electronics

MDPI

Article

Deep Learning for Activity Recognition Using Audio and Video

Francisco Reinolds [1], Cristiana Neto [2,3] and José Machado [2,3,*]

[1] Department of Informatics, University of Minho, 4710-057 Braga, Portugal; sec@di.uminho.pt
[2] Algoritmi Research Center, Department of Informatics, University of Minho, 4710-057 Braga, Portugal; secretaria@algoritmi.uminho.pt
[3] LASI, Intelligent Systems Associate Laboratory, University of Minho, 4800-058 Guimarães, Portugal
* Correspondence: jmac@di.uminho.pt; Tel.: +351-253-604-430

Abstract: Neural networks have established themselves as powerhouses in what concerns several types of detection, ranging from human activities to their emotions. Several types of analysis exist, and the most popular and successful is video. However, there are other kinds of analysis, which, despite not being used as often, are still promising. In this article, a comparison between audio and video analysis is drawn in an attempt to classify violence detection in real-time streams. This study, which followed the CRISP-DM methodology, made use of several models available through PyTorch in order to test a diverse set of models and achieve robust results. The results obtained proved why video analysis has such prevalence, with the video classification handily outperforming its audio classification counterpart. Whilst the audio models attained on average 76% accuracy, video models secured average scores of 89%, showing a significant difference in performance. This study concluded that the applied methods are quite promising in detecting violence, using both audio and video.

Keywords: action recognition; violence detection; real-time video stream; neural networks; audio classifiers; video classifiers

Citation: Reinolds, F.; Neto, C.; Machado, J. Deep Learning for Activity Recognition in Real-Time Video Streams. *Electronics* **2022**, *11*, 782. https://doi.org/10.3390/electronics11050782

Academic Editors: Juan M. Corchado, Sascha Ossowski, Sara Rodriguez and Fernando De la Prieta

Received: 31 January 2022
Accepted: 25 February 2022
Published: 3 March 2022

1. Introduction

In an ever more connected world, smart cities are becoming ever more present in our society. In these smart cities, use cases wherein the innovations will benefit the inhabitants are also growing, improving their quality of life [1–3].

One of these areas is safety, in which machine learning (ML) models reveal high potential in real-time video-stream analysis in order to determine if violence exists in those videos [4]. These ML approaches concern the field of computer vision, a field responsible for transducingdigital images and videos, and being able to extract knowledge and understandable information from them, in order for them to be used in diverse contexts [5]. Depending on the nature of the data that needs to be assessed, there are several ML paradigms that can be used to train a model in order for it to better improve itself, and to correctly predict the outcome, for given data [6].

Some of the available alternatives to recognize actions in video streams are based on ML approaches, such as deep learning (DL), which grew in popularity in the last few years, as it was realized that it had massive potential in several applications that could benefit from having a machine recognizing diverse human actions [7,8]. When talking about DL, it is important to refer to neural networks (NN), which have the goal of mimicking the operation of the human brain, and more specifically the use of biological terms such as neurons and synapses. Usually they do not follow a group of instructions to resolve problems as conventional algorithmic approaches do. Neurons are powerful components that have a high potential in information storage, image recognition, and classification problems [9].

In this sense, the present work describes the research and analysis of the exploration of ML models that detect violence in video streams. Following the CRISP-DM (The CRoss Industry Standard Process for Data Mining) methodology, the application of ML models

to the collected data, allowed the comparison of the performance in terms of audio and video classification.

Regarding the structure of this article, after the present introduction, an exposition of some related works is given, followed by an explanation of the methodology, materials, and methods. The results are then shown and discussed. Finally, some conclusions are drawn.

2. Related Work

Over the last few years, several studies have been carried out in this field, and in this article we present a few of them considered relevant to the framework of this study.

In [10], the researchers highlighted the necessity of violence detecting models being efficient. As such, they proposed a hybrid feature "handcrafted/learned" framework. Handcrafted spatiotemporal features are able to achieve high accuracy of both appearance and motion; however, the extraction of such features is still troublesome for some applications.

The model first attempts to obtain the illustrative image from the video sequence taken as input for feature extraction, using Hough forest as a classifier. Afterwards, a 2D CNN is used to classify the image. The proposed approach was tested in a less-crowded scenario where it achieved accuracy rates ranging between 84% and 96%.

In [11], researchers assume that in a fighting scene, the motion blobs have a specific shape and position. Having been analyzed, the K largest are ultimately classified as either being violent or nonviolent.

To detect objects, this method used an ellipse detection method. To extract features, an algorithm to find the acceleration was used, and finally spatiotemporal features were used in an effort to classify the scenes. This method was tested with both crowded and less-crowded scenes, yielding a near 90% accuracy rate.

While this method was outperformed by SotA methods in terms of accuracy, it has expressively faster computation time, which does make it desirable for real-time applications.

In [12], researchers used independent networks to learn features specifically related to violence, such as blood, explosions, fights, etc. After that, in an effort to describe the violence while using such features, distinct SVM classifiers are trained for each of the concepts, and their results are later merged into a meta-classification.

To detect objects, movement detection and a temporal-robust features model were used, and to extract features, a bag of words method was used. This method was used in sparsely crowded scenarios, attaining a 96% accuracy rate.

3. Methodology, Materials, and Methods

Overall, this study was developed based on the CRISP-DM methodology, which, in short, consists of six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. In this article, the main aspects of this process are explained.

In order to carry out this study, several technologies were leveraged, the most relevant ones being Python and some of its libraries and PyTorch, an open-source ML library based on the Torch Library, used to create and train the models.

Making use of these technologies, it was possible to create the whole pipeline related to the creation, training, and evaluation of the models, described in the next subsection. In these training sessions, each of the chosen pretrained models that were used in each different stage of the pipeline were tested, using different settings, in order to understand which configuration better suited each model and yielded the best results.

In the training stage, each of the models were fed with data from which they try to extract knowledge, understanding what features correlate to the label; in this case, what features lead to a video stream to be considered violent or not. In the validation stage, each of the models were also fed with data; however, they do not learn from it, and, instead, they try to leverage their previously gained knowledge in order to predict if the video stream contains violence. In each training session, data pertaining to each of the models in the pipeline and their performance were saved in order to analyze how well the models performed. Such data encompasses accuracy percentage in each epoch, loss in each epoch,

and also the necessary information to compose a confusion matrix (CM—composed of true positives, true negatives, false positives, and false negatives) regarding the last epoch.

3.1. Pipeline

In order to provide a clearer understanding of the models exposed in this article, the pipeline of the research is presented in Figure 1. This figure describes the flow of information and actions taken in the training and evaluating of the models.

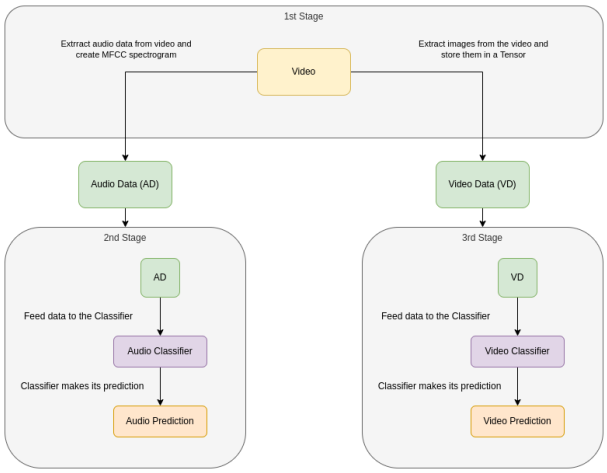

Figure 1. System pipeline chart.

First, the dataset was loaded, and in the first stage, a number of videos equivalent to the defined batch size (BS), a hyperparameter defined later in this paper, had their audio and video data extracted. After being extracted, an MFCC spectrogram was generated for each of the loaded videos, from their extracted audio data, while the video data was extracted and systematically loaded into a Tensor, so that it could be passed to the classifier.

Afterwards, both the classifiers received their respective data, which they processed and made a prediction for, being that the audio classifier completed its processing before the video one.

Once all the videos in the dataset were covered, the results were processed and saved for analysis.

3.2. Data

In the action recognition portion of the ML field, there is no shortage of action recognition datasets, e.g., UCF50 [13] and UCF101 [14], that feature, respectively, 50 and 101 categories of human performed actions. Just as there are datasets that feature nonviolent actions, there are also ones that features violent actions. Some examples of these are the Hockey Fight Detection Dataset [15] and Real Life Violence Situations Dataset [16], the last one containing a variety of violent scenes, ranging from punches and kicks to throws, among other.

For this study, it was deemed that the Hockey dataset would be unsuitable, as despite showing violent actions, these scenes would perhaps not generalize well to violent acts committed by people in day-to-day situations due to the inherent characteristics to that dataset (voluptuous hockey equipment, bats, the ice ring in which they play). Hence, the decision was made to use the Real Life Violence Situations Dataset; however, after running into some errors, a problem became clear. Most of the videos featured in the dataset contained no sound whatsoever. It was then decided that the dataset should be exploited to its full potential, which resulted in around 250 videos, but nearly all of them violent, as very few of the nonviolent videos had sound.

To mend this, scenes from videos of "city walking" were utilized. These videos are filmed by people with GoPros or similar cameras, that do not speak, and limit themselves

to just walking about different parts of different cities, and capturing footage. In these videos, people can be seen walking the streets, talking, eating, and engaging in every day actions. Consequently, "city walking" videos from several cities were downloaded and excerpts from them made in order to give the models a wide range of nonviolent scenes with audio.

In total, 500 videos with a duration of around 10 s each were collected, in which 250 featured violent scenes, and 250 nonviolent scenes. Figure 2 presents some examples of the scenes presented in the dataset. Besides the retrieval of public datasets featuring violent actions, the recording of videos specific to this project was also scheduled to take place, which featured the simulation of several violent scenarios. However, the recordings were involved in some bureaucracy, such as COVID-19 restrictions, which delayed their recording, making them unavailable for use in this article.

(**a**) (**b**)

Figure 2. Screenshots of videos in the dataset: (**a**) features violence, and (**b**) features day-to-day activities in a city center.

3.3. Data Preparation

In ML it is a common practice that some kind of preprocessing needs to be applied to the data that will be used to train the models, as some of its features may contribute negatively to the models' training.

An example of these negative impacts regards images, as they are typically stored as multidimensional arrays of values ranging from 0 to 255. One possible consequence of feeding those images as they are to a model would be what is often referred to as "exploding gradients", where a model's weights will vary dramatically from extremely high to extremely low values, possibly reaching NaN values, rendering the model useless.

After normalizing the data, typically, the data ranges from [0, 1], but it is also possible that the values are normalized to a range of $[-0.5, 0.5]$. In this case, the normalization was performed in such a way that the final values ranged from [0, 1], in order to improve the performance and training stability of the models.

3.4. Modeling

In each classifier, several models were tested in order to have more robust results from which to draw conclusions from. In this chapter, each of the models used in each of the classifiers will be explored, having their architecture detailed and their origins explained, leading to a better understanding of why they are suitable for the task at hand.

3.4.1. Audio

Audio analysis using ML techniques is based on extracting useful information from a sound track, analyzing, and then predicting over it. This useful information consists of the frequency and amplitude of the sound wave over a period of time. Two popular ways of extracting these features from an audio signal are as follows:

1. STFT: an audio waveform is converted to a spectrogram using STFT. This spectrogram displays the time–frequency changes as a 2D array of complex numbers that represent the magnitude and the phase [17].
2. MFCC: consists of calculating the power spectrum that gives the frequency spectrum to identify the present frequencies. To calculate the power spectrum, the Mel scale filter bank must be applied in order to extract the frequency bands. Afterwards, the log filter banks must be applied as well as the DCT coefficients to generate a compressed version of filter banks, and achieve the MFCC [17].

The audio classifiers being used in this study were both proposed in [18], along with a few other networks. The objective of this paper's authors was to present a residual learning framework that would ease the training of networks that are substantially deeper than those that had been used thus far. Their architecture was created having, mainly, VGG nets in mind, having up to eight times the depth (when comparing the 152 layer version), and attempted to achieve similar, if not better, results whilst having lower complexity.

The proposed NNs were highly successful, having earned the authors first place in the ILSVRC 2015 classification task, and having achieved better results than the VGG networks on the ImageNet classification dataset.

The first audio classifier being detailed is the Resnet18, whose architecture is possible to see in Table 1. This model's architecture starts with a convolutional layer with 7×7 kernel size, and is followed by the beginning of the skip connection. The input is then added to the output that is produced by the 3×3 max pool layer, and two pairs of convolutional layers with a kernel size of 3×3, having each 64 kernels. This part represents the first residual block, and five convolutional layers in total.

Table 1. Resnet18's architecture.

Layer Name	Output Size	Resnet18
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	1×1	7×7, average pool
fully connected	1000	1000 fully connective
softmax	1000	-

From there, the output of this residual block is passed on to three additional residual blocks, each with two pairs of convolutional layers. The convolutional layers of these residual blocks have a kernel size of 3×3 each, and 128 such filters. The convolutional layers in each residual block see their number of filters double, comparatively with the convolutional layers in the previous block, and the size of the output decreases by half. With these additional convolutional layers and the fully connected layer that is featured afterwards, the total amount of layers is brought up to 18, from where the model get its name.

The second network that served as audio classifier was the Resnet34, which has an architecture that closely resembles the Resnet18, as is possible to see in Table 2. This model's architecture begins by having almost the same setup for its first convolutional layer, and residual block following it, with the difference that this residual block features three pairs

of convolutional layers, and not two. Afterwards, it also follows the same trend, with the convolutional layers of each residual block seeing their number of filters double, when compared with the convolutional layers in the previous block, but in this model's case, each residual block does not necessarily have only two pairs of convolutional layers. More precisely, the second residual block features four pairs of convolutional layers, so eight in total. From there, the trend is kept, but with the difference that the third residual block features six pairs of convolutional layers, and the fourth, four pairs.

Table 2. Resnet34's architecture.

Layer Name	Output Size	Resnet34
conv1	112 × 112	7 × 7, 64, stride 2
conv2_x	56 × 56	3 × 3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
conv3_x	28 × 28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$
conv4_x	14 × 14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$
conv5_x	7 × 7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$
average pool	1 × 1	7 × 7, average pool
fully connected	1000	1000 fully connective
softmax	1000	-

3.4.2. Video

Video analysis presents different challenges compared to image analysis. When analyzing a video, the previous events that occurred in the sequence must be taken into account, to take into account the whole video, and not individual frames.

In order to do this, one cannot only take into account the two dimensions that are present in image analysis, but also a third dimension: the temporal dimension. To accommodate this new dimension, several strategies are available; however, the one being employed in this article revolves around 3D CNNs, which, as shown in [19], outperform 2D CNNs in challenging action recognition benchmarks, such as Sports-1M [20] and Kinetics [21].

The authors of [19] were inspired by these promising results, and introduced two new forms of spatiotemporal convolutions that can be viewed as middle grounds, between the extremes of 2D (spatial convolution) and full 3D (spatiotemporal convolution). Their first proposal was called mixed convolution (MC), and is explained in previously in the Resnet18.

The first proposal is the Resnet_MC18, whose architecture is possible to see in Table 3. It employs 3D convolutions solely in the early layers of the network, with 2D convolutions in the top layers, the rationale being that the motion modeling is a low/mid-level operation that can be implemented via 3D convolutions in the early layers of a network, and spatial reasoning over these mid-level motion features (implemented by 2D convolutions in the top layers) lead to accurate action recognition, and, in this article, to the accurate recognition of violent actions [19].

The second proposal by the authors of [19] is a "(2+1)D" convolutional block (Table 4), which explicitly factorizes 3D convolution into two separate and successive operations, a 2D spatial convolution and a 1D temporal convolution. The authors justify this "decomposition" with two reasons.

The first advantage is that having an additional nonlinear rectification between these two operations effectively doubles the number of nonlinearities, compared to a network using full 3D convolutions for the same number of parameters, thus rendering the model

capable of representing more complex functions. The second potential benefit is that the decomposition facilitates the optimization, yielding in practice both a lower training loss and a lower testing loss, hence effectively making the (2+1)D blocks easier to optimize.

Table 3. Resnet MC18's architecture.

Layer Name	Output Size	Resnet MC18
conv1	$16 \times 56 \times 56$	3×7, 64, stride 2
conv2_x	$16 \times 56 \times 56$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$16 \times 28 \times 28$	$\begin{bmatrix} 1 \times 3, 128 \\ 1 \times 3, 128 \end{bmatrix} \times 2$
conv3_x	$16 \times 14 \times 14$	$\begin{bmatrix} 1 \times 3, 256 \\ 1 \times 3, 256 \end{bmatrix} \times 2$
conv3_x	$16 \times 7 \times 7$	$\begin{bmatrix} 1 \times 3, 512 \\ 1 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 1$	7×7, average pool
fully connected	400	400 fully connective
softmax	400	-

Table 4. Resnet (2+1)D's architecture.

Layer Name	Output Size	Resnet (2+1)D
conv1	$16 \times 56 \times 56$	1×7, 45, stride 2
conv2	$16 \times 56 \times 56$	3×1, 64, stride 2
conv3_x	$16 \times 56 \times 56$	$\begin{bmatrix} 1 \times 3, 144 \\ 3 \times 1, 64 \\ 1 \times 3, 144 \\ 3 \times 1, 64 \end{bmatrix} \times 2$
conv4_x	$16 \times 16 \times 28$	$\begin{bmatrix} 1 \times 3, 230 \\ 3 \times 1, 128 \\ 1 \times 3, 230 \\ 3 \times 1, 128 \end{bmatrix} \times 1$
conv5_x	$8 \times 8 \times 28$	$\begin{bmatrix} 1 \times 3, 288 \\ 3 \times 1, 128 \\ 1 \times 3, 288 \\ 3 \times 1, 128 \end{bmatrix} \times 1$
conv6_x	$4 \times 14 \times 14$	$\begin{bmatrix} 1 \times 3, 460 \\ 3 \times 1, 256 \\ 1 \times 3, 460 \\ 3 \times 1, 256 \end{bmatrix} \times 1$
conv7_x	$4 \times 14 \times 14$	$\begin{bmatrix} 1 \times 3, 576 \\ 3 \times 1, 256 \\ 1 \times 3, 576 \\ 3 \times 1, 256 \end{bmatrix} \times 1$
conv8_x	$4 \times 7 \times 7$	$\begin{bmatrix} 1 \times 3, 921 \\ 3 \times 1, 512 \\ 1 \times 3, 921 \\ 3 \times 1, 512 \end{bmatrix} \times 1$
conv9_x	$2 \times 7 \times 7$	$\begin{bmatrix} 1 \times 3, 1152 \\ 3 \times 1, 512 \\ 1 \times 3, 1152 \\ 3 \times 1, 512 \end{bmatrix} \times 1$
average pool	$1 \times 1 \times 1$	7×7, average pool
fully connected	400	400 fully connective
softmax	400	-

3.5. Evaluation

In order evaluate the different classifiers' performances, different metrics were considered and analyzed, namely, the following:

1. Average training accuracy (ATA)—average accuracy rate achieved by a classifier with the training set.
2. Average training loss (ATL)—average loss achieved by a classifier with the training set.
3. Average validation accuracy (AVA)—average accuracy rate achieved by a classifier with the validation set.
4. Average validation loss (AVL)—average loss achieved by a classifier with the validation set.

4. Results and Discussion

After having completed the training of every classifier, in diverse conditions, altering the different hyperparameters, a set of results was achieved. These results encompass close to 200 h of training, divided into eight training sessions, in which the four different models were used in their respective classifiers. These results comprise dozens of graphs and text files that range from confusion matrices that plot the last epoch of the training session to line graphs that plot the evolution of the accuracy and loss rates. Having this data available is crucial in order to determine what approach is working best, and why something is not performing to its fullest extent. It is important to note two crucial hyperparameters, namely, BS and learning rate (LR). BS can be defined as the number of samples that normally pass through the neural network at one time. On the other hand, LR controls how much it is necessary to change the model in response to the estimated error each time the model weights are updated [22]. All the other options that were used to configure the model had their values set to the default options, described on PyTorch's documentation of the models.

In the following subsections, the results achieved by the two types of classifiers tested in this article will detailed and analyzed. They will be evaluated under several perspectives, and connections between the different settings related to the hyperparameters and the performance of the classifiers are attempted to draw. Firstly, the average values of each of the models used for the classifier in question will be analyzed, regardless of the hyperparameters set. Afterwards, the hyperparameters will be taken into account, in order to obtain a more comprehensive look at the results, in a better effort to understand if any conclusions can be drawn.

4.1. Audio

The theory behind the classifying of audio arises from the sound characteristics that are typically present in violent acts, such as screaming, battering of surfaces, breaking of items, and so forth. It is believed that the classifier will be able to identify in which scenarios violent acts are occurring simply based on the audio feed.

In Table 5, a general look into each of the models' performances can be seen in terms of ATA, ATL, AVA, and AVL.

Table 5. General audio classifier results.

Model Name	ATA	ATL	AVA	AVL
Resnet18	86.75	0.21	70.5	0.81
Resnet34	85.5	0.21	66.75	0.84

Here, we can see that, in a general sense, the models are extremely similar, which is to be expected as they were conceived by the same people and are based on the same architecture, the only difference being that the Resnet34 has more layers than the Resnet18 one. All of their values are very similar, with perhaps the AVL being the one that has the most significant difference, but even in that case, it is a 3.75% difference, which is very similar.

Table 6 presents the audio classifier results by BS. When comparing the results from Table 5 with those from Table 6, it is possible to realize a trend in which both models performed better with a BS of 7 then 5, mainly in average accuracy. This is particularly notable with the Resnet34, which scored 71% of average accuracy in the validation set, almost 5% more than its AVL, as seen in Table 5.

Table 6. Audio classifier results by BS.

Model Name	BS	ATA	ATL	AVA	AVL
Resnet18	5	85.5	0.25	70.5	0.69
Resnet18	7	88	0.17	70.5	0.94
Resnet34	5	82.5	0.27	65.5	0.84
Resnet34	7	88.5	0.14	71	0.85

When looking at Table 6, it is possible to verify that a wider range of results are present. When comparing the models' performance with the different BSs, the results are somewhat inconclusive. On one hand, the ATA somewhat improves, being accompanied by a drop in the ATL. However, when it comes to the average accuracy in the validation set, in the Resnet18 model's case, there was no improvement gained from increasing the BS from 5 to 7, even having had a worse performance, as the average loss in the validation set increased. In the Resnet34's case, the average accuracy in the validation set improved by almost 6%; however, the average loss barely increased.

Taking a look at the models' results from an LR perspective, the obtained results are presented in Table 7. In this table, it is possible to verify how the LR affected the models' performance. When looking at the performance over the training set, both models performed similarly, scoring close to 90% accuracy with an LR of 1×10^{-4} , and having their accuracy drop to around 80% with an LR of 1×10^{-4}, and with this drop, an increase of the average loss ensued. Furthermore, the models' trend in the training set stayed the same in the validation set as well. With an increase in LR, the models' performance decreased, verified in both average accuracy, with a decrease, and an increase in the average loss.

Table 7. Audio classifier results by learning rate (LR).

Model Name	LR	ATA	ATL	AVA	AVL
Resnet18	1×10^{-4}	90	0.11	73.5	0.59
Resnet18	1×10^{-3}	83.5	0.31	67.5	1.04
Resnet34	1×10^{-4}	90	0.09	69	0.73
Resnet34	1×10^{-3}	81	0.32	67.5	0.96

It is possible, once again, to notice a trend when comparing the results from Tables 5 and 7, which shows us that both the models benefit from using an LR of 1×10^{-4}, which it is possible to see by both the increase in average accuracy in the training and validation set, and the accompanying loss reduction.

In a general sense, the audio classifiers performed as expected, displaying a solid performance in general, but not performing as well as their counterparts. The best result was achieved by the Resnet18 model, with an LR of 1×10^{-4} and a BS of 7, with it having achieved an average accuracy of 91% and an average loss of 0.08 on the training set, and seeing those values worsen to 76% and 0.51 in the validation set. Despite having dropped 15%, the average accuracy in the validation set still falls within the expected values.

4.2. Video

Regarding video classifiers, the expectations shift, as essentially all models used in state-of-the-art papers are video classifiers, so these kind of models are expected to perform at an extremely high level. It is natural that video classifiers are the ones that perform

better, as video features are more descriptive and correlate more to violent actions such as punching, beating, and similar actions, than the sound that these actions produce.

Looking at Table 8, the distinction between the performance of the audio and video classifiers becomes clear. In the training set, the video classifiers outscored the audio classifiers on average by around 5%. Their performance becomes even more impressive when it comes to the average accuracy on the validation set, in which the video classifiers outscored their audio counterparts by a whopping 15%.

Table 8. General video classifier results.

Model Name	ATA	ATL	AVA	AVL
Resnet(2+1)D	90.5	0.08	86.25	0.24
Resnet_MC18	90.5	0.085	85.75	0.25

The general results from the video classifiers showed promise, so it is interesting to understand how the different hyperparameters interact with each one of the models that were chosen as video classifiers and their respective performance.

Firstly, looking at their behavior from the perspective of the BS parameter, the obtained results are presented in Table 9 .

Table 9. Video classifier results by BS.

Model Name	BS	ATA	ATL	AVA	AVL
Resnet(2+1)D	5	90	0.09	86.5	0.23
Resnet(2+1)D	7	91	0.07	86	0.26
Resnet_MC18	5	90	0.1	85	0.25
Resnet_MC18	7	91	0.07	86.5	0.24

These results were quite surprising, due to their unexpected consistency, which seems to indicate that the BS of 5 and 7 are adequate values with which to train the models. Across both models, regardless of which BS was used, the results remained remarkably close, with a maximum difference between the models' average accuracy in the validation set of just 1.5%.

Moving to Table 10, it is possible to verify the LR's influence in the models' results. In this table, much more diverse results are present, which is quite apparent, but not right away. In regards to the average accuracy in the training set, neither the BS nor the LR had too much of an effect, with every model scoring between 90% and 91% and an accompanying low loss. When it comes to the average accuracy in the validation set, however, the scenario changes dramatically. Despite not having an extreme fluctuation, a close to 7% difference in the average accuracy between the models now exists, ranging from 82.5% to 89%. In this regard, it is clear to see that the LR bears a lot more relevance to the models' performance than the BS, at least in the used values.

Table 10. Video classifier results by LR.

Model Name	LR	ATA	ATL	AVA	AVL
Resnet(2+1)D	1×10^{-4}	90	0.08	89	0.13
Resnet(2+1)D	1×10^{-3}	91	0.08	83.5	0.36
Resnet_MC18	1×10^{-4}	90.5	0.09	89	0.12
Resnet_MC18	1×10^{-3}	90.5	0.08	82.5	0.37

The best results, regarding the average accuracy in the validation set, were achieved by both the Resnet(2+1)D and Resnet_MC18, with both of the models attaining scores of 89% accuracy in the set. This impressive accuracy was obtained, as can be expected from the above explanation, with an LR of 1×10^{-4}. Comparing these results with the audio classifiers, it becomes clear why video classifiers are used in virtually every state-of-the-art paper. Even when taking into account all the different testing scenarios, in no circumstance did the audio classifiers score close to their video counterparts.

5. Conclusions

After all the tests and analyses were carried out, it was finally possible to conclude which model and type of classifier performed better, by looking at Table 11. In this table, it is possible to see what model, by type of classifier, scored the highest average accuracy rates over the validation set. It is important to note that the "-" in the BS column of the video classifiers means that with either a BS of 5 or 7, both of these models scored the same average accuracy of 89% over the validation set, so the "-" signifies that, regardless of the BS, the models performed the same when having an LR of 1×10^{-4}.

Table 11. Best average accuracy on the validation set by classifier type.

Classifier Type	Model Name	BS	LR	AVA
Audio	Resnet18	7	1×10^{-4}	76
Video	Resnet(2+1)D	-	1×10^{-4}	89
Video	Resnet_MC18	-	1×10^{-4}	89

Looking at Table 11, it is clear that all the models outperformed the expectations that were previously set for them. The audio classifier, despite being the lowest scorer of the tested classifiers, still achieved a respectable 76% of average accuracy, which, in and of itself, is a good achievement. When we look, however, at the other classifiers, it is possible to truly perceive the potential that these models have for action recognition, and violence detection in particular.

This research proved that the application of models with neural networks for the detection of violence is quite feasible since all of these models scored, on average, between 85% and 89% accuracy on the validation set, which are remarkable results. Compared to existing schemes, these results are slightly inferior to those that are found in state-of-the-art papers, which have slightly superior accuracy rates of 90%. However, with a more robust dataset, these values could surely be improved, and these results could match those of state-of-the-art featured models.

Concerning future work, two main paths could be taken. First, the expansion of the used dataset can lead to a more robustly trained model, which would naturally yield better results, since it would be more comprehensively trained. Next, the implementation of the model in a real-time environment would be advantageous to improve this research. In addition, this study has a novel contribution, as it represents a basis for an even more robust study to be developed by the authors. This future study includes the detection of violence using early fusion and late fusion methods (multimodal classification).

Author Contributions: Conceptualization, F.R. and C.N.; methodology, F.R., C.N. and J.M.; software, F.R.; validation, F.R., C.N. and J.M.; formal analysis, J.M.; investigation, F.R.; resources, F.R. and C.N.; data curation, F.R.; writing—original draft preparation, F.R.; writing—review and editing, C.N.; supervision, J.M.; project administration, J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and the project "Integrated and Innovative Solutions for the well-being of people in complex urban centers" within the Project Scope NORTE-01-0145-FEDER-000086. C.N. thank the FCT—Fundação para a Ciência e Tecnologia for the grant 2021.06507.BD.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.kaggle.com/mohamedmustafa/real-life-violence-situations-dataset, accessed on 1 January 2022.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ATA	Average training accuracy
ATL	Average training loss
AVA	Average validation accuracy
AVL	Average validation loss
BS	Batch size
CM	Confusion matrix
CNN	Convolutional neural network
DL	Deep learning
LR	Learning rate
MC	Mixed convolution
ML	Machine learning
MFCC	Mel frequency cepstral coefficient

References

1. Mohammadi, M.; Al-Fuqaha, A. Enabling cognitive smart cities using big data and machine learning: Approaches and challenges. *IEEE Commun. Mag.* **2018**, *56*, 94–101. [CrossRef]
2. Chen, X.; Qi, L.; Yang, Y.; Luo, Q.; Postolache, O.; Tang, J.; Wu, H. Video-based detection infrastructure enhancement for automated ship recognition and behavior analysis. *J. Adv. Transp.* **2020**, *2020*, 7194342. [CrossRef]
3. Wang, Z.; Hou, Y.; Jiang, K.; Zhang, C.; Dou, W.; Huang, Z.; Guo, Y. A survey on human behavior recognition using smartphone-based ultrasonic signal. *IEEE Access* **2019**, *7*, 100581–100604. [CrossRef]
4. Santos, F.; Durães, D.; Marcondes, F.S.; Hammerschmidt, N.; Lange, S.; Machado, J.; Novais, P. In-car violence detection based on the audio signal. In *International Conference on Intelligent Data Engineering and Automated Learning*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 437–445.
5. Jesus, T.; Duarte, J.; Ferreira, D.; Durães, D.; Marcondes, F.; Santos, F.; Gomes, M.; Novais, P.; Gonçalves, F.; Fonseca, J.; et al. Review of trends in automatic human activity recognition using synthetic audio-visual data. In *International Conference on Intelligent Data Engineering and Automated Learning*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 549–560.
6. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2018.
7. Khurana, R.; Kushwaha, A.K.S. Deep Learning Approaches for Human Activity Recognition in Video Surveillance-A Survey. In Proceedings of the 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018; pp. 542–544.
8. Santos, F.A.O.; Durães, D.; Marcondes, F.S.; Gomes, M.; Gonçalves, F.; Fonseca, J.; Wingbermühle, J.; Machado, J.; Novais, P. Modelling a Deep Learning Framework for Recognition of Human Actions on Video. In *WorldCIST (1)*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 104–112.
9. Neto, C.; Brito, M.; Peixoto, H.; Lopes, V.; Abelha, A.; Machado, J. Prediction of length of stay for stroke patients using artificial neural networks. In *World Conference on Information Systems and Technologies*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 212–221.
10. Serrano, I.; Deniz, O.; Espinosa-Aranda, J.L.; Bueno, G. Fight recognition in video using hough forests and 2D convolutional neural network. *IEEE Trans. Image Process.* **2018**, *27*, 4787–4797. [CrossRef] [PubMed]
11. Gracia, I.S.; Suarez, O.D.; Garcia, G.B.; Kim, T.K. Fast fight detection. *PLoS ONE* **2015**, *10*, e0120448.
12. Peixoto, B.M.; Avila, S.; Dias, Z.; Rocha, A. Breaking down violence: A deep-learning strategy to model and classify violence in videos. In Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg, Germany, 27–30 August 2018; pp. 1–7.
13. Reddy, K.K.; Shah, M. Recognizing 50 human action categories of web videos. *Mach. Vis. Appl.* **2013**, *24*, 971–981. [CrossRef]
14. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
15. Abdali, A.; Al-Tuma, R. Robust Real-Time Violence Detection in Video Using CNN And LSTM. In Proceedings of the 2019 2nd Scientific Conference of Computer Sciences (SCCS), Baghdad, Iraq, 27–28 March 2019; pp. 104–108. [CrossRef]
16. Soliman, M.M.; Kamal, M.H.; Nashed, M.A.E.M.; Mostafa, Y.M.; Chawky, B.S.; Khattab, D. Violence recognition from videos using deep learning techniques. In Proceedings of the 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 8–10 December 2019; pp. 80–85.
17. Purwins, H.; Li, B.; Virtanen, T.; Schlüter, J.; Chang, S.Y.; Sainath, T. Deep learning for audio signal processing. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 206–219. [CrossRef]
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
19. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6450–6459.

20. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
21. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The kinetics human action video dataset. *arXiv* **2017**, arXiv:1705.06950.
22. Brownlee, J. Understand the Impact of Learning Rate on Neural Network Performance. 2020. Available online: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/ (accessed on 1 January 2022).