Weakly-Supervised Violence Detection in Movies with Audio and Video Based Co-training

Jian Lin[1] and Weiqiang Wang[1,2,*]

[1] Graduate University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{jlin,wqwang}@jdl.ac.cn

Abstract. In this work, we present a novel method to detect violent shots in movies. The detection process is split into two views——the audio and video views. From the audio-view, a weakly-supervised method is exploited to improve the classification performance. And from the video-view, we use a classifier to detect violent shots. Finally, the auditory and visual classifiers are combined in a co-training way. The experimental results on several movies with violent contents preliminarily show the effectiveness of our method.

Keywords: Violence, Weakly-supervised, pLSA, Audio, Video, Co-training.

1 Introduction

Nowadays, the flourishing movie industry generates thousands of movies each year. However, not all the plots are suitable for children to watch, especially violent contents. It is useful to find an effective way to automatically detect violent contents in movies. Some approaches have already been proposed to address this problem. Mitchell *et al.* hided violent scenes using video data hiding techniques [1], while Nuno *et al.* did information filtering in movie databases [2]. Datta *et al.* [3] exploited the accelerate motion vector to detect fist fighting, kicking. Nam *el al.* [4] located violent scenes by detecting flame and blood. Cheng *et al.* [5] proposed a hierarchical approach to recognizing gunshots, explosions, and car-braking.

In our method, a video sequence is first split into a set of shots. Then the classification of these shots is performed from two independent views——the audio-view and the video-view. From the audio-view, the audio segment of a specific shot is classified into violence or non-violence categories in a weakly-supervised way. From the video-view, the corresponding video segment is classified by a model which combines three common video violent events (motion, flame & explosion and blood). Finally, the outputs of the two views complement each other through a co-training way.

2 Audio Violence Detection

pLSA has been extensively applied in text categorization [6], language modeling [7], etc., and has shown its good performance. However, it has not been well exploited in

[*] Corresponding author.

P. Muneesawang et al. (Eds.): PCM 2009, LNCS 5879, pp. 930–935, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the audio violence detection. In our method, the pLSA algorithm is modified to locate audio violence.

For a collection of audio documents $D=\{d_1,d_2,\cdots,d_n\}$ given, i.e., the set of audio segments, they are first split into audio clips about 1 second long, and each clip is represented by a feature vector composed of several low-level features, including spectrum power, brightness, bandwidth [8], pitch, MFCC, spectrum flux, high zero cross rate ratio (ZCR) and harmonicity prominence [9] etc., due to their success in speech recognition and audio classification [10]. Then all feature vectors are clustered to get a set of clustering centers, which is denoted as the audio vocabulary $A=\{a_1,a_2,\cdots,a_m\}$. In our work, 20 clustering centers are chosen to achieve a good performance, and the k-means algorithm with *L1* distance is used during clustering. Finally, each clip is represented by a_i $(i\in 1,2,...m)$, and then the audio documents are represented by A .

In the training phase, an aspect model is built to associate a latent class variable $z\in Z=\{1,0\}$ (corresponding to violence and non-violence categories) with each observation of $d_i\in D$ and $a_j\in A$. The joint probability model is defined by the mixture :

$$P(d_i,a_j)=P(d_i)P(a_j\mid d_i),\quad P(a_j\mid d_i)=\sum_{z\in Z}p(a_j\mid z)p(z\mid d_i)\ , \tag{1}$$

The Expectation Maximization (EM) is used to fit the model. Detailed description of the E-step and M-step can be seen in [6][7].

3 Video Violence Detection

3.1 Motion Intensity and Complexity

Motion intensity and complexity are used to detect areas with fast tempo. The motion intensity of the ith macroblock (MB) in the k th frame is defined as $MI_k(i)=x(i)^2+y(i)^2$, where (x_i , y_i) is the motion vector of the ith MB of the k th frame. The total motion intensity of the k th frame is calculated by $MI_k=\sum_{i=0}^{n-1}MI_k(i)$, where n is the number of MBs in the k th frame.

We use motion complexity to depict the dispersion degree of motion vectors. The motion complexity of the k th frame is defined as $MC_k=\sum_{i=0}^{n-1}\mid MI_k(i)-MI_k{}^{\mathrm{ave}}\mid$, where $MI_k{}^{\mathrm{ave}}$ is the average motion intensity of the k th frame which is calculated by $MI_k{}^{\mathrm{ave}}=\frac{1}{n}\sum_{i=0}^{n-1}MI_k(i)$. Finally, the motion measure of the k th frame is obtained by the product of the above two factors $V_k=MI_k\times MC_k$. If V_k exceeds a specific threshold value, the k th frame is classified as violence.

3.2 Flame and Explosion

Flame & explosion is another common violent event. In order to detect flame & explosion, one frame is divided into m regions, each region is 4×4 pixel size. All the regions are divided into yellow-tone area and non-yellow-tone area. All adjacent yellow-tone areas comprise a big yellow-tone area. Suppose there are N_y big yellow-tone areas, and each is represented as A_j (j=1,..., N_y). All the non-yellow-tone areas are represented as B. The motion complexity is utilized to ensure the big yellow-tone area is in flame & explosion. The difference between $MC_k(A_j)$ and $MC_k(B)$ is calculated as $\Delta D_k = MC_k(A_j) - MC_k(B)$. If ΔD_k is larger than a specific threshold value, the big yellow-tone area A_j is classified as a flame & explosion area, then the k th frame is classified as violence.

3.3 Blood

The same as 3.2, if the dominant color of the ith (i=1,..., m) region is red, the region is classified as a red-tone area. All adjacent red-tone areas comprise a big red-tone area. Suppose there are N_R big red-tone areas, and each is represented as C_j (j=1,..., N_R). Blood always comes up with fast motion. So in the next step, the motion intensity in the blood area is utilized to determine whether the area is blood or not. The motion intensity in the jth blood area is calculated as $M_k(C_j) = \frac{1}{l}\sum_{i=1}^{l} MI_k(i)$, where l is the number of regions in the big red-tone area C_j, $MI_k(i)$ is the motion intensity in the ith region. The average motion intensity in the frame is calculated as $M^{ave} = \frac{1}{m}\sum_{i=1}^{m} MI_k(i)$. The difference of local motion intensity $M_k(C_j)$ and average motion intensity M^{ave} in the frame is compared as $\Delta M_k = M_k(C_j) - M^{ave}$. If the ΔM_k in the big red-tone area is larger than a predefined value, the area is classified as a blood area, then the k th frame is classified as violence.

3.4 Video Violence Detection

All three violent events mentioned above are integrated to find violence scenes. V_k, ΔD_k and ΔM_k are all normalized to 0-1 scale, and respectively defined as $\tilde{V}_k$, $\Delta\tilde{D}_k$ and $\Delta\tilde{M}_k$. Then we feed them into a linear weighted model with proper weights ω_1, ω_2 and ω_3 ($\omega_1+\omega_2+\omega_3$=1). The final evaluation value of the k th frame is calculated as $e_k = \omega_1\tilde{V}_k + \omega_2\Delta\tilde{D}_k + \omega_3\Delta\tilde{M}_k$. And on the shot level, the evaluation value of the m th shot which contains n frames is defined as $E_m = (\sum_{k=0}^{n-1} e_k)/n$. If E_m exceeds a specified threshold value E_{thr}, the m th shot is labeled as 'violence'.

4 Co-training

Co-training algorithm works by generating several classifiers which are trained on a few labeled data. Co-training was applied in statistical parsing [11], reference resolution [12], etc. In our work, co-training is utilized in violence detection. The important aspect consists in that two classifiers (C_1 , C_2) are respectively built in two different views of audio (X_1) and video (X_2). The utilization of co-training is described in Fig. 1.

```
0. Given:
- A set L of labeled training examples
- A set U of unlabeled examples
- Classifiers  and .
1. Create a pool U' of examples by choosing P random examples from U
2. Loop for iterations until the unlabeled data is over:
    2.1 Use L to individually train the classifiers  that considers only the audio portion of the movie
    2.2 Use L to individually train the classifiers  that considers only the video portion of the movie
    2.3 For each classifier , select $p$ most confidently examples and $n$ most un-confidently examples from U'
    2.4 Add these labeled examples to L
    2.5 Refill U' with examples from U, to keep U' at a constant size of $P$ examples.
```

Fig. 1. General bootstrapping process in co-training

In our work, L is a formally defined data set which contains a collection of labeled shots randomly selected from several movies. The threshold value E_{thr} in video violence detection is firstly obtained from the labeled violent shots in L, then automatically adjusted by the labeled violent shots during the co-training process. For the m th shot, the threshold value E_{thr} is calculated by the 10 latest labeled violent shots:

$$E_{thr} = 1.2 \times \min\{E_{m-l}\}, l \in \{1,2,3,...,10\} . \tag{2}$$

where E_{m-l} (l=1,2,3,…,10) is the video violence evaluation value of the $m-l$ th shot.

5 Experimental Results

The performance of the proposed approach is tested on four movies. The detailed information is listed in Table 1. Precision *P*, recall *R* and F1-measure F_1 are 3 measurements for our system.

To verify the validity of the proposed approach, a comparison between SVM and our method is presented in Table 2. All three measurements of our method are much better than those generated by the supervised way SVM.

Table 1. Experimental dataset information

No.	Movie Title	Violence Type	Duration	Testing Set
1	The Terminator (1984)	Gun Shots/Explosion/Murder	108 min	53 min
2	Kill Bill: Vol1 (2003)	Fighting	111min	36 min
3	The Rock (1996)	Gun Shots/Explosion/fighting	136 min	55 min
4	Hot Fuzz (2007)	Fighting/Gun Shots	121 min	54 min
5	DOA Dead or Alive (2006)	Fighting	82min	72min

Table 2. Results of our method and SVM

Movie No.	SVM			Our Method		
	Precision	**Recall**	**F1-measure**	**Precision**	**Recall**	**F1-measure**
1	0.6246	0.8253	0.7111	0.8507	0.9585	0.9058
2	0.6833	0.8836	0.7706	0.8429	0.9321	0.8853
3	0.6027	0.8017	0.6881	0.8156	0.9105	0.8604
4	0.6155	0.8059	0.6979	0.8321	0.8975	0.8636
5	0.6723	0.8796	0.7621	0.8496	0.9026	0.8738

Table 3 lists the average weights of ω_1 , ω_2 and ω_3 in the three violent shot types (fast tempo, explosion or blood type). It can be seen that the three weights alter according to specific types. So our scheme in video part to deal with different kinds of shots is effective.

Table 3. Weight statistics in video modal

Violent Shot Type		fast tempo	flame & explosion	blood
Weight	ω_1	0.956	0.157	0.103
	ω_2	0.043	0.815	0.075
	ω_3	0.001	0.028	0.822

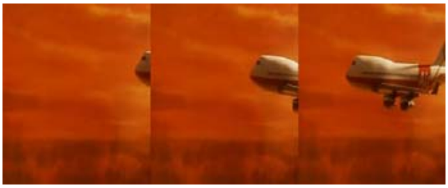

Fig. 2. Sunset glow scene

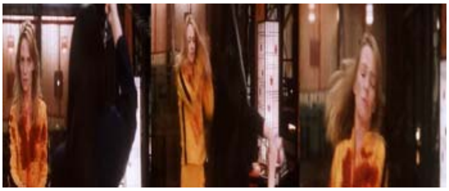

Fig. 3. Fight scene

In Fig. 2, the background is red-tone, but it does not move faster than the other area, so it will not be regarded as violence area. In Fig. 3, the red-tone area on the women's breast has faster motion than the other background area. So the red-tone area is classified into blood area, and the scene is regarded as violence area.

6 Conclusions

In this paper, a novel violent shot detection scheme is presented. From the audio-view, the violent shot is detected with the modified pLSA. And from the video-view, the violent shot is detected with the motion, flame & explosion and blood analysis. Finally, the proposed audio and video classifiers are combined into co-training. Experimental results show that the proposed method is effective in violent shots detection.

Acknowledgments. This work was supported in part by National Key Technologies R&D Program under Grant 2006BAH02A24-2 and by National Natural Science Foundation of China under Grant 60873087.

References

1. Swanson, M.D., Zhu, B., Tewfik, A.H.: Data Hiding for Video-in-Video. In: IEEE International Conference on Image Processing, vol. 2, pp. 676–679 (1997)
2. Vasconcelos, N., Lippman, A.: Towards Semantically Meaningful Feature Spaces for The Characterization of Video Content. In: Proceedings of International Conference on Image Processing, 1997, vol. 1, pp. 25–28 (1997)
3. Datta, A., Shah, M., Lobo, N.D.V.: Person-on-Person Violence Detection in Video Data. In: IEEE International Conference on Pattern Recognition, pp. 433–438 (2002)
4. Nam, J., Alghoniemy, M., Tewfik, A.H.: Audio-Visual Content-Based Violent Scene Characterization. In: IEEE International Conference on Image Processing, vol. 1, pp. 353–357 (1998)
5. Cheng, W., Chu, W., Wu, J.: Semantic Context Detection Based on Hierarchical Audio models. In: Proceedings of the 5th ACM SIGMM international Workshop on Multimedia information Retrieval, pp. 109–115 (2003)
6. Cai, L.J., Hofmann, T.: Text Categorization by Boosting Automatically Extracted Concepts. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and Development, pp. 182–189 (2003)
7. Akita, Y., Kawahara, K.: Language Modeling Adaptation Based on PLSA of Topics and Speakers. In: 8th International Conference on Spoken Language Processing, pp. 1045–1048 (2004)
8. Wold, E., Blum, T., Keislar, D., Wheaten, J.: Content-Based Classification, Search, and Retrieval of Audio, Multimedia, IEEE 3, 27–36 (1996)
9. Cai, R., Lu, L., Hanjalic, A., Zhang, H.J., Cai, L.H.: A Flexible Framework for Key Audio Effects Detection and Auditory Context Inference. IEEE Transaction on Audio, Speech and Language Processing 14, 1026–1039 (2006)
10. Wang, Y., Liu, Z., Huang, J.C.: Multimedia Content Analysis Using Both Audio and Visual Clues. IEEE Signal Processing Magazine 17, 12–36 (2000)
11. Sarkar, A.: Applying Co-Training Methods to Statistical Parsing. In: Proceedings of the 2nd Annual Meeting of the NAACL (2001)
12. Ng, V., Cardie, C.: Weakly Supervised Natural Language Learning Without Redundant Views. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 94–101 (2003)