Simple Approach for Violence Detection in Real-Time Videos Using Pose Estimation With Azimuthal Displacement and Centroid Distance as Features

Felipe Boris De Moura Partika, violencedetector.org, USA*

ABSTRACT

Detecting violence in real-time videos is not an easy task even for the most advanced deep learning architectures, considering the subtle details of human behavior that differentiate an ordinary from a violent action. Even with the advances of deep learning, human activity recognition (HAR) in videos can only be achieved at a huge computational cost, most of the time also requiring special hardware for reaching an acceptable accuracy. The author presents in this paper a novice method for violence detection, a sub-area of HAR, which outperforms in speed and accuracy the state of the art methods. The method is based on features extracted from the Pose estimator method OpenPose. These features are then transformed into more representative elements in the context of violence detection, which are then submitted to a LSTM neural network to learn how to identify violence. This work was inspired by the violencedetector.org, the first open-source project for violence detection in real-time videos.

KEYWORDS

3DCNN, Deep Neural Networks, Human Action Recognition, LSTM, OpenPose, Spatio-Temporal Features

INTRODUCTION

In this paper, the author demonstrates a novel method for violence detection in real-time videos, allowing machines to better interpret human actions in videos, therefore being able to differentiate between ordinary and violent behavior.

This differentiation is very complex since there is no single parameter that indicates the existence of violence individually; actually, a complex combination of several parameters and its variations is required (Ullah et al., 2017), such as position of individuals’ parts, contact points between them, and how each part moves along the time (Serpush & Rezaei, 2020). Additionally, human activity recognition (HAR) also requires multiple frames analysis, which increases exponentially the amount of parameters to be processed (Sharif et al., 2019).

Currently, the field of computer vision is dominated by convolutional neural networks (CNNs) (Varior et al., 2016; Zeiler & Fergus, 2013); the same has proven its worth becoming the basic
construction block for many deep learning architectures. The convolutions turn two-dimensions images into more abstract elements called activation maps, which hold the learned abstract representations of the images. Therefore, CNNs are capable of learning image details such as contours, edges and patterns based on the pixel intensity variations of the image (Almaadeed et al., 2021), and keeping immune to small transformations in the input image (e.g., translation, scaling, skewing, and distortion) (LeCun et al., 1989). Unfortunately, convolutions normally result in a large footprint over an expensive computational cost.

Since CNNs were initially designed for 2D image classification, for dealing with multiple frames in HAR, one additional temporal dimension had to be added to this architecture. This new architecture is known as 3DCNN. This additional dimension enhances even more the computational cost (Aktı et al., 2020) as described before.

In order to address this increase in the computational cost, the author’s method translates the real-time data in which the neural network can learn the subtleness that indicates violence in a simpler and lighter approach, based on more representative features than the pixel intensity variation patterns (Javidani & Mahmoudi-Aznaveh, 2018).

For this purpose, the author’s method implies the use of OpenPose (Cao et al., 2019) for localizing the anatomical key points in human bodies by what OpenPose refers to as part affinity fields (PAFs). PAFs learn to associate body parts with individuals in the image, and this process achieves high accuracy and real-time performance (Kim & Lee, 2020). OpenPose (Cao et al., 2019) defines each individual through 18 key points (Figure 1).

Those relationships between joint points are transformed into more representative features (please see the next section) to feed a long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) network, which will then learn these features over a temporal series extracted from the video. Figure 2 illustrates the processing flow.

The LSTM (Varior et al., 2016) is an important component of this method, since it is a special type of recurrent neural network (Schuster & Paliwal, 1997; Sherstinsky, 2020). However, differently from feed forward networks (e.g., CNNs), this architecture includes recursive loops that retain information obtained in the previous time slots (Arif et al., 2019). This means that the output is conditional on the context of the input sequence, not only what has been presented as input, allowing dealing with temporal distributed information, and having a nature of remembering information for a period of time as default behavior. This capability allows the network to learn the subtle differences in the displacements of the key points along the time series (Nunez et al., 2018; Sudhakaran & Lanz, 2017).

Figure 1. Process of pose estimation using OpenPose (Cao et al., 2019); (left) original image with the key points and connections; (right) only the key points and connections extracted from the first image.
AZIMUTHAL DISPLACEMENT AND CENTROID DISTANCE

As Figure 3 shows, the key points from frame $t$ are compared with frame $t+1$ yielding a displacement gradient vector; this vector is then converted to azimuthal coordinates (Figure 4). This transformation turns this feature into location and scale invariant, since the displacement of the key point is considered, instead of its position. The magnitude and angle of the gradient vector indicate displacement speed and direction of the movement, respectively, which proved to be more meaningful than the raw coordinates of the key points for action recognition, namely violence detection, in the case of this study (Kim et al., 2020).

Figure 2. The network architecture is composed by Openpose (Cao et al., 2019), then the features are transformed into more representative information which feeds a long short-term memory network.

Figure 3. Openpose (Cao et al., 2019) key points extraction in two consecutive frames (left and center); right demonstrates how the displacement is calculated for three key points for these frames. The gradient vectors are defined by the points 1a to 2b, 2a to 2b, and 3a to 3b.

Figure 4. Polar coordinates sphere, in the case of this study the distances to the plane z, are the same, therefore $\theta$ is constant.

\[
\nabla d(t) = \begin{bmatrix} X_{t+1} - X_t \\ Y_{t+1} - Y_t \end{bmatrix} = (r, \phi)
\]
Another representative aspect of human interaction is the distance between the individuals (Muralikrishna et al., 2020); the features extracted from the distance are very relevant for the violence detection. Figure 5 demonstrates how this feature is acquired.

For each individual in a particular frame \( t \), the distance of all key points is calculated against the centroid function \( c \) of the nearest individual. This calculation is performed for all frames along the time series, yielding 18 features per individual.

The gradient vector of the azimuthal displacement also yields 18 features, both are 36 features for each individual at a particular frame.

The final result is a feature vector of 36 azimuthal displacement elements and 36 coordinates (x and y axes) elements for the centroid distances. These 72 features are stacked through the time series (Tasmin et al., 2021) (the author’s experiment utilized 64 frames) yielding a 72x63 feature matrix per individual (Figure 6).

**NETWORK ARCHITECTURE**

The author’s architecture uses two sequences of one dimension convolution, bath normalization, maximum pooling and dropout to process the extracted features from OpenPose (Cao et al., 2019) translated into azimuthal displacement and centroid distances (Figure 7).

The one-dimension convolution provides a very effective feature extraction method over a low computational cost, since the convolution happens only over the time dimension (Kiranyaz et al., 2020).

Assuming the function \( O \) as the order of complexity of the convolution, and having a NxN image, with a KxK kernel, the complexity comparison between one and two dimensions is as follows:

**Figure 5.** Centroid distance calculation, where the point C is the other individual centroid. The distances between the key points A, B, D and the centroid C are the three centroid distance features.

**Figure 6.** Every two frames yields one line in the feature matrix, combining the azimuthal displacement with centroid distance. Every individual yields one matrix.
These two sequences of convolutions, normalizations and maximum poolings allow the second layer to operate in a higher level of abstraction, yielding a more compact representation of the most relevant information (Krizhevsky et al., 2012).

This compact representation then feeds the LSTM network, exploring its ability to capture long-term temporal dependencies without being so affected by common problems from other recurrent neural networks, such as vanishing gradients (Mahasseni & Todorovic, 2016; Yu et al., 2019).

The last layer in this architecture is a fully connected one, which performs the final classification putting the LSTM output into the two final classes, namely violence and nonviolence (Caetano, 2020; Peixoto et al., 2019).

EVALUATION

Dataset
The author used the Real Life Violence Situations Dataset (Soliman et al., 2019) for this evaluation; it contains 1000 videos of violence and 1000 videos of nonviolence collected from YouTube.

The creators identified a shortage in available datasets related to violence between individuals; this was the motivation for this dataset.

Test Methodology
In order to ensure test equivalence, the author used the same randomized set of 20% of the dataset for both network architectures; this sample was not part of the training. The author defined the frame resolution being 224x224 pixels, which is also the default for MobileNet avoiding unnecessary resizing.

The tests were performed at the same computer, running Tensorflow 2.5 over a RTX 2080ti GPU. The timings were measured during the inference of both architectures, but, for their network, the author...
also added the OpenPose (Cao et al., 2019) feature extraction time plus the feature transformation time, using the azimuthal displacement and centroid distances method proposed here.

OpenPose (Cao et al., 2019) can work with different networks; for these tests the author utilized the MobileNet v2.

Both network architectures receive sequences of images as input, but the author’s method works internally with a deeper level of detail, since OpenPose (Cao et al., 2019) extracts individuals’ features in each frame. Then, these transformed features feed the author’s proposed LSTM architecture.

Comparison Criteria

For their comparison, the author utilized the 3DCNN architecture (Pijackova & Gotthans, 2021), since several authors consider it as a state of the art method for HAR. As the author discussed in the previous section, 3DCNN is computationally expensive, due to its iterative processing nature along the axes $x$ and $y$ for 2D convolution, and axes $x$, $y$, and $z$, in case of 3D convolution (Figure 8).

RESULTS

Table 1 demonstrates the test results: The author’s method achieved superior precision, recall, and F1-score in comparison with 3DCNN, considering the same validation dataset sample the researcher described in the subsection Test Methodology.

As Table 2 shows, for the same tests, the author’s method achieved an incredible speed, more than 14 times faster than the 3DCNN, with almost half of the size considering the lighter model. In order to keep a fair comparison, the author is considering the model size and speed as the sum of OpenPose (Cao et al., 2019) (2.1MB/0.0076s) and his network architecture (6.8MB/0.0019s), since both are required to perform HAR.

Figure 8. The 2D convolution uses a 2D kernel to generate the activation maps (left), while the 3D convolution uses a 3D kernel to generate the activation maps (right)

<table>
<thead>
<tr>
<th></th>
<th>3DCNN Precision</th>
<th>3DCNN Recal</th>
<th>3DCNN F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro average</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>Weighted average</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>The Author’s Precision</th>
<th>The Author’s Recal</th>
<th>The Author’s F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro average</td>
<td>0.90</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td>Weighted average</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
</tbody>
</table>
The author expected the superior speed, due to the simplicity of this method, compared with the high computational cost described in the previous subsection for 3DCNN (Pham et al., 2018).

**VIOLENCEDETECTOR.ORG**

Technology utilization for violence prevention is the objective of violencedetector.org (Partika, 2020), which is the first open source project for violence detection in real time videos.

The work the author presented in this paper directly benefits this project. Considering its lightweight, this method can run under limited resources, for example edge devices.

Those devices can monitor violence and trigger local responses to avoid situations or even alert authorities. Their simplicity may allow the development of smart security devices at a low cost, generating a new device ecosystem for violence prevention.

**CONCLUSION**

Detecting violence in videos is not an easy task, even for the most advanced deep learning architectures. In this paper, the author presented a novel method based on the transformation of OpenPose (Cao et al., 2019) features, which has proven to be very effective for violence detection in real-time videos.

This method yields a compact footprint and superior accuracy, compared with other state of the art techniques for the same purposes, with incredible gains in speed confirmed by a realistic benchmark (subsection Results) (Perez et al., 2019).

Ultimately, the author has open-sourced this work as part of violencedetector.org (Partika, 2020), the first open source project for violence detection in real-time videos, which aims to democratize this technology, allowing the development of new smart security devices and enabling violence prevention at low cost.

**ACKNOWLEDGMENT**

The author acknowledge the work done by the authors of OpenPose (Cao et al., 2019), as well as Real Life Violence Situations dataset (Soliman et al., 2019).

**FUNDING AGENCY**

The Open Access Processing fee for this article was covered in full by the authors of this article.
REFERENCES


