A Visualization Tool for Violent Scenes Detection

Dominique Maniry, Esra Acar, Frank Hopfgartner, Sahin Albayrak
DAI Laboratory, Technische Universität Berlin
Ernst-Reuter-Platz 7, TEL 14
10587, Berlin, Germany
{name.surname}@tu-berlin.de

ABSTRACT
We present a browser-based visualization tool that allows users to explore movies and online videos based on the violence level of these videos. The system offers visualizations of annotations and results of the MediaEval 2012 Affect Task and can interactively download and analyze content from video hosting sites like YouTube.

Categories and Subject Descriptors
H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems

General Terms
Algorithms, Design, Experimentation

Keywords
Multimedia IR, Video, Audio Analysis, Violence Detection

1. INTRODUCTION
The MediaEval Affect Task [2] aims to detect violent segments in movies. Defining the term “violence”, when applied to characterize movies, is a hard and subjective (i.e., person-dependent) task. In our work, we aim at sticking to the common definition of violence: “physical violence or accident resulting in human injury or pain” which is the definition of “violence” in the frame of the MediaEval Affect Task. Detailed description of the task, the dataset, the ground truth and evaluation criteria are given in the paper by Demarty et al. [2]. The development and evaluation of Violence Scenes Detection (VSD) creates the need for detailed visualization to assess the strengths and weaknesses of algorithms. Our system consists of three parts: the Ranked List view shows the results on the test set of the MediaEval Affect Task, the Annotations view of our visualization tool shows the annotations of the MediaEval training set and the Online Analysis carries out our analysis pipeline [1] to arbitrary online videos.

The paper is structured as follows. Section 2 gives a brief overview of our violence analysis method and the visualization tool. In Section 3, we discuss the performance of the violence analysis method. Finally, we explain our demonstration in Section 4.

2. SYSTEM OVERVIEW

2.1 The Method
Among the plurality of audio features, Mel-Frequency Cepstral Coefficients (MFCC) are shown to be indicators of the excitement level of video segments [4]. Therefore, we employ them as low-level audio features. For the representation of video segments, we use mid-level audio features based on MFCC (i.e., a Bag-of-Audio Words (BoAW) approach). We apply the BoAW approach with two different coding schemes, namely vector quantization and sparse coding. We train a pair of two-class SVMs in order to learn violence models using both mid-level feature representations. Normally, in a basic SVM, only class labels or scores are output. The class label results from thresholding the score, which is not necessarily a probability measure. The scores output by the SVM are converted into probability estimates using the method explained in [3].

2.2 Ranked List
The user first selects the run (algorithm and parameters), of which the results will be visualized. The user can also select a specific test movie or the whole test set. The Ranked List view (Figure 1) then shows the thumbnails of all segments overlayed by the violence score (i.e., the probability of violence), time information and a notice whether the classification matches the ground truth. If a segment is classified as violent, the thumbnail is highlighted with an orange frame around it. This enables the user to interpret the results easily and quickly. A click on the thumbnail plays the given segment without leaving the Ranked List. The user can sort the list by violence scores of the segments that the algorithm identified as the most or least violent, or sort by time to see the classification results from the beginning to the end of the movie. We also added a button that jumps to a random part of the list to enable a more dynamic exploration experience.

2.3 Annotations
The training set of the MediaEval 2012 Affect Task provides annotations for 15 Hollywood movies. The annotations mark the presence of audio, visual and audio-visual con-
cepts such as explosions, gunshots, screams, blood, fights, car chases, fire, firearms, cold weapons and gore. The user can query any or all movies for any of these concepts (e.g., show all segments with fire in *Saving Private Ryan*). The annotations are then displayed in a view (Figure 2) similar to the one of the Ranked List.

![Figure 1: Ranked List view](image1)

![Figure 2: Annotations view](image2)

2.4 Online Analysis

The Online Analysis (Figure 3) executes our VSD pipeline to any video hosted by YouTube (or any other site supported by the youtube-dl script). After the user entered the URL, the video is downloaded, transcoded and split into segments. The MFCC feature vectors of the audio of each segment are subsequently computed and used to build mid-level features with sparse coding and vector quantization as explained in [1]. Both mid-level feature representations are used to classify the segment and produce two violence scores. Even though our methods only use audio features, the Online Analysis pipeline can be applied to any method using audio, visual or audio-visual features. In addition to the Ranked List view, the Online Analysis produces a summary box with the maximum violence score and the number of violent segments. The results are cached so that a query of a previously seen video can return the results immediately without downloading or classifying again.

![Figure 3: Screenshot of Online Analysis view (showing classification results)](image3)

3. DISCUSSION

Our method is able to suitably detect violent content such as fights and disasters with explosions. Video segments which contain no excitement (e.g., containing a man giving a speech or strong music in the background) are also easily classified as non-violent. On the other hand, the method wrongly classifies a video segment as violent when the segment contains very strong sounds or exciting moments such as a plane taking off or loud ringing bell. The most challenging violent segments to be detected are the ones which are “violent” according to the definition of violence within the MediaEval Affect Task, but which actually contain actions such as self-injuries, or other moderate actions such as an actor pushing or hitting slightly another actor. Our method is also unable to detect violent video segments which are “violent” according to the definition of violence, but which contain no audio cues exploitable for the identification of violence (e.g., a man bleeding). More detailed discussion on the performance of our method is given in [1].

4. DEMONSTRATION

We will demonstrate all three parts of the system and show the different ways to navigate through the results. The demonstration will show results on various videos and show that the system can be used to find the strengths and weaknesses of a given method.

5. REFERENCES


