Supplementary Materials
Not only Look, but also Listen: Learning Multimodal Violence Detection under Weak Supervision

Peng Wu, Jing Liu*, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang

School of Artificial Intelligence, Xidian University, Xi'an, China
xdwupeng@gmail.com,yjsun@stu.xidian.edu.cn,
{neouma,shiyujiaaaa,shaofangtao96,15191737495,zwyang97}@163.com

1 Video Collection

Our dataset is collected from both movies and YouTube (in-the-wild scenes). To generate high-quality video clips from movies, we first search multiple types of movies, e.g., action movies, military movies, blood movies, literary movies, romantic movies, cartoons, etc. Then we invite eight annotators having high levels of computer expertise to watch movies, randomly cut sections of different length that contain clear violent or non-violent events and make video-level labels. Finally, annotators perform two checks to correct wrong videos and remove ill-suited videos annotated by others. We also collect in-the-wild videos by YouTube. We first search and download a mass of video candidates using text search queries. In order to prevent violence detection systems from discriminating violence based on the background of scenarios rather than occurrences, we specifically collect large amounts of non-violent videos whose background is consistent with that of violent videos. After that, we remove videos which fall into any of the following conditions: soundless, only containing background sounds, ambiguity, blurry scenes, and containing very little violence.

Besides, we randomly split our dataset into training and test sets, repeat this process multiple times, and keep the best one with suitable proportion.

2 Dataset Comparisons

We list violence types of common datasets in Table 1.

3 Similarity Computation Functions

Two other versions of f are defined as follows,

* Corresponding author: neouma@163.com

Table 1. Comparisons of violence types.

Dataset	Violence types
Hockey	Fighting
Movie	Fighting
Violent-Flows	Fighting
CCTV-Fights	Fighting
VSD	Fighting, fire, weapon, car chase, gunshot, explosion, gory scene,and scream
UCF-Crime	Abuse, arrest, arson, assault, accident, explosion, fighting, robbery, and shooting
XD-Violence (Ours)	Abuse, car accident, explosion, fighting, riot, and shooting

Table 2. AP comparison of different similarity computation functions on the XD-Violence dataset.

Function	AP (%)
Version 1	78.64
Version 2	79.04
Version 3	77.37

[Version 2]

$$f(x_i, x_j) = \frac{(wx)_i^T (w^{'}x)_j}{\|(wx)_i\|_2 \cdot \|(w^{'}x)_j\|_2} \tag{1}$$

[Version 3]

$$f(x_i, x_j) = exp\left(x_i \cdot x_j - max(x_i \cdot X)\right) \tag{2}$$

From Table 2, we observe that three versions achieve similar performance, and the Version 2 outperforms other two versions by a narrow margin since the version 2 has learnable weights and can learn better similarity.

4 The Effect of Length of Sampling

Untrimmed videos have large variance in length, from a few seconds to several hours. On the one hand, we need to process the entire video at once because we only have video-level labels. On the other hand, it is impractical to directly process a very long video due to GPU memory constraints. We use a simple yet effective sampling. Consider a video V and corresponding features X^F, we process the entire video if its feature length $T^{'}$ is less than the pre-defined $\mathit{\Gamma}$ length necessary to meet the GPU bandwidth. Otherwise, we uniformly extract a segment of length $\mathit{\Gamma}$ from X^F to represent the whole video. In this paper, we set $\mathit{\Gamma}$ as 200 because this is a good tradeoff between accuracy and computation burden.

Results from Table 3 show that with the increase of threshold, the run time of each training epoch increases, but the performance increases firstly and then

Table 3. Performance comparisons with respect to length of sampling on the XD-Violence dataset.

Threshold	AP (%)	Run Time /s
100	78.30	69
200	78.64	71
300	78.04	72
400	77.94	75
500	78.32	78

Table 4. Perclass AP comparison of different multimodal cues.

Class	Audio	RGB	Audio+RGB
Fighting	85.04	85.97	88.02
Shooting	71.53	83.51	90.30
Riot	65.07	70.54	76.42
Abuse	76.48	90.10	83.43
Car Accident	65.21	68.83	74.89
Explosion	68.36	84.04	86.17

fluctuates slightly. Therefore, we choose 200 as the pre-defined Γ in this paper due to the good tradeoff between accuracy and computational costs.

5 Investigating Perclass Performance with Different Multimodal Cues

Following [4], we show comparison results in Table 4. As for per-class breakdown, we observe that 1) compared single signal, Audio+RGB improves the performance of perclass (except for the abuse, possible reason is that the number of abuse samples is small); 2) adding audio gets clear performance boosts for some classes, e.g., Shoot, Riot, Car Accident.

6 Comparisons with State-of-the-Arts

We compare our method with several baselines on the UCF-Crime dataset, and show the results in Table 5, respectively. It is obvious that our method can outperform current state-of-the-art methods.

We also show the PRC on the XD-Violence dataset as Fig. 1. As Fig. 1 shows, the curve of our method completely encloses others, which means our method is superior to the competitors at various thresholds. Besides, online detection and RGB-only do not obtain the maximum area under curve due to lacks of contextual information and audio information, respectively.

Table 5. AUC comparisons on the UCF-Crime dataset.

Method	AUC (%)
SVM baseline	50.00
Hasan *et al.* [1]	50.60
Lu *et al.* [2]	65.51
Sultani *et al.* [3]	75.51
Ours	82.44

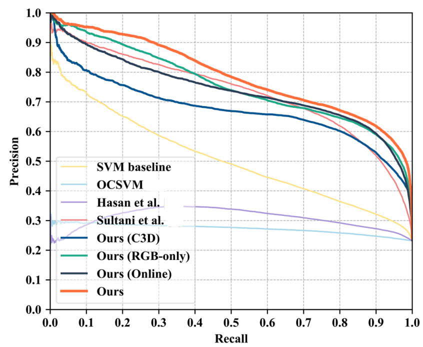

Fig. 1. PRC on the XD-Violence dataset.

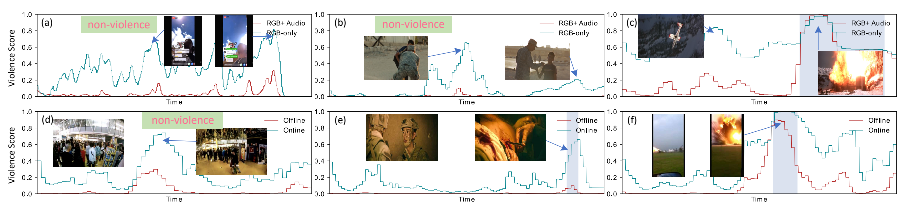

Fig. 2. Qualitative results of our method on test videos. The 1^{st} row shows qualitative comparisons between Audio+RGB and RGB-only input. The 2^{nd} row shows qualitative comparisons between offline detection and online detection. Colored window shows the ground truth of violent regions. [Best viewed in color.]

7 More Qualitative Results

We present several qualitative examples in Fig. 2. As we can see, RGB-only input produces many false alarms when: scene keeps changing in the live video (a), playing football looks like a fight (b), and an airplane plummet through the sky (c). For the false alarm in (d), we find the possible cause is that there is a mirror on the ceiling, which confuses our method. We argue that the missed alarm of offline detection in (e) is caused by over-smoothing, which usually occurs in GCN. Specifically, the violent features are smoothed by non-violent features since violent segment accounts for a little part of the entire video.

References

1. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 733–742 (2016)
2. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2720–2727 (2013)
3. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6479–6488 (2018)
4. Zhu, Y., Newsam, S.: Motion-aware feature for improved video anomaly detection. In: Proceedings of the British Machine Vision Conference (BMVC) (2019)