19th Computer Vision Winter Workshop
Zuzana Kúkelová and Jan Heller (eds.)
Křtiny, Czech Republic, February 3–5, 2014

Violent Scenes Detection based on Automatically-generated Mid-level Violent Concepts

Shinichi Goto[1] and Terumasa Aoki[2]

[1]Graduate School of Information Sciences, Tohoku University, Miyagi, Japan
s-goto@riec.tohoku.ac.jp

[2]New Industry Creation Hatchery Center, Tohoku University, Miyagi, Japan
aoki@riec.tohoku.ac.jp

Abstract *Violent scenes detection in videos is a challenging problem because of the ambiguity of the word "violence." In this paper we introduce Mid-level Violence Clustering to solve this problem. Assuming three resource layers exist, it automatically generates mid-level violent concepts to infer violence without manually annotated tags of violent concepts such as fire, fights, etc. Our work is based on the combination of visual and audio features with machine learning at fixed segment-level. Multiple Kernel Learning is applied so that multimodality of data can be maximized, and finally a violence-score for each shot is calculated. We trained the whole system on a dataset from MediaEval 2013 Affect Task and evaluated it by its official metric MAP@100. The obtained results outperformed the best score in Affect Task.*

1 Introduction

Violent scenes detection is a task to detect violent actions in videos. It has been gathering attention just as MediaEval Affect Task [10] represents, which is intended to detect violent scenes in movies. MediaEval is a benchmarking workshop dedicated to evaluating systems for multimedia analysis and retrieval, including Affect Task, in which Technicolor [1] proposes the need of a system which enables users to choose movies that are suitable for their children by providing a preview of violent segments beforehand. Though even children can easily reach violent contents on the Internet nowadays, manually tagging or removing them is almost impossible because of their enormous number. This fact also makes it essential to develop the automatic classification system for violent videos.

The performance of previously proposed systems for violent scenes detection, however, is still unsatisfactory because of its complexity, as well as its ambiguous definition: e.g. Chen et al. defines violence as "a series of human actions accompanying with bleeding" in [5], though Giannakopoulos et al. defines it as "violent-related classes such as shots, fights and screams" in [11]. Simultaneously, rather than simply classifying each segment, it is required to claim which segment is more violent. This is the difference from general video classification problem. As a matter of fact, in Affect Task participants are asked to submit scores for violent segments.

The purpose of this study is to propose a novel system for shot-level violence classification and scoring in videos and to compare it with other algorithms. We use the definition of violence by 2013 Affect Task, which is "*physical violence accident resulting in human injury or pain*." Our system is based on fixed segment-level processing, which means first videos are divided into segments, each of which contains a fixed number of frames. Both of visual and audio feature vectors for each segment are extracted, and they are used to train classifiers. In order to make the most use of multimodality of data, Multiple Kernel Learning is applied to our system. In addition, Mid-level Violence Clustering is proposed in order for mid-level violent concepts to be learned automatically, without using manually annotated tags of concepts such as *fire*, *fight*, etc. "Mid-level" means this layer lies between low-level features and high-level final targets. Classifiers produce segment-level violence-scores, and finally they are converted to shot-level scores. Our system is trained and tested on a dataset from 2013 Affect Task, and evaluated by its official metric $MAP@100$. The effectiveness of Mid-level Violence Clustering is evaluated, and fusion methods are compared as well. We also compare our results with results by other participants who did not use external data. Finally, an investigation for each mid-level violence cluster is performed for further understanding.

2 Previous Work

Relatively few researches have been done for violent scenes detection. Some works used only audio information such as energy entropy and zero crossing [11], or utilized only visual features such as Bag-of-Visual-Words (BoVW) [6], Space Time Interest Points (STIP) [14] and camera motion [4, 5]. After extracted usually they are fed as input for Machine Learning such as Support Vector Machine (SVM) to give classification on test videos.

On the other hand, adopting both of visual and audio features is the current mainstream and have shown to improve results. The work by Nam et al. at 1998 [18] utilized this multimodality, proposing that violent signatures are represented as the combination of multiple features. Their fea-

ture extraction is based on flame detection, blood detection and audio features. In [15] PLSA was adopted to locate audio violence. PLSA is a probabilistic model utilizing the Expectation Maximization algorithm. For visual violence they used a linear weighted model fed with the results of violent event detection such as motion intensity, frame, explosion and blood. Finally co-training is carried out to utilize both modalities. Penet et al. compared two modality-fusion methods, namely Early Fusion and Late Fusion [20]. Early Fusion concatenates features from both modalities before machine learning, while Late Fusion fuses probabilities of both modalities already calculated. They reported Late Fusion was superior to Early Fusion. Derbas et al. [17] proposed Joint Audio-Visual Words representation, which constructs a codebook in the context of Bag-of-Words (BoW) by combining audio and visual features. Dai et al. [8] used external data from ImageNet and MIT scene dataset in addition to usual training and testing videos, in order to detect part-level attributes in each frame, each of which is expected to represent the likelihood of containing a certain object. Combining them with other low-level features from both of visual and audio modalities, the SVM classifier is built.

Researches above tried to detect violence directly from low-level features. Instead, some works have used violent concepts such as *fire*, *fights* and so on. Those concepts are manually annotated by humans and given in MediaEval Affect Task [10]. Ionescu et al. proposed a frame-level violence prediction, applying a multi-layer perceptron in order to utilize these concepts [12, 23]. They put the first layer for the concept prediction, and the second layer for the violence prediction. In addition to those provided concepts, Tan and Ngo [26] have utilized extra 42 violence concepts such as bomb and war from ConceptNet [16]. ConceptNet is composed of nodes representing concepts in the form of words or short phrases with their relationships. On their system those extra concepts are trained using YouTube videos which are crawled additionally. Afterwards a graphical model of those concepts are generated, and Conditional Random Fields [28] refines it by using relationships in ConceptNet and co-occurrence information of concepts. Their $MAP@100$ result was the first place in 2013 Affect Task with external data.

3 Violent Scenes Detection Based on Mid-level Violence Clustering

3.1 Approach Overview

Fig. 1 illustrates the overview of our approach. Feature extraction and training/classification are carried out at fixed segment-level. Here we define a segment as a sequence of 20 frames, and this means its time length is for 0.8 seconds if FPS is 25. The reason 20 is chosen is that if its length is too short, it might lack statistical meaning for its features, especially for its audio features. Or if the length is too long, features might get affected too much by changes of environments in scenes, such as a switch from a violent scene to a non-violent scene or camera motion.

First all training videos are divided into segments. Then both of visual and audio features are extracted for each segment (described in 3.2). Segments tagged as violent are gathered and divided into $K(> 0)$ clusters. As described in 3.3, we assume that each mid-level cluster implicitly represents a concept or a combination of concepts led to violence. Multiple Kernel Learning (MKL) is applied to generate a classifier for each cluster.

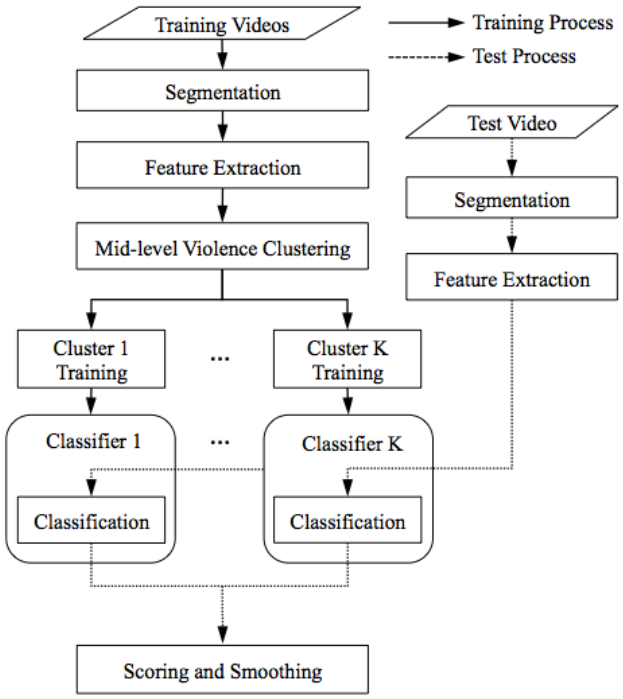

Figure 1: The overview of our approach.

In the test process, segmentation and feature extraction are performed in the same way as the training process. Classifiers in all clusters evaluate each segment, producing violence-scores. Then scores are integrated to generate one segment-level score for that segment. Smoothing is applied in order to take the context of videos into account, and finally segment-level scores are converted to shot-level scores. The following sections explain our feature vectors and training system more precisely.

3.2 Low-level Feature Extraction

Recent works on violent scenes detection such as [26] and [8] have shown the effectiveness of using trajectory-based features as visual information and MFCC-based features as audio information. Similar to those researches, in total six feature spaces exist on our system: Trajectory, HOG, MBHx, MBHy, RGB and Audio.

Dense Trajectory

Trajectories have been used to capture local motion of videos, especially in the field of action recognition. We use Dense Trajectory [29], a trajectory to which dense sampling is applied. Except for those in homogeneous areas, densely sampled points are tracked by calculating optical flows in each spatial scale until they reach the length of $L = 15$ frames. Every frame newly sampled points are added if no tracked point is found in the neighborhood of each pixel. We use 32 for a neighbour

range, 5 for a sampling step, and 6 for a spatial scale size. Displacement vectors of trajectories are extracted for both of x-direction and y-direction and concatenated (30-dimension). Following [29], descriptors are extracted around each trajectory: HOG, MBHx, MBHy, and RGB-histogram. Although originally HOF (Histograms of Oriented Optical Flow) is extracted as well, expected to have poor contribution on our task because of its frequent camera motion, it is removed.

HOG

HOG (Histograms of Oriented Gradients) is a descriptor for local gradient orientations, and is largely used for object detection. In the same way as [29], the neighbour of trajectories are divided into $2 \cdot 2$ areas. For each area 8-dimensional HOG is calculated and averaged every 5 frame. Since each trajectory has 15 frames length, concatenating all of them generates $12 (= 2 \cdot 2 \cdot 3)$ histograms, resulting in $8 \cdot 12 = 96$ dimensional vectors.

MBHx and MBHy

MBH (Motion Boundary Histograms) was originally proposed in the field of human detection by Dalal et al. [9] to represent the changes in the optical field, namely local motion information independent of camera motion, by calculating the gradient of the optical flow. MBH is generated separately along the vertical direction (MBHx) and the horizontal direction (MBHy). Since each MBH is represented as an 8-dimensional vector, similar to HOG, both of MBHx and MBHy are described as 96-dimensional vectors around trajectories.

RGB-histogram

Although originally RGB information are not extracted in [29], in violence detection since color information is expected to be helpful just as blood and flame detection have contributed to the results in some previous researches [15, 18], 64-bin RGB histograms around trajectories are calculated every 5 frame. Spatial division is not carried out for RGB-histogram, and it results in a 192-dimensional vector.

Audio

Similar to Bag-of-Audio-Words in [19], MFCC (Mel Frequency Cepstrum Coefficients) and the log energy are first extracted every 10ms with 5ms overlap for audio features. The first derivative of MFCC and its energy are also calculated as delta-MFCC, producing a 26-dimensional feature.

Trajectory-based features are assigned to a segment in which their trajectory has reached 15 frames length. For each segment these features are gathered, converted to the BoW form by using already calculated codebooks, and normalized. Codebooks are generated by using randomly selected 100,000 features and k-means++ algorithm beforehand in each feature space respectively. Finally 200-dimensional Trajectory, 400-dimensional HOG, 200-dimensional MBHx, 200-dimensional MBHy, 400-dimensional RGB histogram, and 200-dimensional Audio vectors are obtained.

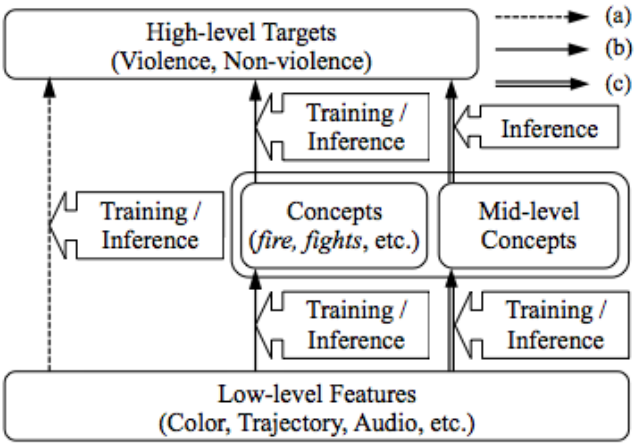

Figure 2: Three layers in violence detection and our actions: (a) trains and infers violence directly from low-level features, (b) trains and infers violent concepts using annotations first, and uses them to train and infer violence, (c) trains and infers mid-level concepts without annotations of concepts first, and uses them to infer violence. Our system follows (c).

3.3 Mid-level Violence Clustering

We assume there are three layers in violence detection as Fig. 2 displays. Most of previous works only used low-level features to directly train and infer high-level targets, namely Violence and Non-violence (Fig. 2(a)). Some works have started using manually annotated violent concepts. Using those given concepts, they train and infer concepts in test videos to train and infer violence finally (Fig. 2(b)). The reason why this mid-level layer is needed that the diversity of "violence" is huge: even though two segments are annotated as violent, their low-level features might be largely different depending on their characteristics of violence. For instance, although explosion scenes labelled as violent might have distinctive visual features, those of scream scenes might not similar even if they are also labelled as violent. Our system, however, takes the approach Fig. 2(c). Instead of actual violent concepts, it detects violence using *mid-level concepts* which have been automatically inferred without annotations of violent concepts. We apply Mid-level Violence Clustering and generate clusters for mid-level concepts prediction.

Fig. 3 illustrates the process of Mid-level Violence Clustering. First all of the violent segments in training videos are gathered. Then they are divided into $K (> 0)$ clusters, each of which is expected to contain similar segments. Then non-violent training segments are assigned to those clusters sequentially, whose results construct clusters for mid-level violence classifiers. After concatenating feature vectors of them, k-means++ algorithm with Euclidean distance is applied to generate clusters. Here we assume that feature vectors for violent segments are capable of representing one or multiple concepts related to violence in each cluster. If mid-level concepts are correctly clustered, "training violence in one cluster" corresponds to "training one or multiple mid-level violent concepts." This means manual annotations for violent concepts are unnecessary if our previous assumption is correct. Mid-level Violence Clustering also contributes to reduction in complexity. Since the number of feature points

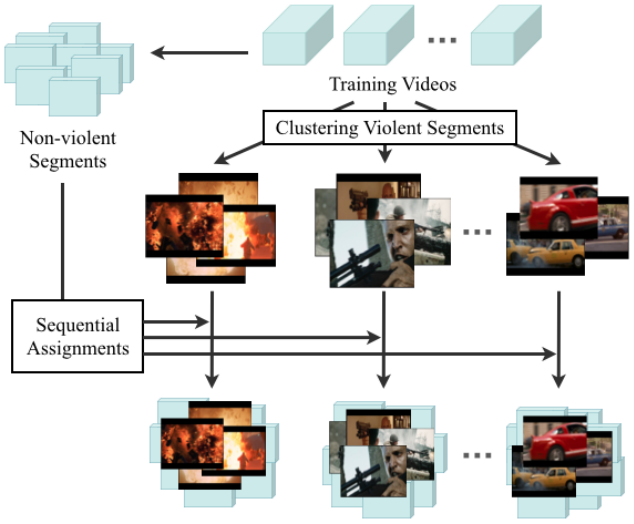

Figure 3: The process of Mid-level Violence Clustering.

and dimensions are huge in the task of Violent Scenes Detection, the cost for training is expensive. Training multiple classifiers needs much less time compared to training one classifier using all feature vectors. The process of actual training, classification and scoring are described in the following sections.

3.4 Multiple Kernel Learning

BoVW with SVM [6] has contributed greatly to the filed of image classification over the last few years. For the task of violence detection, however, multiple feature spaces have to be handled, and then simply concatenating feature vectors and training classifiers might not always be the best way, according to the work by Penet et al. [20]. As they studied, when multimodal features exist, there are two available fusion schemes: Early Fusion (EF) and Late Fusion (LF). EF concatenates features from both modalities before training, meaning it can take correlations of those feature spaces into account while training, though it is dealing with each feature space uniformly. On the other hand, on LF training and classification are performed for each feature space independently. Generated results, which are violence probabilities in their experiment, are fused afterwards. They compared EF with LF when two modalities exist (visual and audio). Although they concluded that LF has more effectiveness, if more modalities exist just as our system, it has some drawbacks: 1) it cannot take correlations of multimodal features into account while training and classification, 2) how to fuse both results has to be decided manually beforehand. Besides, in [20] low-level audio features and higher-level visual features such as shot length were directly combined, and this might have resulted in poor correlations between two modalities.

To cope with this problem and to maximize the multimodality of data, we apply Multiple Kernel Learning (MKL), which can be regarded as a kind of EF, but aims at finding optimized weights for each feature space when multiple SVM kernels are applied [25]. This means MKL considers correlations of multiple feature spaces while training and classification, but at the same time it considers differences of violent characteristics among multiple feature spaces. In MKL the whole kernel is composed of multiple sub-kernels, and is defined as a following equation:

$$\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_p \beta_p \boldsymbol{K}_p(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{1}$$

where $\boldsymbol{K}_p$ are sub-kernels, and β_p is a weight for p-th sub-kernel. Sub-kernels for Trajectory, HOG, MBHx, MBHy, RGB-histogram and Audio are prepared in our case. The dual for the MKL primary problem is proposed by Bach et al. [2] and parameters can be learned.

Histogram Intersection Kernel (HIK), which has been reported to perform well on histogram-based features [3], is adopted as a sub-kernel. HIK is defined as follows:

$$\boldsymbol{K}_{int}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{m} min(a_i, b_i) \tag{2}$$

where $\boldsymbol{A} = [a_1, a_2, \ldots, a_m]$ and $\boldsymbol{B} = [b_1, b_2, \ldots, b_m]$. It measures the degree of similarity between two histograms. MKL is applied to all K clusters, generating K classifiers. SHOGUN Toolbox [24] is used for our MKL implementation.

3.5 Scoring and Smoothing

Each segment in test videos is classified as violent or non-violent by K classifiers. If a video has N segments, results obtained by k-th classifier $(1 \leq k \leq K)$ are:

$$\boldsymbol{C}_k = [c_{1,k}, c_{2,k}, \ldots, c_{N,k}] \tag{3}$$
$$\boldsymbol{D}_k = [d_{1,k}, d_{2,k}, \ldots, d_{N,k}] \tag{4}$$

where $c_{n,k} \in \pm 1$ $(1 \leq n \leq N)$ represents that n-th segment is violent if its value is $+1$, while it represents non-violent if its value is -1. $d_{n,k}$ denotes a distance between a feature point of n-th segment and a hyperplane which has classified it. Using $\boldsymbol{D}_k$, we define $\boldsymbol{S}_k$, scores for all segments by k-th classifier as follows:

$$\boldsymbol{S}_k = [s_{1,k}, s_{2,k}, \ldots, s_{N,k}], \tag{5}$$
$$s_{n,k} = \begin{cases} d_{n,k} \; (if \; c_{n,k} = +1) \\ 0 \; (if \; c_{n,k} = -1) \end{cases} \tag{6}$$

They are integrated to produce pre-final scores $\boldsymbol{S}$:

$$\boldsymbol{S} = [s_1, s_2, \ldots, s_N], \tag{7}$$
$$s_n = \frac{\sum_{l=1}^{K} s_{n,l}}{K_{vio}} (1 \leq n \leq N) \tag{8}$$

where K_{vio} is the number of classifiers whose $c_{n,k}$ is +1, in other words, the number of classifiers which classify n-th segment as violent. This means for each segment, if no cluster classifies it as violent, its violence-score is zero, while the mean value of violence-scores of classifiers which classify it as violent is assigned if one or more clusters classify it as violent.

In order to take the context of a video into account, scores are smoothed as a final step. Although in [8] the average value over a three-shot window is calculated, we adopt a

moving average calculation so that the further neighbour segments are positioned, the lesser their effects are considered. Smoothed scores $\boldsymbol{S}'$ are calculated by using pre-final scores $\boldsymbol{S}$ as follows:

$$\boldsymbol{S}' = [s'_1, s'_2, \ldots, s'_N], \tag{9}$$

$$s'_i = \frac{s_i + \sum_{m=1}^{M} \alpha^n \cdot (s_{i-m} + s_{i+m})}{2M+1} \tag{10}$$

where α $(0 < \alpha < 1)$ is a smoothing coefficient, and M is a neighbor range around a segment. We used 0.5 for α and 2 for M.

Scores for shots are calculated by converting segment-level scores after calculating frame-level scores. Because the numbers of frames in segments are consistent except for a final segment of a video, frame-level scores are simply given as scores for segments which have those frames. Then for each shot, scores for frames it contains are summed and divided by the number of frames. This score is used as a final score for each shot.

4 Experiment

Though multiple tasks exist in 2013 Affect Task, we focus on shot-level violence detection in movies with objective definition, which is "*physical violence accident resulting in human injury or pain*," without external data. For the evaluation, shot-level violence-scores have to be generated rather than merely classifying shots.

In order to ascertain the improvement by Mid-level Violence Clustering, multiple numbers of K are tried. Additionally a system with clusters constructed for each training movie instead of using Mid-level Violence Clustering is tested, in which multiple violent concepts are mixed in each cluster. We call this as Training Movie Grouping, and compare it with Mid-level Violence Clustering. Results are also compared with runs by other participants. To evaluate the effect of MKL, EF and LF are performed using normal SVM with HIK. Though various kinds of implementations are possible for LF, on our system visual (Trajectory, HOG, MBHx, MBHy and RGB) features are concatenated, and trained separately from audio features. Since results by classifiers are scores rather than probabilities, simply a higher score is used as a score by that cluster.

4.1 Dataset

With automatically generated shot boundaries by Technicolor's software [10], 18 training movies and 7 test movies are provided. (Training movies: *Armageddon, Billy Elliot, Eragon, Harry Potter 5, I am Legend, Leon, Midnight Express, Pirates of the Caribbean 1, reservoir Dogs, Saving Private Ryan, The Sixth Sense, The Wicker Man, Kill Bill 1, The Bourne Identity, The Wizard of Oz, Dead Poets Society, Fight Club and Independence Day*. Test movies: *Fantastic Four, Fargo, Forrest Gump, Legally Blond, Pulp Fiction, The God Father 1 and The Pianist*.) Training movies are given with frame-level violence ground truth annotated by several human assessors. Though in 2013 Affect Task participants were allowed to use prepared violent concepts, our algorithm uses only low-level features extracted from movies, violence ground truth and shot boundaries. To reduce the complexity, all frames are resized to half of their original size as pre-processing.

4.2 Evaluation Metric

MediaEval 2013 Affect task adopted Mean Average Precision at the 100 top ranked violent shots ($MAP@100$) as its official metric. MAP is the most standard evaluation of ranked retrieval results among the TREC (Text Retrieval Conference) community [7], and it provides a single-figure measure of quality across recall levels. MAP is the mean value of the Average Precision (AP), which can consider the order which targets are presented in. It computes the average value of $Precision$ over the interval from $n = 1$ to $n = N$:

$$AP@N = \frac{1}{N} \sum_{n=1}^{N} Precision(\boldsymbol{R}_n) \tag{11}$$

where N is the maximum rank number one wants to calculate, and $\boldsymbol{R}_n$ is the set of ranked retrieval violent segments from the top result to the n-th result. Then MAP is calculated as follows:

$$MAP@Q = \frac{1}{Q} \sum_{q=1}^{Q} AP@q \tag{12}$$

For instance, if ranked results are judged as $[true, false]$, $AP@1 = 1 \cdot 1 = 1$, $AP@2 = (1 \cdot 1 + 1 \cdot 0.5)/2 = 0.75$, and $MAP@2 = (AP@1 + AP@2)/2 = (1 + 0.75)/2 = 0.875$.

4.3 Results and Discussion

For the number of clusters K, we tried every 5 numbers from 5 to 120. Fig. 4 displays these results before smoothing since smoothing improved scores largely, especially when the numbers of clusters were low, making it difficult to evaluate the effectiveness of Mid-level Violence Clustering. Results with small numbers of clusters seem to be unstable compared to results with high numbers. This reveals the effectiveness of Mid-level Violence Clustering, since a small number means there are not enough clusters to represent violent concepts. This figure also compares them with a result from Training Movie Grouping, and scores by Mid-level Violence Clustering were superior to its score. On Training Movie Grouping, each cluster might have multiple violent concepts whose feature vectors can be largely different each other, and then it also proves the effect of Mid-level Violence Clustering.

While we expected low scores for results with high numbers of clusters (e.g. $K = 100$), they seem to be able to keep promising scores. This is because when the number of clusters was high, some clusters had only a few violent segments assigned to themselves due to their anomalies. For instance, when we chose $K = 100$, the smallest number of violent segments in one cluster was 2, although other clusters tended to contain about 50-100 violent segments. Even though this cluster could not find violent segments, it did not affect a final score either because our scoring equation (6) depends only on scores by classifiers that have classified a target segment as violent.

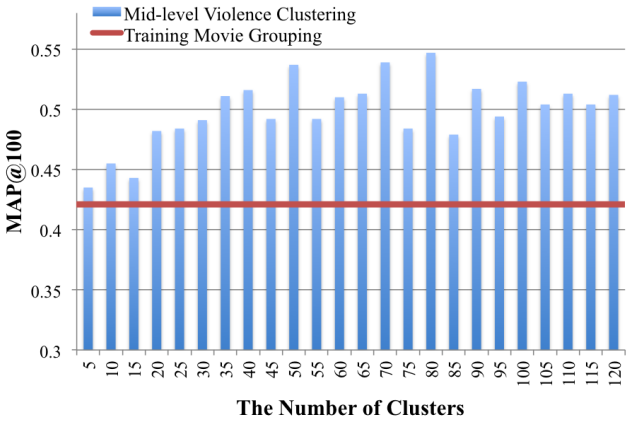

Figure 4: Results of Mid-level Violence Clustering with multiple numbers of clusters without smoothing, and a result of Training Movie Grouping.

Run	$MAP@100$
LIG [17]	0.520
FAR [23]	0.496
Fudan [8]	0.492
NII [13]	About 0.400
Technicolor [21]	0.338
VISILAB [22]	0.150
MTM [27]	0.070
Our system (K=50)	0.558
Our system (K=80)	0.577

Table 1: Comparison with other teams on MediaEval 2013 Affect Task without external data. Our result outperformed them.

Smoothing always improved scores, and then only two best results are shown in Tab. 1 with results by other participants in Affect Task 2013. Although some teams used given concepts for their runs, one can find our score outperforms their scores.

Fusion methods are compared in Fig. 5. All of them are results after the smoothing step. MKL was always better than or equal to EF, and always superior to LF, proving the effectiveness of MKL. Partly because of the difference of how to implement LF, EF was always superior to LF, being different from a report in [20]. This can be considered as being caused by the difference of features. In [20] low-level audio features such as zero crossing rate and energy were used, although higher-level visual features such as shot duration and number of flashes were chosen. This led to few correlations among these two modalities as authors mentioned. In our case, however, since both features were low-level, there seem to have existed correlations among them, which could be maximized when they were combined by applying MKL or EF.

For the run with ($K = 50$), example frames of shots that had high scores are shown in Fig. 6. Among these 4 shots, (a), (b) and (c) were correct estimations for scenes containing explosions, gunshots and car chases. However, the estimation (d) was wrong. In this shot multiple people start

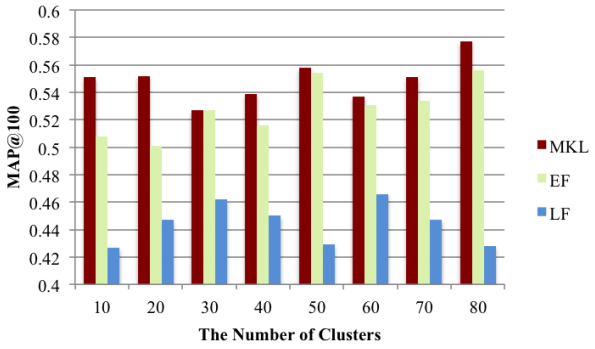

Figure 5: Comparison of fusion methods with smoothing. MKL=Multiple Kernel Learning, EF=Early Fusion, LF=Late Fusion.

standing up suddenly and cheering. Just as this example, shots that contain multiple people, sudden motion and big sound tended to be miss-classified as violent. Meanwhile, common missed violent shots were violent scenes without sound, such as a scene in which a man is wringing other man's neck.

Though our system has achieved promising scores, its performance is still insufficient and multiple points can be argued. The first point to be considered here is that our feature vectors might not be distinct enough. Although we have used trajectory-based features as visual information, they can be easily affected by camera motion. Even though features such as MBH, which are supposed to be robust to camera motion, were extracted, they might be noisy if trajectories themselves are unreliable. Similarly, shot boundaries are not considered on our system although shots often change in the middle of segments and are expected to affect visual features. The second point is that though Euclidean distance is used for the similarity while Mid-level Violence Clustering, Histogram Intersection is used for sub-kernels in MKL. Performing clustering by using Histogram Intersection might be essential to keep consistency.

4.4 Extensive Study

In order to confirm our assumption in 3.3, we examined the amount of violence-related concepts in it using the cluster number $K = 50$. In 2013 Affect Task, participants were provided with violent concepts annotated at frame-level by human assessors. They consist of 7 visual concepts: *presence of blood, presence of fire, fights, gory scenes, presence of firearms, presence of cold weapons car chases*, and 3 audio concepts: *explosions, presence of screams, gunshots*. It should be noted that these concepts are not always related to violence ground truth, and often multiple concepts are tagged in one frame. Since they are at frame-level, we converted them to segment-level annotations by simply tagging each segment if half of frames it contains are annotated.

The ratios of segments annotated by each concept in each cluster are shown in Fig. 7. Since it is inadequate to display ratios for all 50 clusters and for all concepts in this figure due to the limit of the available spaces, only 6 representative clusters are displayed. Also annotation *car chases* is

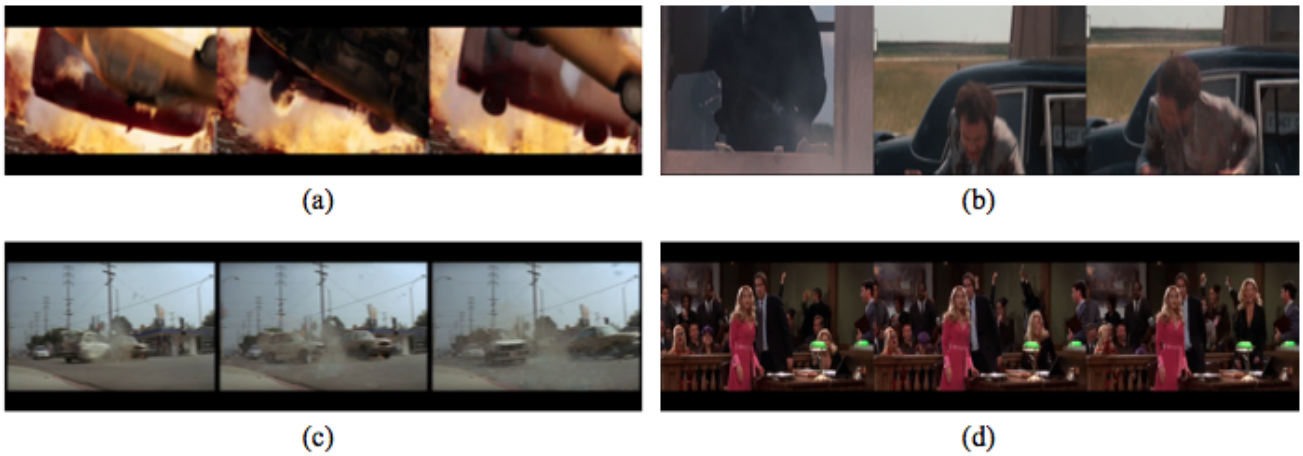

Figure 6: Example frames of shots which had high violence-scores on our system. ($K = 50$): (a) a car is blown away by the explosion, (b) a man is shot multiple times from a window and his shirt gets soaked with blood, (c) a car crashes into another, (d) people start standing up and cheering. Note in (b) the camera perspective seems to change but this is because shot boundaries were automatically generated.

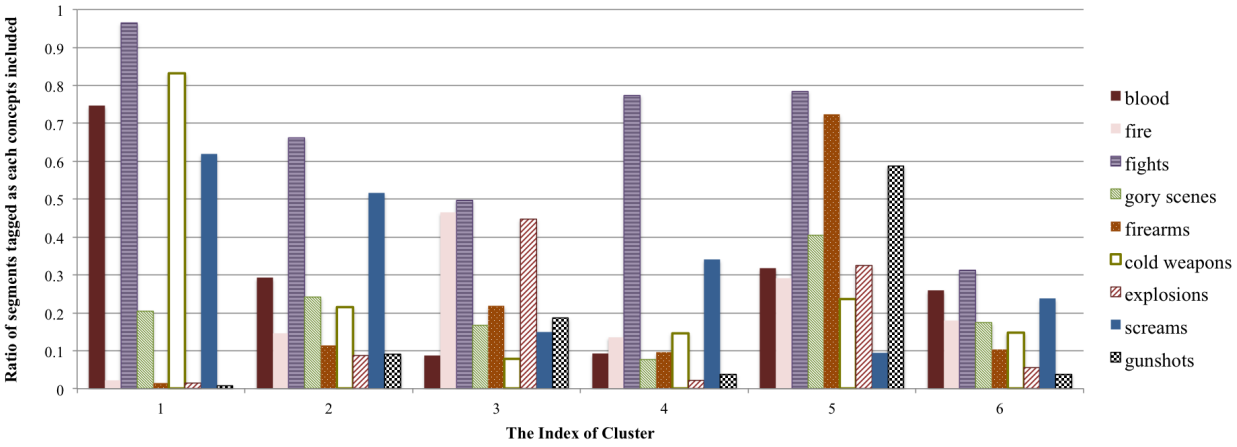

Figure 7: Ratios of segments annotated by each concept for clusters ($K = 50$). Note only 6 representative clusters are shown and a tag "*car chases*" is excluded.

excluded due to its low number. By studying this figure one can find some clusters reflect violent concepts. For instance, although both of Cluster 1 and Cluster 4 have high ratios for *fights*, Cluster 1 has more *blood* and *cold weapons*, meaning these two clusters represent different kinds of violence. Cluster 5 includes a high number of segments tagged as *firearms* and *gunshots*. We investigated this cluster and found it contains gunfire scenes.

On the other hand, there exist clusters which seem not to reflect actual violent concepts like Cluster 6. Though clusters generated by our system and concepts in MediaEval are unrelated essentially, and so characteristics of clusters in Fig. 7 do not always have to be distinctive, it could have been caused by the lack of distinctiveness of features or inconsistency of the clustering method.

5 Summary and Conclusions

In this paper, we proposed a novel system to detect violent scenes in videos by using Mid-level Violence Clustering with multimodal features. Our experiments showed that automatic inference of mid-level concepts is effective for this task, and results outperformed the best $MAP@100$ in MediaEval 2013 Affect Task even without manually annotated concepts. In addition, comparison of fusion methods, as well as investigation for concepts in mid-level violence clusters were performed. Future work is to find more discriminative feature vectors, as well as to adopt more suitable clustering method in the context of our system.

Acknowledgement

We acknowledge MediaEval 2013 Affect Task: Violent Scenes Detection http://www.multimediaeval.org/ for providing a dataset that has been supported, in part, by the Quaero Program http://www.quaero.org.

References

[1] Technicolor. http://www.technicolor.com. Last visited Dec. 2013.

[2] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple Kernel Learning, Conic Duality, and

the SMO Algorithm. In *ICML '04 Proceedings of the twenty-first international conference on Machine Learning*, 2004.

[3] A. Barla, F. Odone, and A. Verri. Histogram Intersection Kernel for Image Classification. In *Proceedings of ICIP 2003.*, pages 513–516, Sept. 2003.

[4] E. Bermejo, O. Deniz, G. Bueno, and R. Sukthankar. Violence Detection in Video Using Computer Vision Techniques. In *CAIP'11 Proceedings of the 14th international conference on Computer analysis of images and patterns - Volume Part II*, pages 332–339, 2011.

[5] Liang-Hua Chen, Hsi-Wen Hsu, Li-Yun Wang, and Chih-Wen Su. Violence Detection in Movies. In *Computer Graphics, Imaging and Visualization (CGIV), 2011 Eighth International Conference*, Aug. 2011.

[6] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cdric Bray. Visual Categorization with Bags of Keypoints. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 1–22, 2004.

[7] Christopher D., Manning, Prabhakar Raghavan, and Hinrich Schutze. In *Introduction to Information Retireval*, 2008.

[8] Qi Dai, Jian Tu, Ziqiang Shi, Yu-Gang Jiang, and Xiangyang Xue. Fudan at MediaEval 2013: Violent Scenes Detection Using Motion Features and Part-Level Attributes. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, Oct. 2013.

[9] N. Dalal, B. Triggs, and C. Shmid. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, 2006.

[10] C. Demarty, C. Penet, M. Schedl, B. Ionescu, V.L. Quang, and Y. Jiang. The Mediaeval 2013 Affect Task: Violent Scenes Detection. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.

[11] Theodoros Giannakopoulos, Dimitrios Kosmopoulos, Andreas Aristidou, and Sergios Theodoridis. Violence Content Classification Using Audio Features. In *SETN'06 Proceedings of the 4th Helenic conference on Advances in Artificial Intelligence*, pages 502–507, 2006.

[12] Bogdan Ionescu, Jan Schluter, Ionut Mironica, and Markus Schedl. A Naive Mid-level Concept-based Fusion Approach to Violence Detection in Hollywood Movies. In *ICASSP - 37th International Conference on Acoustics, Speech, and Signal Processing*, 2012.

[13] Vu Lam, Duy-Dinh Le, Sang Phan, Shin'ichi Satoh, and Duc Anh Duong. NII-UIT at MediaEval 2013 Violent Scenes Detection Affect Task. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, Oct. 2013.

[14] I. Laptev. On Space-Time Interest Points. In *International Journal of Computer Vision*, volume 64, pages 107–123, 2005.

[15] Jian Lin and Weiqiang Wang. Weakly-Supervised Violence Detection in Movies with Audio and Video Based Co-training. In *PCM '09 Proceedings of the 10th Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, pages 930–935, 2009.

[16] H. Liu and P. Singh. ConceptNet — a practical commonsense reasoning tool-kit. In *BT Technology Journal*, volume 22, pages 211–226, Oct. 2004.

[17] Bahjat Safadi Nadia Derbas and Georges Quenot. LIG at Mediaeval 2013 Affect Task: Use of a Generic Method and Joint Audio-Visual Words. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, Oct. 2013.

[18] Jeho Nam, Masoud Alghoniemy, and H. Tewfik. Audio-Visual Content-Based Violent Scene Characterization. In *International Conference on Image Processing*, Oct. 1998.

[19] Stephanie Pancoast and Murat Akbacak. Bag-of-audio-words Approach for Multimedia Event Classification. In *Interspeech Conference*, 2012.

[20] Cedric Penet, Claire-Helene Demarty, Guillaume Gravier, and Patrick Gros. Multimodal Information Fusion and Temporal Integration for Violence Detection in Movies. In *ICASSP - 37th International Conference on Acoustics, Speech, and Signal Processing*, 2012.

[21] Cedric Penet, Claire-Helene Demarty, Guillaume Gravier, and Patrick Gros. Technicolor/INRIA Team at the MediaEval 2013 Violent Scenes Detection Task. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, Oct. 2013.

[22] Ismael Serrano, Oscar Deniz, and Gloria Bueno. VISILAB at MediaEval 2013: Fight Detection. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, Oct. 2013.

[23] Mats Sjoberg, Jan Schluter, Bogdan Ionescu, and Markus Schedl. FAR at MediaEval 2013 Violent Scenes Detection: Concept-based Violent Scenes Detection in Movies. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, Oct. 2013.

[24] Soeren Sonnenburg, Gunnar Raetsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtech Franc. The SHOGUN Machine Learning Toolbox. In *Journal of Machine Learning Research, 11*, pages 1799–1802, June 2010.

[25] S. Sonnenburg, G. Raetsch, C. Schaefer, and B. Schoelkopf. Large Scale Multiple Kernel Learning. In *Journal of Machine Learning Research*, 2006.

[26] Chun Chet Tan and Chong-Wah Ngo. The Vireo Team at MediaEval 2013: Violent Scenes Detection by Mid-level Concepts Learnt from Youtube. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.

[27] Bruno Do Nascimento Teixeira. MTM at Mediaeval 2013 Violent Scenes Detection: Through Acoustic-visual Transform. In *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, Oct. 2013.

[28] Hanna M. Wallach. Conditional Random Fields: An Introduction. In *Technical Report MS-CIS-04-21*, 2004.

[29] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States, June 2011.