of violence in surveillance videos. Our main goal is to achieve the aforementioned balance: computational efficiency during training and inference, translated into reducing the number of trainable parameters, while keeping a high model accuracy. The main contributions of the proposed architecture are summarized as follows:

1. We simplify as much as possible the model's input by extracting the essential information needed to detect violence: human bodies and their interaction. Such simplification benefits the development of a real-time architecture for crime detection.
2. We combine the extracted features from the interactions and the dynamic temporal changes of human bodies in novel ways to exploit the full potential of both sources of information.
3. We propose an efficient usage of the ConvLSTM (Shi et al., 2015) for two-stream architectures devoted to violence detection, resulting in 5x fewer parameters than public alternatives and running in real time without sacrificing accuracy.

* Corresponding author.

E-mail addresses: ggcobo806@gmail.com (G. Garcia-Cobo), juancarlos.sanmiguel@uam.es (J.C. SanMiguel).

In order to describe our proposal, the rest of the document is organized as follows. In Section 2, we do a thorough review of the state of the art in the violence detection task. The proposed architecture is presented in Section 3, with details for the main components and their connections. The performance of the architecture and its components are evaluated through Section 4. Finally, in Section 5 we present the conclusions of this paper.

2. Related work

As for many Computer Vision problems, related work in the violence detection topic has been heavily influenced by the rise of Deep Learning (DL) and Convolutional Neural Networks (CNN). Thus, we should distinguish between approaches employing classical or DL approaches. More importance is given to the latter, as they achieve better results. Two general surveys that describe previous work for both approaches are Omarov et al. (2022), Stergiou and Poppe (2019), while Jain and Vishwakarma (2020) focuses on CNN-based approaches.

2.1. Classical approaches

Classical approaches for image processing consist in extracting a numerical vector (called *descriptor*) from images in order to train standard classification models. Taking this into account, classical work in the field of violence detection focus their efforts on defining descriptors that gather spatio-temporal information of the videos.

The early proposal of Bermejo Nieves et al. (2011) used two well-known descriptors in the activity recognition field: the Spatio-temporal Interest Points (STIP) (Laptev, 2005) and the Motion Scale Invariant Feature Transform (MoSIFT) (Chen and Hauptmann, 2009). The former is an extension of the Harris corner detector (Harris and Stephens, 1988) which aims to detect pixels with significant intensity variation across space and time, while the latter adds motion information via optical flows to the popular SIFT (Lowe, 2004) descriptor. Based on Bermejo Nieves et al. (2011), MoSIFT demonstrates an overall better performance than STIP in the analyzed datasets, although it is also more computationally expensive.

On the other hand, Hassner et al. (2012) designed a specific descriptor to represent violence information: Violent Flows (ViF). ViF considers how optical flow changes over time, rather than the magnitude of the flows themselves. The authors claim that ViF enables real-time violence detection, which is not possible with previous descriptors. Incorporating the orientation of the flows into the ViF descriptor resulted in Oriented Violent Flows (OViF), proposed by Gao et al. (2016). OViF improves ViF in non-crowded scenarios and a general performance boost is observed when combining both descriptors.

Moreover, Ben Mabrouk and Zagrouba (2017) proposed DiMOLIF, a descriptor that combines both local (STIP) and global (optical flow) information by analyzing the magnitude and orientation of optical flows in blocks that contain a sufficient number of STIP points. This combination and filtering of information makes DiMOLIF outperform ViF and OViF individually, but achieves similar results than when combining both. Finally, Ribeiro et al. (2016) tried to capture violent movements based on Histograms of Optical Flow.

2.2. Deep Learning approaches

Deep Learning techniques have surpassed classical approaches in many research areas. The well-known Convolutional Neural Networks (CNNs) efficiently learn the previously handcrafted features, with greater generalization capabilities. Ding et al. (2014) used 3D CNNs to detect violence in videos. These networks are just an extension of 2D CNNs, expanding 2D filters to the time axis and thus capturing spatio-temporal features. However, because of this expansion, 3D CNNs require many more parameters. Trying to improve on this, Li et al. (2019) proposed an efficient 3D CNN architecture by reducing the

kernel size and reusing features (each layer receives the feature maps produced by all its preceding layers). To reduce even more the number of parameters, Wang et al. (2022) proposed to mix 3D convolutions with 2D ones in their lightweight architecture.

The need to work with 3D CNNs comes from having to process not only spatial but also temporal information. An alternative to having filters managing both is to process each type of information at a time. That is, first use a 2D CNN to extract spatial features, then use a network that is able to process temporal information in these spatial features, as proposed in Asad et al. (2019). They use the pretrained VGG16 (Simonyan and Zisserman, 2015) network to extract low and high level spatial features which are then fed into a Long-Short Term Memory network (LSTM). A special case of this combination of 2D CNNs with LSTMs is described in Vijeikis et al. (2022), where an encoder-decoder (U-Net Ronneberger et al., 2015) is used before the LSTM.

However, a regular LSTM takes in 1-dimensional inputs, so projecting the spatial 2-dimensional features learned by the CNN may result in a loss of information. To overcome this, Sudhakaran and Lanz (2017) proposed using a ConvLSTM (Shi et al., 2015) to aggregate, across time, the spatial features produced by a pretrained CNN. Their experiments show that ConvLSTM achieves better results with fewer parameters than a regular LSTM. Moreover, they feed difference of adjacent frames to the model instead of the raw video, as a time-efficient change detector, substituting the computationally expensive optical flow. Hanson et al. (2019) extended this architecture by using bidirectional ConvLSTMs, which process the videos forward and backwards.

Lately, given the success of incorporating extra input streams of information on general activity recognition tasks (Simonyan and Zisserman, 2014), there have been proposals that have adopted this strategy to detect violence. For instance, Cheng et al. (2021) used two streams of 3D CNNs to process raw videos and optical flows independently, and then combined the spatio-temporal features learned in both pipelines to output the prediction. Another example is Islam et al. (2021), which used two independent pipelines that process raw videos with their background suppressed and difference of adjacent frames, respectively. Each pipeline consists of a 2D CNN and a Separable ConvLSTM (a ConvLSTM but with depthwise separable convolutions). It is again noticeable the use of difference of adjacent frames as a replacement of the optical flow.

With the surge of two stream approaches, there has been a number of proposals on how to combine the information produced by the two pipelines. Cheng et al. (2021) multiplied element-wise the feature maps produced by the two pipelines, and then used a 3D CNN to aggregate information. Islam et al. (2021) explored various ways of combining the information: concatenating by channel, element-wise addition and element-wise multiplication after applying a ReLU and a sigmoid activation to each pipeline. The latter was the best performing. In the general action recognition domain, there has been other proposals of combination using addition. For instance, Shi et al. (2019) computed probabilities of each class for each stream, and then added them for the final prediction. On the other hand, Luvizon et al. (2018) multiplied element-wise the feature maps and collapsed spatial information using addition (similar to a 2D average pooling but without normalizing).

An alternative to the previously described literature is proposed in Su et al. (2020). Violence is detected by studying the interrelationships between human body points. To do that, they first detect human skeletons points using a pretrained network. Following this, they model the interaction between points using a custom Graph Neural Network (GNN), which they called Skeleton Points Interaction Learning (SPIL) module.

Finally, a summary of the relevant aspects used by the different Deep Learning alternatives, compared to those that compose our proposal, is given in Table 1.

Table 1
Summary of state-of-the-art DL approaches.

Model	Year	2D CNN	3D CNN	LSTM	ConvLSTM	Person detector	Skeletons	2 pipelines
3D CNN (Ding et al., 2014)	2019		✓			✓		
Efficient 3D CNN (Li et al., 2019)	2019		✓					
Flow Gated Net (Cheng et al., 2021)	2020		✓					✓
SPIL (Su et al., 2020)	2020						✓	
CNN + LSTM (Asad et al., 2019)	2020	✓		✓				
ConvLSTM (Sudhakaran and Lanz, 2017)	2017	✓			✓			
BiConvLSTM (Hanson et al., 2019)	2018	✓			✓			
SepConvLSTM (Islam et al., 2021)	2021	✓			✓			✓
Spatio-Temporal Modeling (Kang et al., 2021)	2021	✓						
Lightweight 2D+3D CNN (Wang et al., 2022)	2022	✓	✓					
Person detector + CNN (Choqueluque-Roman and Camara-Chavez, 2022)	2022	✓	✓			✓		
U-Net + LSTM (Vijeikis et al., 2022)	2022	✓		✓				
Ours	2022				✓		✓	✓

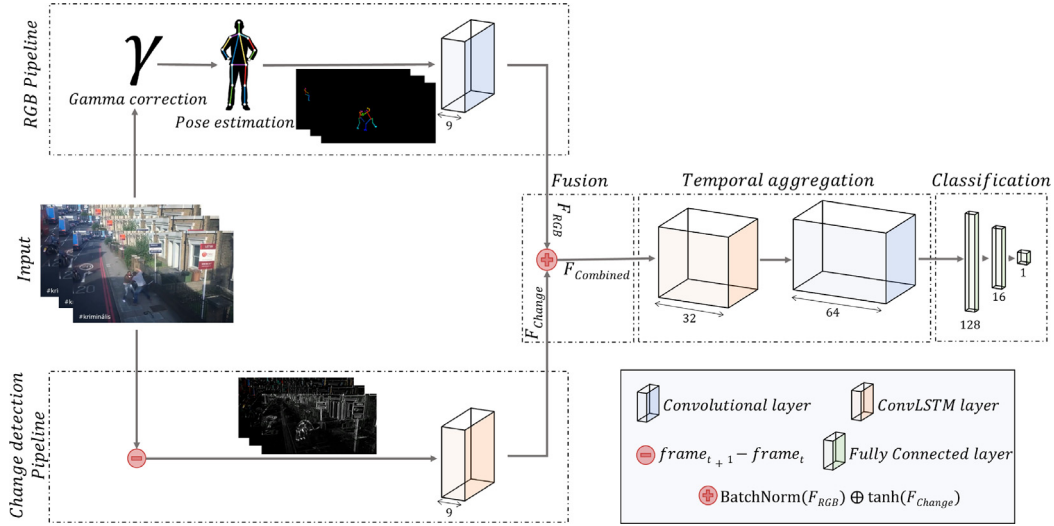


Fig. 1. Explicit proposed architecture. It comprises two pipelines: one for extracting people's skeletons in the scene and another for estimating their dynamic temporal changes between frames. After being combined, the spatio-temporal features are temporary aggregated using a ConvLSTM, followed by a final classifier. The architecture processes a sequence of frames of arbitrary length, resulting in a single prediction for the entire sequence. Filters are applied to each frame in the sequence until temporal information is aggregated by the ConvLSTM. Optimal parameters for the convolutional and fully connected layers have been determined empirically and are indicated below the respective layers.

3. Proposed architecture

Our proposal is shown in Fig. 1, with its logical blocks explicitly detailed. Note that we have also added the optimal number of filters of the components that have convolutions, which have been determined empirically. In the following subsections, we describe the different blocks that the architecture consists of, as well as the connections between them.

3.1. RGB pipeline

The input to this pipeline is the original sequence of RGB frames, and thus is in charge of adding intra-frame information. To reduce the amount of processed information and facilitate the learning process, we believe that violence may be identified by the interaction of certain body parts instead of using all the RGB pixels contained in each frame. Therefore, we propose to extract the skeletons of the people in the scene.

3.1.1. Human pose estimation

Although there have been many proposals to extract pose information (Dang et al., 2019), we are interested in models able to detect more than one person in a single frame, which is called *Multi-person Pose Estimation*. Once the pose information is extracted, there would be different ways to present it to our model. First, we could represent the coordinates of the detected joints of the skeletons as a set of

interconnected points, i.e., building a graph. However, efficiency would much depend on the number of people in the scene and the interrelations considered. Another option would be to render the detected skeletons and process the resulting images. For this, we could render the confidence heatmaps of the detector or the final predicted skeletons. While the former could contain richer information, we believe that it could add an extra layer of undesired noise that would require a more complex model to process. Aiming to use the most lightweight option, we decide to render the predicted skeletons in their simplest form.

An example of the chosen skeleton representation is shown in Fig. 2. Note that different parts of the skeletons are colored with specific colors. This coloring adds a valuable piece of information to the data being processed by our architecture, since it enables our proposal to differentiate between different parts of the body. Moreover, in an effort to reduce even further the background noise, we decide to render the skeletons over a black background, as shown in Fig. 2(b).

3.1.2. Enhancing skeletons extraction

While keeping the skeletons over a black background can remove useless background noise, we risk losing performance if the skeletons are not well detected. This is because, in that case, the model would not have extra information (just black pixels) to overcome the misdetection. Authors in Pedersen et al. (2019) have demonstrated that specific pre-processing techniques can enhance the performance of pose estimators in certain scenarios. Among the options tested, we focused on the most effective ones: increasing contrast and histogram equalization.

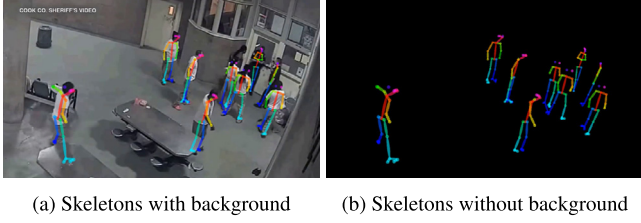


Fig. 2. Skeletons' representation example for video *0_DzLklZa0.3.avi* of RWF-2000 (Cheng et al., 2021) fight training set, obtained using OpenPose (Cao et al., 2021).

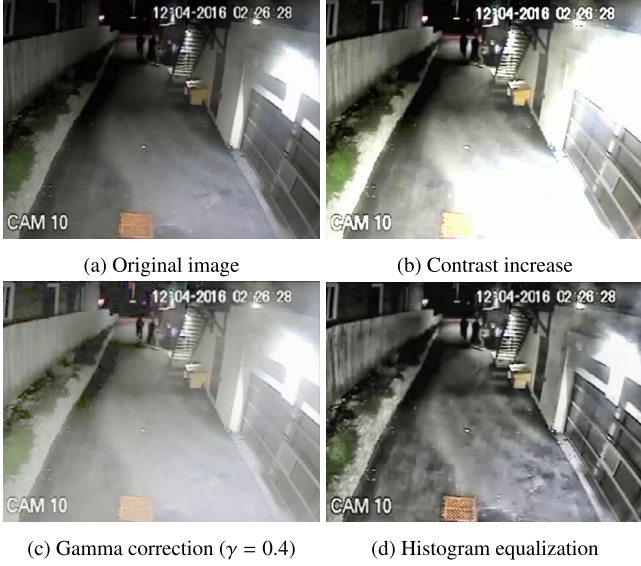


Fig. 3. Alternatives to enhance pose estimators' performance. Gamma correction and histogram equalization output a similar result, but the latter is much more computationally expensive.

Increasing image contrast involves multiplying each pixel by a factor, but it is important to consider variations in brightness across different areas of the image. To address this, we propose using gamma correction (see section 3.2 of Gonzalez and Woods, 2018). For an image I with pixel values in the range $[0, 255]$, the gamma correction is given by

$$I' = \left(\frac{I}{255} \right)^\gamma \cdot 255, \quad (1)$$

where selecting $\gamma < 1$ achieves the desired effect. Another technique to improve performance is histogram equalization (see section 3.3 of Gonzalez and Woods, 2018), which spreads pixel values across the range of $[0, 255]$, resulting in a wider and more uniform distribution compared to the original image.

From the example outputs shown in Fig. 3, we can see that both the gamma correction and the histogram equalization achieve similar outputs. However, the latter is much more computationally expensive than the former, so the gamma correction is preferred. Nevertheless, the pre-processing techniques may not be needed at all, if the settings where the video is recorded are appropriate.

3.2. Change detection pipeline

Once we have extracted pose information from the original frames, it may be useful to also feed the model with inter-frame information. Other authors (Sudhakaran and Lanz, 2017; Hanson et al., 2019) have used change detectors as their sole input with good overall results.

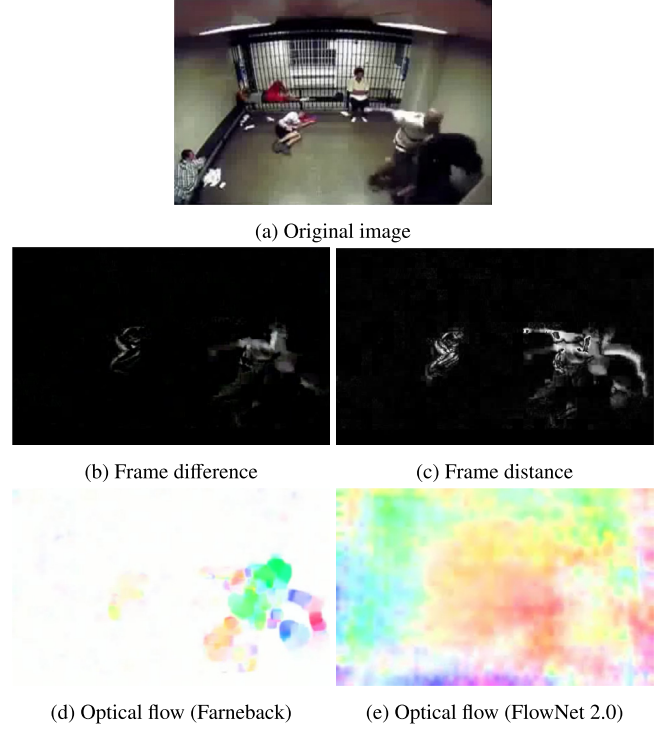


Fig. 4. Comparison of change detectors. The parts that are moving in the original video are the two men in the bottom right corner and the man laying down in the center. Optical Flow results are rendered as RGB following the algorithm available in NVIDIA (2022).

There are different alternatives to detect changes between a pair of frames, which we describe in the following lines:

- **Frame difference:** the difference between the current and the next frame is used as the input. That is,

$$frame_diff_t = frame_{t+1} - frame_t. \quad (2)$$

Note that the result of this operation has the same number of channels as the frames. Sudhakaran and Lanz (2017), Hanson et al. (2019) use this approach.

- **Frame distance:** the distance between two frames is defined as the L^2 norm of the frame difference. That is,

$$frame_dist_t = \sqrt{\sum_{d=1}^3 (frame_{t+1}^d - frame_t^d)^2}, \quad (3)$$

where d is the channel. In this case, the result of the operation compacts the information into a single channel. In addition, all the pixels of $frame_dist_t$ will be non-negative. Kang et al. (2021) uses this approach.

- **Optical Flow:** many authors have used optical flow in the action recognition field to detect changes, estimating movement, and Cheng et al. (2021) specifically in violence detection. Two ways of computing the optical flow have been considered: a classical one (Farneback's algorithm Farneback, 2003, implemented in OpenCV) and a Deep Learning one (FlowNet 2.0 Ilg et al., 2017). The 2-dimensional results are converted to 3 channels using NVIDIA (2022).

Frame difference, has an additional source of information as compared to frame distance: the sign. Apart from containing information about what parts of the frame are changing, by taking the sign into account one can determine the temporality of the change. This is

because negative values indicate that there is something in the current frame that will not be there in the next one, and vice versa for the positive sign. Fig. 4 shows a sample output of all the proposed estimators for a specific frame. At first sight one could think that frame distance (Fig. 4(c)) offers more information, but it has to be mentioned that in Fig. 4(b), we are not able to visualize the full potential of frame difference because there is no way to render the negative values without rescaling. Moreover, we see that FlowNet (Fig. 4(e)) produces much more noise than the classical Farneback's algorithm (Fig. 4(d)). Also, both optical flow approaches are very computationally expensive.

Based on the previous discussion, frame difference appears to be the best available option for the change detection pipeline and thus is chosen as the input to this pipeline. This is mostly because the sign information it provides and the computation efficiency.

3.3. Fusion of the two pipelines

Based on our hypothesis that pose and inter-frame information are essential to detect violence, together with finding examples in the literature (Simonyan and Zisserman, 2014; Cheng et al., 2021; Islam et al., 2021) that used two streams of information, we decided to combine both pipelines in the proposed architecture.

In order to incorporate as much inter-frame information as possible, we decided to further pre-process the change detection pipe before merging it to the other pipeline. By applying a ConvLSTM to the result of the frame difference before the merge, we supply the model with more information of the inter-frame relationships. Moreover, so as to match dimensions and prepare the skeletons' data for the merge, we add a convolutional layer to the RGB pipeline.

On the other hand, combining both sources of information would also reduce the risk of missing information when the skeletons are not well detected, as the change detection pipeline could still supply information to the architecture. An example of this situation is shown in Fig. 5, where skeletons of the people that are fighting (see bottom right corner of Fig. 5(a)) are not detected, but the frame difference is able to capture this information. However, the combination has to be done in an intelligent way to take full advantage of both sources.

Several authors have explored different schemes for combining two pipelines (Cheng et al., 2021; Islam et al., 2021; Shi et al., 2019; Luvizon et al., 2018). Concatenating, adding and multiplying features are the most common options, with the latter performing better in the literature. When multiplying, in Cheng et al. (2021) and Islam et al. (2021), the sigmoid activation is applied to one of the pipelines, interpreting it as a weighting factor between 0 and 1. Following this logic, we could apply the sigmoid function to the feature maps of the change detection pipeline, so they can weight which areas of the RGB pipeline the model should pay more attention to. Formally,

$$F_{combined} = \text{ReLU}(F_{RGB}) \odot \text{sigmoid}(F_{Change}), \quad (4)$$

where F_{RGB} and F_{Change} are the corresponding feature maps.

However, multiplying has the restriction that both pipelines have to carry information for it to transmit a non-zero signal. That is, if there are zero-valued regions in the RGB pipeline, it is not possible for the multiplication to provide an effective combination in those regions, regardless change detection information, and vice versa. As we want both pipelines to enrich the information that is provided to the next layer, not restricting it, we designed a novel combination technique. It is based on the following principles. First, we substitute the multiplication operation with addition, so both pipelines can contribute. As explained in Section 2, performing simple addition has already been explored, but it was not as effective as multiplication (an explicit comparison is done in Islam et al., 2021). Thus, we propose an addition-based replication of the weighting behavior of the multiplication, which we believe is the differential factor. For that purpose, we apply a tanh to the change detection pipeline, so the sign of the values of this function (in the $[-1, 1]$ interval) can play the role of activation/deactivation when being

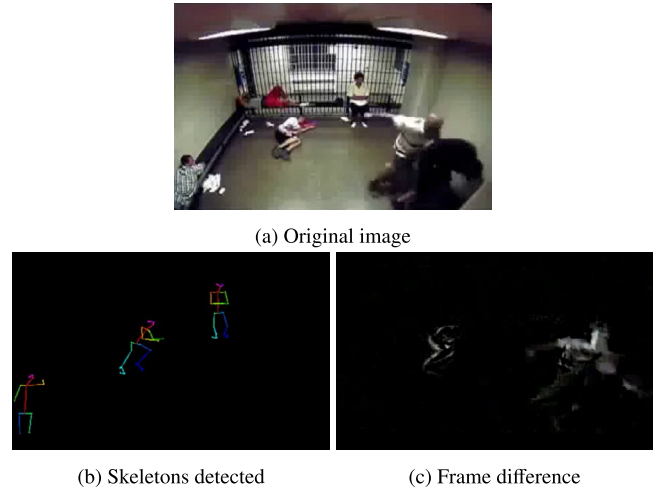


Fig. 5. Motivation of merging pipelines: they can complement each other in case one fails to provide information. In this case, the skeletons of the people that are fighting (bottom right corner) are not detected, but the frame difference is able to capture this information.

added that was previously played by the outputs of the sigmoid when multiplying. Finally, if the values of the other pipeline are large or with large variance, adding a -1 or a 1 will be totally irrelevant. To understand this reasoning, one can think of the impact of adding a ± 1 to values in $[0, 255]$ in comparison with values in $[-0.5, 0.5]$. Therefore, it is desired that values in the RGB pipeline are transformed to be around zero with small variance, which can be achieved by applying a Batch Normalization layer (Ioffe and Szegedy, 2015) (by setting $\gamma = 1$, $\beta = 0$ of the original paper to avoid rescaling). Thus, the proposed combination is

$$F_{combined} = \text{BatchNorm}(F_{RGB}) \oplus \tanh(F_{Change}), \quad (5)$$

where \oplus is the pointwise addition.

3.4. Temporal aggregation and classification

At this point of the architecture, we have obtained a set of feature maps $F_{combined}$ for each frame in the input sequence. We would like to aggregate them into video-level features, since our objective is to detect if there is violence in the entire input sequence.

3.4.1. Temporal aggregator - ConvLSTM

In order to obtain the desired video-level feature map, we would like to use a layer that is able to process spatial and temporal features. As described in Section 1, the ConvLSTM is able to take both into account. Moreover, to understand how the ConvLSTM could also serve as the desired temporal aggregator, we have to recall how a standard LSTM works. This layer produces a state (feature maps) for each element of the input sequence. Since the last state's computation involves all the elements of the sequence, via the information accumulated in the previous state, it can be considered as the aggregated state of the entire sequence. Therefore, after processing the last frame of the video sequence, the ConvLSTM outputs a set of feature maps, i.e., the last state of the network, as a compact video-level summary.

Now that we have aggregated the information, we may want to post-process the obtained representation before feeding it to the classification block. It has to be taken into account that we have just collapsed a significant amount of information of the entire sequence into a set of feature maps. Therefore, refining the extracted spatio-temporal summary seems like a reasonable step before losing spatial and temporal sense in the fully connected layer. To perform this task efficiently, we

Table 2

Comparison of available datasets.

Source: Adapted from Cheng et al. (2021).

Authors	Year	Dataset	Data Scale	Length/Clip (sec)	Resolution	Annotation	Scenario
Blunsden and Fisher (2009)	2010	BEHAVE	4 Videos (171 Clips)	0.24–61.92	640 × 480	Frame-Level	Acted Fights
Rota et al. (2015)	2015	RE-DID	30 Videos	20–240	1280 × 720	Frame-Level	Natural
Demarty et al. (2014)	2015	VSD	18 Movies (1 317 Clips)	55.3–829.4	Variable	Frame-Level	Movie
Perez et al. (2019)	2019	CCTV-Fights	1 000 clips	5–720	Variable	Frame-Level	Natural
Bermejo Nieves et al. (2011)	2011	Hockey Fight	1 000 Clips	1.6–1.96	360 × 288	Video-Level	Hockey Games
Bermejo Nieves et al. (2011)	2011	Movies Fight	200 Clips	1.6–2	720 × 480	Video-Level	Movie
Hassner et al. (2012)	2012	Crowd Violence	246 Clips	1.04–6.52	Variable	Video-Level	Natural
Yun et al. (2012)	2012	SBU Kinect Interaction	264 Clips	0.67–3	640 × 480	Video-Level	Acted Fights
Sultani et al. (2018)	2018	UCF-Crime	1 900 Clips	60–600	Variable	Video-Level	Surveillance
Cheng et al. (2021)	2020	RWF-2000	2 000 Clips	5	Variable	Video-Level	Surveillance

consider the feature maps as independent features that should be post-processed independently. This type of independent spatial processing between feature maps (channels) can only be achieved by a Depthwise convolution (Chollet, 2017). Apart from treating information across channels independently, Depthwise convolutions require much less parameters than Standard convolutions.

3.4.2. Classification

Finally, we have to transform the refined feature maps in a 1-dimensional vector that could be fed to the fully connected layer. This is done by using a *Global Average Pooling layer* (Lin et al., 2014). For the fully connected layer, we use a simple architecture, trusting that the previously extracted features are sufficiently representative, without adding a significant number of computations. Islam et al. (2021) uses $128 \rightarrow 16 \rightarrow 1$ with good results, so we will use this simple architecture.

4. Experiments and results

In this section, we evaluate the contribution of the different components of the proposed architecture and the overall performance of our best model, both on the reference RWF-2000 dataset and on other benchmarks.

4.1. Datasets

Over the years, many datasets have been released for violence detection. Some of them were explicitly created for this task, while others are a subset of more general datasets for activity recognition. A great comparison of available options, elaborated by Cheng et al. (2021), can be found in Table 2. The RWF-2000 dataset stands out as the preferred choice because of having the largest number of videos with fixed length that eases training DL architectures. Moreover, our goal is to detect violence in surveillance scenarios, so the fact that videos in this dataset are recorded from surveillance cameras suits the reasoning behind our choice. Finally, this dataset of 2000 videos is completely balanced and comes with a validation split of 20% of the videos, which eases comparisons with other proposals.

Although the chosen option to train and validate our architecture is RWF-2000, we also evaluate our proposal on other datasets to demonstrate its generalization ability. To this end, we also report our performance for the Hockey (Bermejo Nieves et al., 2011), Movies (Bermejo Nieves et al., 2011) and Crowd (Hassner et al., 2012) datasets, as they are the ones usually used in the literature.

4.2. Implementation details

For the pose estimation module in the proposed architecture, we use OpenPose (Cao et al., 2021) because of the following reasons. First, it achieves state-of-the-art results in public benchmarks for *Multi-person Pose Estimation*. Second, as reported in their paper, its computing time does not depend on the number of people in the video, while other alternatives' runtime grow with the number of people. Depending on the surveillance scenario, this can be a determining factor.

In terms of the input sequence, it is a usual practice in the literature to reduce the number of frames processed. This comes from the fact that adjacent frames usually have redundant information. For example, Cheng et al. (2021) uses 64 frames, while Islam et al. (2021) uses 32. We propose an intermediate option with 50 uniformly sampled frames per video, so we reduce FPS from 30 to 10, since videos in RWF-2000 are composed of 150 frames. Next, the selected frames are resized to a dimension of 100×100 . This image size is significantly lower than the one used in many of the state-of-the-art alternatives (224×224), which will positively contribute to the efficiency of our proposal. The network is trained with batches of 10 videos for 30 epochs. The optimizer used is Adam (Kingma and Ba, 2014) with a fixed learning rate of 0.001. Finally, in an effort to have reproducible results, the seed of all the random number generators is set to 0. For each experiment, we show the maximum accuracy obtained in the training and the validation set over the 30 epochs. Only the accuracy is reported to ease comparisons between experiments and with other proposals, as it is the standard metric used in the literature. Nevertheless, we provide a deeper analysis of the performance of our best model at the end of this section. To run the experiments, 16GB of GPU are required for training and 3GB for inference.

4.3. RGB pipeline analysis

We first analyze the contribution of the RGB pipeline. To do that, we remove the change detection pipeline and evaluate performance of different options for the RGB pipeline. The architecture used in this section is shown in Fig. 6. Note that the convolutional layer that precedes the merging is also removed, as its objective was to adapt the input to the merging layer. We will be adding the different components to test (skeletons, gamma, ...) to this schematic architecture.

4.3.1. Results for different inputs

Table 3 shows the performance of the architecture of Fig. 6 with different inputs. First, we see that the architecture performs relatively well when working with the raw videos. So, even with a lot of extra noise, i.e., parts of the image not interesting for violence detection, the model is able to learn something. If we add the pose information, i.e., skeletons, the performance increases even with the background noise. However, in both cases, there are significant variations between the performance in the train and in the validation set, due to the noise. Finally, removing all the noise and only keeping the essential information, the skeletons, there is a significant boost in the model accuracy, both in the train and in the validation set. This strengthens our hypothesis that analyzing human bodies and their interactions should be enough to detect violence, and that the model benefits from the reduction of the amount of noise.

4.3.2. Results for video pre-processing

As we mentioned when describing the proposed architecture, we risk losing information if skeletons are not well detected. So as to maximize the performance of the pose estimation model in our videos, we test two pre-processing operations: gamma correction and histogram

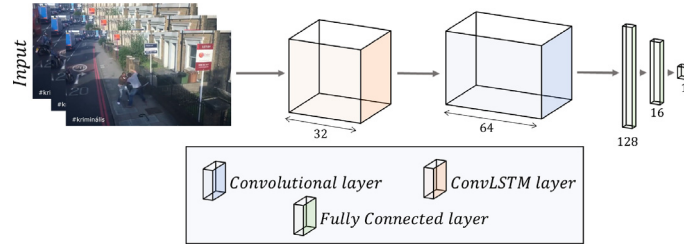


Fig. 6. Architecture to test the contribution of the RGB pipeline alone. Numbers below the different layers represent the number of filters or neurons used, which are fixed for all the experiments of the RGB pipeline.

Table 3

Results for different inputs using RGB pipeline (see Fig. 6).

Model	Train accuracy	Validation accuracy
RGB frames	75.06%	78.50%
Skeletons with background	73.50%	80.00%
Skeletons without background	87.88%	84.75%

Table 4

Results for video pre-processing using RGB pipeline (see Fig. 6).

Model	Train accuracy	Validation accuracy
Skeletons	87.88%	84.75%
Skeletons + Gamma correction	88.13%	85.50%
Skeletons + Histogram equalization	86.19%	84.75%

equalization. In particular, gamma correction is done with $\gamma = 0.67$ and histogram equalization is computed with the Contrast Limited Adaptive Histogram Equalization (CLAHE) implemented by OpenCV (OpenCV, 2022). The results are shown in Table 4. While histogram equalization does not provide any improvements over not using it, gamma correction results in a great performance boost.

4.4. Change detection pipeline analysis

After analyzing different alternatives for the RGB pipeline, we now move on to the change detection pipeline. Again, to understand its potential as a sole input, we remove the other pipeline from the architecture. Moreover, the ConvLSTM that precedes the merging is also removed. This is because we want the architecture to be as similar as possible to the one used for the RGB pipeline, in order to be able to compare the results. In fact, the testing architecture is the same as the shown in Fig. 6, with the addition of the change detectors right after the input.

Table 5 shows the architecture performance for each change detector, which were described in Section 3.2. A few lessons are learned from these results. First and foremost, inter-frame information is much better than raw videos, but not as good as pose information. Out of the different alternatives, frame difference is the most effective. We conclude that the model is able to extract the sign information described in Section 3.2, as frame difference performs better than frame distance. Furthermore, the optical flow is not as effective as the other simpler detectors. Moreover, both optical flow methods are very computationally expensive. Finally, the Deep Learning approach, which is less expensive than the classical Farneback algorithm, appears not to be as effective as the classical one.

4.5. Fusion and post-processing analysis

Once the potential of each pipeline has been evaluated separately, it is time to study the influence of the components that merge these pipelines. To do so, we simplify the architecture to its minimal form and start adding components to it. Thus, we start with the architecture shown in Fig. 7. Note the absence of the ConvLSTM layer of the change detection pipeline and the post-processing layer. The temporal aggregator is kept as it is an essential component.

Table 5

Results for different change detectors using only the change detection pipeline (similar to Fig. 6).

Model	Train accuracy	Validation accuracy
Frame difference	87.75%	83.75%
Frame distance	84.38%	81.75%
Optical Flow (farneback)	84.87%	80.50%
Optical Flow (flownet)	80.19%	79.25%

Table 6

Results for fusion alternatives in comparison to single streams.

Model	Streams	Train accuracy	Validation accuracy
Skeletons + Gamma (F_{RGB})	1	88.13%	85.50%
Frame difference (F_{Change})	1	87.75%	83.75%
Multiply(F_{RGB} , F_{Change})	2	85.00%	85.50%
Ours: Add(F_{RGB} , F_{Change})	2	88.94%	87.00%

4.5.1. Influence of fusion

As mentioned in Section 3.3, we propose a novel technique that aims to combine the two pipelines in a more effective way than other authors have done in the literature. Both alternatives were formally described in Eqs. (4) and (5). Table 6 shows that our proposed combination method outperforms by a great margin the previous combination alternative.

4.5.2. Influence of post-processing

In Section 3.3 it was also mentioned that we decided to extract further temporal information by using a ConvLSTM in the change detection pipeline before the merge. Note that, as commented, we have to add a convolutional layer to the RGB pipeline to match the new dimension of feature maps. Moreover, in Section 3.4.1, we proposed to post-process the output of the temporal aggregator before feeding the features to the final fully connected layer. Although our initial suggestion is to consider the feature maps as independent features, another option is to consider them as a single frame with multiple channels. For the former, we have to use a Depthwise convolution, while for the latter we can use Standard convolutions. Table 7 shows the performance of the different alternatives for post-processing. Apart from the training and validation accuracy, we have included the number of trainable parameters, to give a better understanding of the complexity of each proposal. First, we can observe that performance increases significantly after adding these post-processing layers. Also, as noted in Section 3.4.1, we can see that Standard convolutions require much more parameters than Depthwise convolutions, which leads to overfitting. All in all, the best proposal is able to detect violence with great accuracy and very few trainable parameters.

4.6. Performance analysis of the best trained model

4.6.1. Detection capabilities

After demonstrating the advantages of the modules in the proposed architecture to increase accuracy, we can now further analyze its performance. First, our best model is trained during 50 epochs, to

Table 7

Results with post-processing. ‘Pipelines merged’ refers to the baseline shown in Fig. 7. ‘RGB_Conv + CD_ConvLSTM’ refers to the post-processing before merging, both the convolution applied to the RGB pipeline and the ConvLSTM applied to the Change Detection pipeline. StandardConv and DepthConv refer to the two proposed options for the post-processing before fully connected layer. Both consist of 64 filters.

Pipelines merged	RGB_Conv + CD_ConvLSTM	StandardConv	DepthConv	Train accuracy	Validation accuracy	Trainable parameters
✓				88.94%	87.00%	46 921
✓	✓			90.94%	88.75%	57 847
✓	✓	✓		94.19%	88.75%	80 439
✓	✓		✓	89.88%	89.50%	62 583

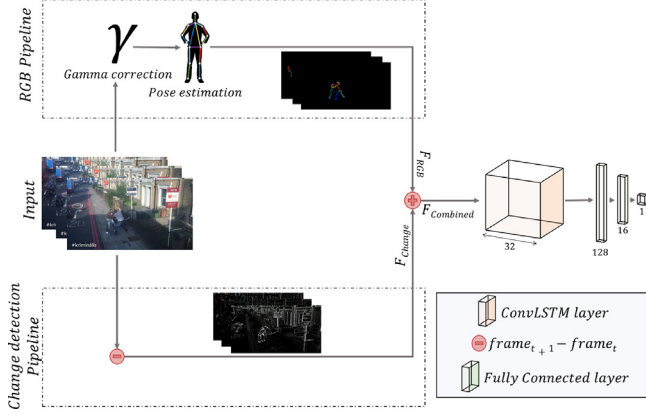


Fig. 7. Minimal architecture for testing the combination of the two streams, as well as the other components. Numbers below the layers indicate the number of filters or neurons.

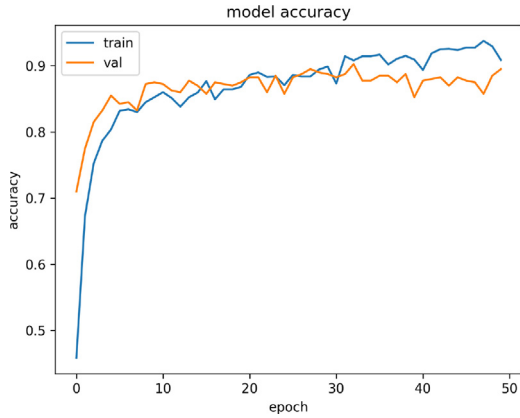


Fig. 8. Final model accuracy evolution. After 50 epochs, the best model achieves 90.25% in the validation set and 92.37% in the training set.

extract the full potential and achieve the maximum accuracy. After this training, our proposal achieves a maximum accuracy of 90.25% in the validation dataset, with 92.37% of accuracy in the training set. The evolution of the accuracy during the training is shown in Fig. 8.

Although accuracy is the standard metric used in the literature, we provide the predictions in the validation set to fully understand the performance of our proposal. In Table 8 we see that the model has a slightly higher capacity to detect fights (*recall* = 90.5%) than non-fights (*specificity* = 90.0%), which, in a real scenario, would be a desirable property.

4.6.2. Efficiency and qualitative analysis

Throughout the paper, we have highlighted multiple times the importance of efficiency when attempting to detect violence. Besides demonstrating the efficiency of our proposal through the number of

Table 8

Confusion matrix of our best model for the validation set.

		Predicted	
		Fight	Non Fight
Actual	Fight	181	19
	Non Fight	20	180

Table 9

Runtime analysis. Results correspond to the average of 50 videos from RWF-2000 dataset. Each video is 5 s of length.

	OpenPose	Our model	Total
Time (s)	1.562	0.106	1.668
FPS	32	470	30

trainable parameters, we also measure its inference time to verify the real-time capabilities. Moreover, inference time not only includes the time taken by our model, but also the time needed to extract the skeletons by OpenPose. To measure inference time, we randomly selected 50 videos from the RWF-2000 dataset. From each 5-second video, originally consisting of 150 frames, we extract 50 frames as described in Section 4.2. We process each subsampled video independently, i.e., using a batch size of 1, and average the inference time, using an NVIDIA A100. As shown in Table 9, it takes 1.668 s to detect violence in each 5 s video, which effectively means that our proposal as a whole runs in real time. Furthermore, the cost of pose estimation represents 94% of the total time, whereas our proposal takes only 6% of the total time. Note that we obtained slightly higher FPS for OpenPose than the one reported in the official paper (Cao et al., 2021) (32 vs. 28 FPS) due to hardware differences.

Finally, in order to understand the limitations of our proposal and how it could be further improved, we analyze its failure cases. In particular, we present in Fig. 9 a summary of the videos where our model fails with the greatest confidence in the RWF-2000 validation set. We observe that false positives are mainly related to people adopting certain poses and movements that could be associated with violence in other circumstances. With respect to the false negatives, they mainly happen when violence is partially occluded.

4.7. Comparison with alternatives on Benchmark Datasets

Once we have our best model trained, we can compare its performance with the state-of-the-art approaches. First, we do so in the reference dataset: RWF-2000. As seen in Table 10, our model improves all the proposals by a large margin in terms of the trainable parameters. Moreover, our accuracy is not affected by the great reduction in the number of trainable parameters. A special mention should be made about *Spatio-Temporal Modeling* (Kang et al., 2021). In a point of our research, based on the fact that their approach uses very different components than other proposals, and thanks their public GitHub repository,¹ we tried to include part of their components in our architecture, with unsuccessful results. Nevertheless, we took this

¹ https://github.com/ahstarwab/Violence_Detection



Fig. 9. Failure cases of our best model in the RWF-2000 validation set. False positives are mainly related to people adopting certain poses and movements that could be associated with violence in other circumstances, while false negatives happen when violence is partially occluded.

Table 10

Comparison with state of the art in RWF-2000. ‘-’ indicates that the number was not published. ‘*’ retrained from scratch using our random seed following their public code and specifications.

Model	Year	Validation accuracy	Trainable parameters
Flow Gated Net (Cheng et al., 2021)	2020	87.30%	272 690
SPIL (Su et al., 2020)	2020	89.30%	-
SepConvLSTM (Islam et al., 2021)	2021	89.75%	333 057
Spatio-Temporal Modeling (Kang et al., 2021)	2021	92.00%	1 300 000
Spatio-Temporal Modeling (Kang et al., 2021) (retrained*)	2021	88.00%	1 300 000
Lightweight 2D+3D CNN (Wang et al., 2022)	2022	81.98%	352 000
Person detector + CNN (Choqueluque-Roman and Camara-Chavez, 2022)	2022	88.70%	-
U-Net + LSTM (Vijeikis et al., 2022)	2022	82.00%	3 457 219
Ours	2022	90.25%	62 583

opportunity to train their architecture from scratch, following their specifications but using our random seed, and got the accuracy reported in Table 10.

To demonstrate the generalization ability of our proposed architecture, we evaluate its performance on other public benchmarks. Since these datasets do not provide a validation set, and to get a robust sense of the performance, we evaluate the model using a five-fold cross-validation. Although this is the standard way to report results in many proposals of the state of the art, some of them use a train-test split which may lead to less robust but higher metrics. In terms of the results that we provide, we report three versions of our trained model. First, we evaluate the model trained on the RWF-2000 dataset. Second, we fine-tune this previous model during 5 epochs for each dataset. Finally, we train the model during 50 epochs. We believe that by showing the performance of these three versions, one can get a better understanding of the proposal’s generalization ability. The obtained accuracy is shown in Table 11.

In Table 11, we observe that the pre-trained model obtains better results in the Movies and Crowd datasets than in the Hockey dataset. This may be due to the nature of the videos, with the Hockey fights being too much different from the actions observed in surveillance cameras. On the other hand, the fine-tuned versions obtain great results with very short training, demonstrating a good ability to quickly adapt

to new settings and necessities. However, the accuracy in the Crowd dataset has not improved as much as the others, which is probably related to the difficulty of the dataset. After training for 50 epochs, we obtain comparable results to state-of-the-art alternatives, with far less trainable parameters. Although our main goal was to detect violence in surveillance scenarios, we have proved that our model can be used to detect violence in other domains as well.

5. Conclusion

Automatically detecting violence in surveillance videos is a critical task with huge potential benefits to society. Throughout this paper, we have presented an architecture able to efficiently and accurately perform this challenging task.

The proposed architecture has reached state-of-the-art performance with much fewer trainable parameters than other alternatives, which enables it to perform in real time. To achieve these results, we first extracted the most essential information from the videos: people skeletons, via a pose estimation model. Then, we decided to add another important source of information to the model: the dynamic temporal changes in the scene. Each pipeline appeared to be able to detect violence independently, but were even more powerful when combined. However, this combination has to be done with care, so as to get

Table 11

Comparison with state-of-the-art in other datasets. First three papers are classical approaches. The rest are based on DL. ‘-’ means that the number was not published.

Model	Year	Hockey	Movies	Crowd	Trainable parameters
STIP/MoSIFT (Bermejo Nievas et al., 2011)	2011	91.70%	89.50%	–	–
ViF (Hassner et al., 2012)	2012	82.90%	–	85.00%	–
ViF + OViF (Gao et al., 2016)	2016	87.50%	–	88.00%	–
3D CNN (Ding et al., 2014)	2019	96.00%	99.90%	98.00%	78 000 000
Efficient 3D CNN (Li et al., 2019)	2019	98.30%	100.00%	97.17%	7 400 000
Flow Gated Net (Cheng et al., 2021)	2020	98.00%	100.00%	88.80%	272 690
SPIIL (Su et al., 2020)	2020	96.80%	98.50%	94.50%	–
CNN + LSTM (Asad et al., 2019)	2020	98.80%	99.10%	97.10%	–
ConvLSTM (Sudhakaran and Lanz, 2017)	2017	97.10%	100.00%	94.57%	9 619 544
BiConvLSTM (Hanson et al., 2019)	2018	96.96%	100.00%	92.18%	–
SepConvLSTM (Islam et al., 2021)	2021	99.50%	100.00%	–	333 057
Spatio-Temporal Modeling (Kang et al., 2021)	2021	99.60%	100.00%	98.00%	1 300 000
Lightweight 2D+3D CNN (Wang et al., 2022)	2022	94.71%	100.00%	91.41%	352 000
Person detector + CNN (Choqueluque-Roman and Camara-Chavez, 2022)	2022	97.30%	–	–	–
U-Net + LSTM (Vijeikis et al., 2022)	2022	96.10%	99.50%	–	3 457 219
Ours (from RWF-2000)	2022	64.50%	70.00%	72.92%	62 583
Ours (fine-tuned)	2022	91.00%	95.50%	84.10%	62 583
Ours (full trained)	2022	94.50%	98.50%	94.30%	62 583

the most out of each pipeline. Hence, we have proposed a novel combination technique that works significantly better than previous alternatives.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Code has been published at <https://github.com/atmguille/Violence-Detection-With-Human-Skeletons>.

Acknowledgment

All authors approved the version of the manuscript to be published.

References

- Asad, M., Yang, Z., Khan, Z., Yang, J., He, X., 2019. Feature fusion based deep spatiotemporal model for violence detection in videos. In: Gedeon, T., Wong, K.W., Lee, M. (Eds.), *Neural Information Processing*. Springer International Publishing, Cham, pp. 405–417.
- Ben Mabrouk, A., Zagrouba, E., 2017. Spatio-temporal feature using optical flow based distribution for violence detection. *Pattern Recognit. Lett.* 92, 62–67.
- Bermejo Nievas, E., Deniz Suarez, O., Bueno García, G., Sukthankar, R., 2011. Violence detection in video using computer vision techniques. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (Eds.), *Computer Analysis of Images and Patterns*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 332–339.
- Blunsden, S., Fisher, R., 2009. The BEHAVE video dataset: ground truthed video for multi-person. *Ann. BMVA* 4.
- Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., Sheikh, Y., 2021. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (1), 172–186.
- Chen, M.-Y., Hauptmann, A., 2009. Mosift: Recognizing human actions in surveillance videos. *CMU-CS-09-161*.
- 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, IEEE, pp. 4183–4190.
- Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1800–1807.
- Choqueluque-Roman, D., Camara-Chavez, G., 2022. Weakly supervised violence detection in surveillance video. *Sensors* 22 (12).
- Dang, Q., Yin, J., Wang, B., Zheng, W., 2019. Deep learning based 2D human pose estimation: A survey. *Tsinghua Sci. Technol.* 24 (6), 663–676.
- Demarty, C.H., Penet, C., Soleymani, M., Gravier, G., 2014. VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation. *Multimedia Tools Appl.* 74.
- Ding, C., Fan, S., Zhu, M., Feng, W., Jia, B., 2014. Violence detection in video by using 3D convolutional neural networks. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., McMahan, R., Jerald, J., Zhang, H., Drucker, S.M., Kambhamettu, C., El Choubassi, M., Deng, Z., Carlson, M. (Eds.), *Advances in Visual Computing*. Springer International Publishing, Cham, pp. 551–558.
- Farneback, G., 2003. Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (Eds.), *Image Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 363–370.
- Gao, Y., Liu, H., Sun, X., Wang, C., Liu, Y., 2016. Violence detection using oriented Violent flows. *Image Vis. Comput.* 48–49, 37–41.
- Gonzalez, R.C., Woods, R., 2018. *Digital Image Processing*, fourth ed. Pearson, Upper Saddle River.
- Hanson, A., PNVR, K., Krishnagopal, S., Davis, L., 2019. Bidirectional convolutional LSTM for the detection of violence in videos. In: Leal-Taixé, L., Roth, S. (Eds.), *Computer Vision – ECCV 2018 Workshops*. Springer International Publishing, Cham, pp. 280–295.
- Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: *In Proc. of Fourth Alvey Vision Conference*. pp. 147–151.
- 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. pp. 1–6.
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T., 2017. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1647–1655.
- Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*. In: *Proceedings of Machine Learning Research*, vol. 37, PMLR, Lille, France, pp. 448–456.
- Islam, Z., Rukonuzzaman, M., Ahmed, R., Kabir, M.H., Farazi, M., 2021. Efficient two-stream network for violence detection using separable convolutional LSTM. In: 2021 International Joint Conference on Neural Networks. IJCNN, pp. 1–8.
- Jain, A., Vishwakarma, D.K., 2020. State-of-the-arts violence detection using ConvNets. In: 2020 International Conference on Communication and Signal Processing. ICCSP, pp. 0813–0817.
- Jenkins, N., 2016. *Video Surveillance: new installed base methodology yields revealing results*. White Paper, IHS Markit - Video Surveillance Group.
- Kang, M.S., Park, R.H., Park, H.M., 2021. Efficient spatio-temporal modeling methods for real-time violence recognition. *IEEE Access* 9, 76270–76285.
- Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. In: *International Conference on Learning Representations*.
- Laptev, I., 2005. On space-time interest points. *Int. J. Comput. Vis.* 64 (2), 107–123.
- Li, J., Jiang, X., Sun, T., Xu, K., 2019. Efficient violence detection using 3D convolutional neural networks. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance. AVSS, pp. 1–8.
- Lin, M., Chen, Q., Yan, S., 2014. Network in network. *CoRR* abs/1312.4400.
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60 (2), 91–110.
- Luvizon, D.C., Picard, D., Tabia, H., 2018. 2D/3D pose estimation and action recognition using multitask deep learning. In: *The IEEE Conference on Computer Vision and Pattern Recognition*. CVPR.
- NVIDIA, 2022. Optical flow to RGB image algorithm. https://github.com/NVIDIA/flownet2-pytorch/blob/master/utis/flow_utils.py#L72. (Accessed 6 May 2022).
- 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, e920.

- OpenCV, 2022. Histogram equalization tutorial. https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html. (Accessed 6 May 2022).
- Pedersen, J., Jensen, N., Lahrissi, J., Hansen, M., Staalbo, P., Wulff-Abramsson, A., Sander, M., 2019. Improving the accuracy of intelligent pose estimation systems through low level image processing operations. In: International Conference on Digital Image & Signal Processing. DISP19.
- Perez, M., Kot, A.C., Rocha, A., 2019. Detection of real-world fights in surveillance videos. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 2662–2666.
- Philippou, O., 2019. Video Surveillance Installed Base Report. White Paper, IHS Markit - Video Surveillance Group.
- Ribeiro, P.C., Audigier, R., Pham, Q.C., 2016. RIMOC, a feature to discriminate unstructured motions: Application to violence detection for video-surveillance. *Comput. Vis. Image Underst.* 144, 121–143, Individual and Group Activities in Video Event Analysis.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), *Medical Image Computing and Computer-Assisted Intervention. MICCAI 2015*, Springer International Publishing, Cham, pp. 234–241.
- Rota, P., Conci, N., Sebe, N., Reh, J.M., 2015. Real-life violent social interaction detection. In: 2015 IEEE International Conference on Image Processing. ICIP, pp. 3456–3460.
- Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c., 2015. 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*, MIT Press, Cambridge, MA, USA, pp. 802–810.
- Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: CVPR.
- Simonyan, K., Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1. NIPS '14*, MIT Press, Cambridge, MA, USA, pp. 568–576.
- Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition.. In: Bengio, Y., LeCun, Y. (Eds.), *ICLR*. pp. 1–14.
- Stergiou, A., Poppe, R., 2019. Analyzing human-human interactions: A survey. *Comput. Vis. Image Underst.* 188, 102799.
- Su, Y., Lin, G., Zhu, J., Wu, Q., 2020. Human interaction learning on 3D skeleton point clouds for video violence recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (Eds.), *Computer Vision. ECCV 2020*, Springer International Publishing, Cham, pp. 74–90.
- Sudhakaran, S., Lanz, O., 2017. 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.
- Sultani, W., Chen, C., Shah, M., 2018. Real-world anomaly detection in surveillance videos. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6479–6488.
- United Nations Office on Drugs and Crime, 2019. Global study on homicide.
- Vijeikis, R., Raudonis, V., Dervinis, G., 2022. Efficient violence detection in surveillance. *Sensors* 22 (6).
- Wang, W., Dong, S., Zou, K., Li, W., 2022. A lightweight network for violence detection. In: 2022 the 5th International Conference on Image and Graphics Processing. ICIGP, Association for Computing Machinery, New York, NY, USA, pp. 15–21.
- Yun, K., Honorio, J., Chattopadhyay, D., Berg, T.L., Samaras, D., 2012. Two-person interaction detection using body-pose features and multiple instance learning. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 28–35.