Crowd Violence Detection from Video Footage

Konstantinos Gkountakos
Information Technologies Institute
CERTH
Thessaloniki, Greece
gountakos@iti.gr

Konstantinos Ioannidis
Information Technologies Institute
CERTH
Thessaloniki, Greece
kioannid@iti.gr

Theodora Tsikrika
Information Technologies Institute
CERTH
Thessaloniki, Greece
theodora.tsikrika@iti.gr

Stefanos Vrochidis
Information Technologies Institute
CERTH
Thessaloniki, Greece
stefanos@iti.gr

Ioannis Kompatsiaris
Information Technologies Institute
CERTH
Thessaloniki, Greece
ikom@iti.gr

***Abstract*—Surveillance systems currently deploy a variety of devices that can capture visual content (such as CCTV, body-worn cameras, and smartphone cameras), thus rendering the monitoring of the video footage obtained from multiple such devices a complex task. This becomes especially challenging when monitoring social events that involve large crowds, particularly when there is a risk of crowd violence. This paper presents and demonstrates a crowd violence detection system that can process, analyze, and alert the potential stakeholders when violence-related content is identified in crowd-based video footage. Based on deep neural networks, the proposed end-to-end framework utilizes a 3D Convolutional Neural Network (CNN) to deal with the (near) real-time analysis of video streams and video files for crowd violence detection. The framework is trained, evaluated, and demonstrated using the most recent dataset related to crowd-violence, namely the Violent Flows dataset. The presented framework is provided as a standalone application for desktop environments and can analyze video streams and video files.**

***Index Terms*—crowd analysis, violence detection, (near) real-time, 3D-CNN**

I. Introduction

Surveillance systems are widely used for numerous purposes, including, but not limited to, crime prevention, monitoring, and evidence discovery. The main characteristic of surveillance systems is the need for operational personnel to continuously monitor the available video footage. Given that, nowadays, various devices that can capture visual content are deployed for such purposes, including Closed-Circuit TeleVision (CCTV), body-worn cameras, and smartphone cameras, the monitoring of the video footage obtained from multiple such devices has become a complex task that requires the availability of high numbers of human resources. Moreover, as this is a highly demanding task, it may occasionally result in human operators missing the detection of events of interest.

Particularly challenging to monitor are social events (e.g., football matches and music festivals) which are becoming more and more crowded, and where there is often a risk of crowd violence erupting. An Artificial Intelligence (AI) based framework that could continuously monitor multiple sources of video footage and adequately inform the security personnel, in (near) real-time, whether activities and events of interest (such as crowd violence) are detected would be particularly advantageous, and efforts towards this have attracted significant interest in recent years.

In this work, we showcase a beta version of the crowd violence detection framework[1] introduced in [1] that reports higher accuracy against several state-of-the-art approaches [2]–[5] and outperforms the deep-learning-based [6]–[8] approaches. The proposed framework incorporates a 3D Convolutional Neural Network (3D-CNN) [6] architecture that can process video footage in (near) real-time. More specifically, the presented framework can analyze video files and video streams by processing mini-batches of 16-length frames at every step. Regarding the real-time processing, the proposed framework consists of four sub-components that run in a loop, as illustrated in Figure 1.

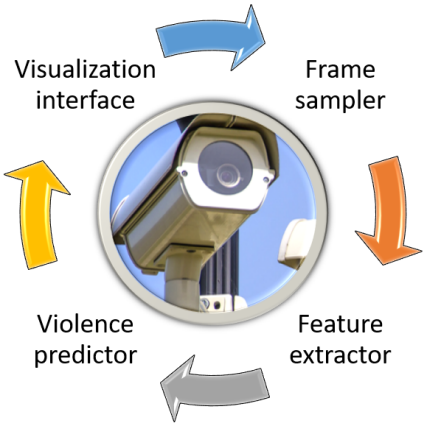

Fig. 1. Illustration of the crowd violence detection framework sub-components in a loop: first, the frames for further processing are sampled, next their features are extracted, followed by the predictor that estimates a violence level, and ending with the visualization in the graphical user interface.

[1]Demo available at: https://m4d.iti.gr/crowd-analysis-tool/

The four components comprising the proposed crowd violence detection framework are the following:

- the *sampler*, which selects the frames for the next processing step;
- the *feature extractor*, which encodes the visual input of frames to visual features;
- the *predictor*, which analyzes the features and estimates a violence score; and
- the *application interface*, which visualizes the results and notifies the end-user.

This work is organized as follows. Section II describes the crowd violence detection framework and its sub-components. Section III demonstrates the framework while highlighting the information flow and the use cases. Section IV concludes by discussing future directions for improving the framework.

II. Crowd Violence Detection Framework

Based on four sub-components, the proposed crowd violence detection framework executes a sequential loop whereby various sub-modules collaborate to detect violence-related scenes from video footage, namely video files and video streams. In this section, these sub-components are illustrated as designed for the processing of such video footage.

A. Frame sampler

The frame sampler is a sub-component that is enabled only when the analysis is performed using video streams, since all frames are processed in the case of video files. The frame sampler sub-component aims to extract 16-length sequential frames in order to prepare the input for the feature extractor. For the case of video files, all the video's frames are extracted using a sliding window of size equal to 16 and step equal to 16; on the other hand, a queue of frames is considered for the case of video streams. The queue size is equal to 32 as we want to retain the last second of the visual stream, assuming a frame rate equal to 30 frames per second (fps).

Every time a video frame is available from the stream, it is incorporated in the queue. When the next sub-component in the loop requests the next mini-batch for processing, a sampling of 16 from the 32 frames of the queue is forwarded when the queue is full; otherwise, mini-batches of 16 frames are sampled. Following this approach, the 16 frames of the mini-batch comprise the representative samples of approximately the last second of streamed content, when considering a frame rate equal to 30 fps.

Figure 2 illustrates the frame sampler's various functionalities for the video file and the video stream processing. On the top, the sliding window approach is presented. On the bottom, the queue approach is illustrated for the two different ways of sampling: (i) when the queue is full (depicted in yellow), and (ii) with sequential sampling (depicted in orange).

B. Feature extractor

The feature extractor is the core sub-component of the proposed framework. In this step, the video frames are encoded to visual features in order to be used for the prediction of the crowd violence level. To this end, a 3D-CNN ResNet [9] architecture trained using the Violent Flows [10] dataset is deployed. For each mini-batch of 16 frames, the tool extracts a feature vector of 2048 length from the fully connected layer before the binary classifier. Then, the next step estimates the crowd violence level for each feature vector.

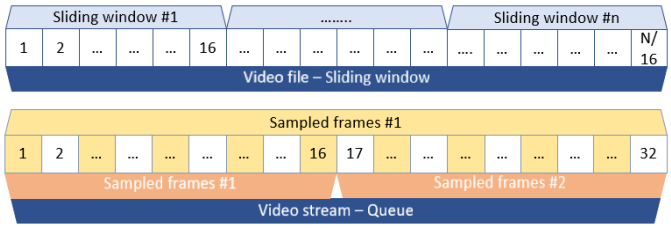

Fig. 2. Illustration of different frame sampling options. On the top, the sliding window dealing with video files. On the bottom, the two modifications of the queue for sampling video streams.

The extracted features have been learned using the Violent Flows dataset's training set consisting of 246 videos with an overall duration of 14.76 minutes that have a balanced split into the two categories of crowd violence/non-violence. Similarly to the works reported in the recent literature, we follow the five folds cross-validation approach using the four subsets for the training and the remaining one for the testing. The training process was performed using an NVidia RTX 2080ti GPU with 11GB memory for 100 epochs and lasted approximately 4 hours.

C. Predictor

The objective of the predictor is to estimate the crowd violence severity level for each mini-batch of 16 frames. This process could be integrated into the feature extractor; nonetheless, a real application prerequisites (near) real-time processing, which enables a detached prediction for each feature vector. The crowd violence level is estimated with a prediction score from 0.0 to 1.0 using a sigmoid function [11] as activation function on the last node of the architecture. The values closer to 0 denote non-violent scenes, while values close to 1 denote crowd violence-related content.

D. Application interface

Once the first mini-batch analysis of the 16 frames is completed, the results are visualized onto the user interface. For illustration purposes, an appropriate colour palette has been identified and incorporated, as depicted in Figure 3, in order to quantize the severity level representation. In particular, the red color indicates the prediction level of crowd violence from 0.8 to 1.0, while the green color illustrates predictions of violence levels from 0.0 to 0.2. The other colors depict the rest of the ranges ([0.2, 0.4), [0.4, 0.6), [0.6, 0.8]), respectively.

III. Demonstration

This section includes a detailed illustration of the framework's core functionalities and the user interface, while some implementation details are also presented.

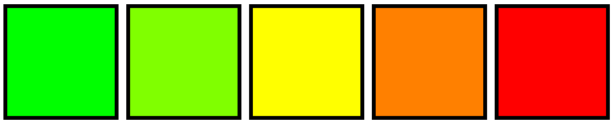

Fig. 3. Colour palette used for illustrating the different crowd violence levels. The red color depicts a higher level of crowd violence, while the green color indicates non-violence.

A. Implementation details

The user interface developed for the crowd violence detection framework involves three basic functionalities: (i) the source definition, (ii) the selection of video footage, and (iii) the video player, as illustrated in Figure 4.

The source definition is depicted on top and allows to define the video files or video streams to be processed. The user can browse to a folder that contains video files for analysis, or type a URL that corresponds to a video stream so as to initiate its processing. The framework can process video streams transferred through Real-Time Streaming Protocol (RTSP), the most common solution for such applications.

Once the source definition is completed, the user shall press the ”Analyze” button to initiate the analysis of the corresponding source. After completing the analysis, the list of the analyzed videos is presented below the source definition box for the case of video files. For video streams, the list remains empty as our application does not support the storing of the video streams. For both types of sources, the video player is responsible for playing the analyzed videos and for visualizing the results onto the stream footage.

In Figures 5 and 6, the results of processing video files depicting non-violent and crowd violence-related scenes are shown, respectively. In Figure 5, the video shows a crowd walking on the road during a celebration day. The video is colorized with a green frame that indicates non-violent content, while the word ‘Prediction’ is depicted over the content, followed by the score of the crowd violence level, which is, in this case, equal to 0%. Figure 6 illustrates the analysis of a crowd that participates in a riot at a stadium’s stands after the end of a football game. The frame is colorized with a red bounding box, while the crowd violence level score is estimated at 97%.

As it is challenging to illustrate and depict on paper how the proposed framework works in (near) real-time, we provide three samples that represent in a temporal manner the continuous monitoring performed by the framework. Figure 7 presents continuously sampled frames, every one second, for three different videos. On the top, there is a crowd violence-related sample that clearly illustrates the crowd violence level predicted between 97% and 100%. In the middle, a non-violence related example is displayed, with the framework accurately predicting a score equal to 0% for each of the four frames. Finally, on the bottom, a non-violence related example is depicted that reports score close to 0% for the four indicative frames.

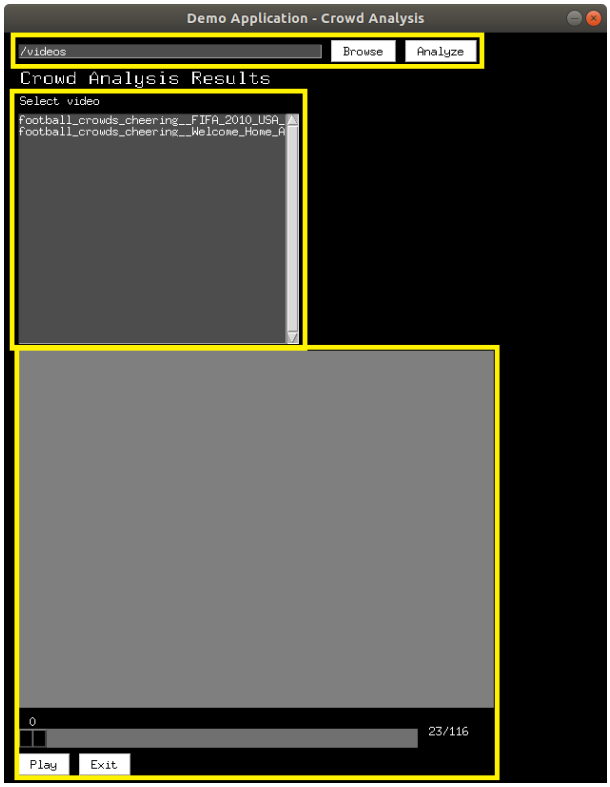

Fig. 4. The user interface of the crowd violence detection application. The source definition, the selection of video footage, and the video player are depicted using yellow-colored boxes.

B. Implementation details

For developing, training, evaluating, and demonstrating the proposed crowd violence detection framework, specific configurations/libraries are utilized to support its deployment. The operating system is Linux based Ubuntu 18.04; the programming language used is python 3 [12] with additional libraries including, but not limited to, PyTorch [13] and python-OpenCV, while for the graphical user interface, the python-tkinder library is used.

IV. Conclusions

This work presented a crowd violence detection framework and the associated application. The framework is focused on crowd-centred scenes and aims to detect and estimate the violence level in (near) real-time while processing video files or video streams. In addition, the various components comprising the framework, as well the user interface of the application were presented. Future steps include improvements towards reducing the latency when video streams are processed and enabling support for other platforms, while the simultaneous processing of multiple cameras will also be investigated.

Acknowledgment

This work was supported by the Horizon 2020 projects CONNEXIONs (H2020-786731) and PREVISION (H2020-833115) funded by the European Commission.

Fig. 5. A non-violence related example of the proposed framework user interface after completing the analysis of three videos. On the top, a list of the analyzed videos is depicted. On the bottom, the outcomes of the analysis for the corresponding selected video are shown.

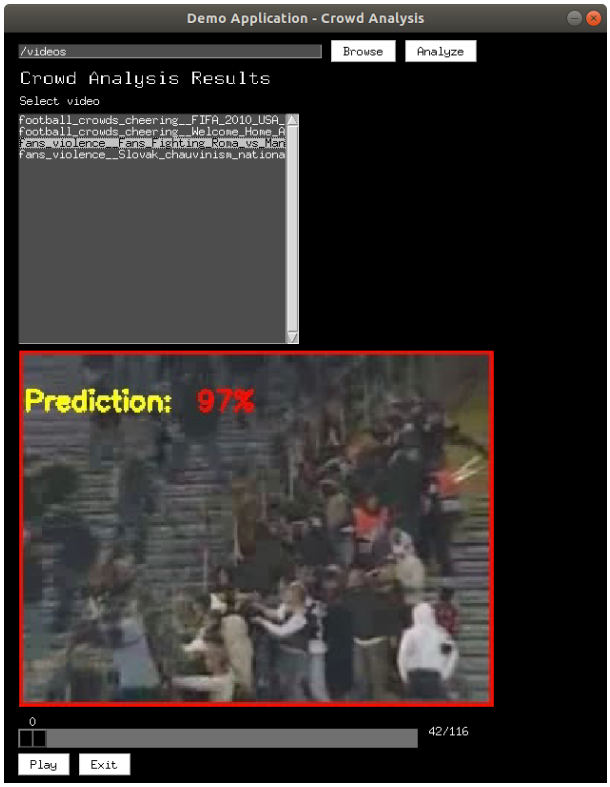

Fig. 6. A crowd violence related example in the user interface after completing the analysis of three videos. On the top, a list of the analyzed videos is depicted. On the bottom, the outcomes of the analysis for the corresponding selected video are shown.

Fig. 7. Three indicative samples for temporal illustration of the proposed framework. On the top, a crowd violence-related example is presented, while on the middle and the bottom, non-violence related examples are shown.

REFERENCES

[1] K. Gkountakos, K. Ioannidis, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "A crowd analysis framework for detecting violence scenes," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020, pp. 276–280.

[2] P. Bilinski and F. Bremond, "Human violence recognition and detection in surveillance videos," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016, pp. 30–36.

[3] T. Zhang, W. Jia, C. Gong, J. Sun, and X. Song, "Semi-supervised dictionary learning via local sparse constraints for violence detection," *Pattern recognition letters*, vol. 107, pp. 98–104, 2018.

[4] T. Zhang, W. Jia, X. He, and J. Yang, "Discriminative dictionary learning with motion weber local descriptor for violence detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 696–709, 2016.

[5] P. Zhou, Q. Ding, H. Luo, and X. Hou, "Violence detection in surveillance video using low-level features," *PLoS one*, vol. 13, no. 10, p. e0203668, 2018.

[6] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, "Violence detection using spatiotemporal features with 3d convolutional neural network," *Sensors*, vol. 19, no. 11, p. 2472, 2019.

[7] A. Hanson, K. Pnvr, S. Krishnagopal, and L. Davis, "Bidirectional convolutional lstm for the detection of violence in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[8] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *arXiv preprint arXiv:1510.01553*, 2015.

[9] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[10] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2012, pp. 1–6.

[11] Y. Ito, "Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory," *Neural Networks*, vol. 4, no. 3, pp. 385–394, 1991.

[12] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani,

S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf