Breaking Down Violence Detection: Combining Divide-et-Impera and Coarse-to-Fine Strategies

Esra Acar[a,*], Frank Hopfgartner[b], Sahin Albayrak[a]

[a]*Technische Universität Berlin, Distributed Artificial Intelligence Laboratory, Ernst-Reuter-Platz 7, 10587, Berlin, Germany*
[b]*University of Glasgow, Humanities Advanced Technology and Information Institute, University Gardens, Glasgow, UK*

Abstract

In today's society where audio-visual content is ubiquitous, violence detection in movies and Web videos has become a decisive functionality, e.g., for providing automated youth protection services. In this paper, we concentrate on two important aspects of video content analysis: Time efficiency and modeling of concepts (in this case, violence modeling). Traditional approaches to violent scene detection build on audio or visual features to model violence as a single concept in the feature space. Such modeling does not always provide a faithful representation of violence in terms of audio-visual features, as violence is not necessarily located compactly in the feature space. Consequently, in this paper, we target to close this gap. To this end, we present a solution which uses audio-visual features (MFCC-based audio and advanced motion features) and propose to model violence by means of multiple (sub)concepts. To cope with the heavy computations induced by the use of motion features, we perform a coarse-to-fine analysis, starting with a coarse-level analysis with time efficient audio features and pursuing with a fine-level analysis with advanced features when necessary. The results demonstrate the potential of the proposed approach on the standardized datasets of the latest editions of the *MediaEval Affect in Multimedia: Violent Scenes Detection (VSD) task* of 2014 and 2015.

Keywords: Event Detection, Violence Concept, Ensemble Learning, Feature Space Partitioning, Coarse-to-Fine Violence Analysis, Support Vector Machine

*Corresponding author
Email address: esra.acar@tu-berlin.de (Esra Acar)

1. Introduction

The amount of multimedia content accessible to consumers becomes more and more abundant. This creates a need for automatic multimedia analysis solutions which can be used to find relevant semantic search results or to identify illegal content present on the Internet. In parallel, the developments in digital media management techniques have simplified delivering digital videos to consumers. As a consequence, gaining access to online film productions offered on platforms such as Video-On-Demand (VOD) services has literally become a *child's play*, with the risk that children be exposed to movies or reality shows which have not been checked by parents. Thus, these might contain inappropriate content, as one cannot expect that parents constantly and precisely monitor what their children are viewing. Violence constitutes one typical example of such inappropriate content, whose negative effects have been evidenced [1]. Consequently, a need for automatically detecting violent scenes in videos (e.g., movies, Web videos) has appeared.

Nowadays, movies receive different ratings in different countries (e.g., age of 0, 12, etc.). Even if there is an agreement between different national rating institutes, the perception of violence can still differ from person to person, even within a group of persons of identical age. Due to the subjective nature of the "violence" concept, one of the challenges is to adequately delimit the boundaries of what can be designated as a "violent" scene. Therefore, one preliminary step is the adoption of a definition of violence to work with. We adhere to the definition of violence as described in [2]: *subjective violence*. According to [2], *subjective violent* scenes are "those which one would not let an 8 years old child see because they contain physical violence".

In this context, the *MediaEval Affect in Multimedia: Violent Scenes Detection* (VSD) task [3], held yearly since 2011, has provided a consistent evaluation framework to the research community and enabled various approaches to be evaluated and compared by using the same violence definition and a standardized dataset. Interested readers will find a comprehensive description of the task, dataset, ground truth and evaluation criteria in [3]. The task stems from a use case attributed to the company *Technicolor*[1]. The French producer of video content and entertainment technologies adopted the aim of helping users to select movies that are suitable to watch with their children. This helps them decide if, according to their own criteria, the movie is adequate to be watched by their child.

[1] `https://research.technicolor.com/rennes/`

For the reasons we stated above, an effective violence detection solution, which is designed to automatically detect violent scenes in movies (or in videos in general), is highly desirable. Such an automated solution requires working with a proper representation of data which is an essential processing step. Recently, solutions using mid-level feature representations have gained popularity. These solutions shifted away not only from the traditional approaches which represented videos using low-level features (e.g., [4, 5]) but also from the use of state-of-the-art detectors designed to identify high-level semantic concepts (e.g., "a killing spree"). The earlier solutions could not carry enough semantic information, and the latter ones have not reached a sufficient level of maturity. Hinted by these recent developments, we adopt here mid-level audio and motion representations as they may help modeling video segments one step closer to human perception. As a basis for the mid-level audio and motion representations, we employ MFCC and dense trajectory features, respectively. Using simultaneously audio and visual information is computationally expensive. We approach this issue by exploiting audio and visual information in a coarse-to-fine setup to reduce computations and boost the velocity of violence detection. In addition, this can be used for designing scalable solutions, i.e., adjustable depending on the processing power or accuracy requirements.

In parallel to the progress in feature representation, machine learning techniques are constantly improved in order to effectively use features. A development in this direction is feature space partitioning [6]. A classifier is usually trained on a given dataset to detect a unique class (e.g., the concept of violence). However, such a class might not be expressed in a "compact" manner in the feature space. Partitioning the feature space to build multiple models that correspond to the same concept might help in properly recognizing a given concept. Therefore, instead of building a unique model to detect violence, we use feature space partitioning. This presents several advantages. It enables a faithful modeling of "violence". It also constitutes a data-driven operation, as it does not require defining manually several "violence" concepts (e.g., there is no need to have a separate concept for "explosion", "fire" or other similar concepts), as it directly builds on the data. Finally, this aspect is not hardwired to "violence" only, but can be extended to other concepts.

The paper is organized as follows. Section 2 explores the recent developments by reviewing video violent content detection methods which have been proposed in the literature, and presents the contributions of the paper. In Section 3, we introduce our method and the functioning of its various components. We provide

and discuss evaluation results obtained on the latest MediaEval datasets of 2014 and 2015 in Section 4. Concluding remarks and future directions to expand our current approach are presented in Section 5.

2. Related Work and Contributions

2.1. Related Work

Although video content analysis has been extensively covered in the literature, violence analysis of movies or of user-generated videos does not enjoy a comparable coverage and is restricted to a few studies. We present here a selection of the most representative ones, from a machine learning and classification perspective. As a preliminary remark, we would like to emphasize that, with respect to prior art studies, the definition of violence poses a difficulty. In some of the works presented in this section, the authors do not explicitly state their definition of violence. In addition, nearly all papers in which the concept is defined consider a different definition of violence; therefore, whenever possible, we also specify the definition adopted in each work discussed in this section.

One popular type of approach adopted in the literature is classification based on SVM models. An illustration to SVM-based solutions is the work by Giannakopoulos et al. [7], where violent scenes are defined as those containing shots, explosions, fights and screams, while non-violent content corresponds to audio segments containing music and speech. Frame-level audio features both from the time and the frequency domain are employed and a polynomial SVM is used as the classifier. In [8], de Souza et al. adopt their own definition of violence, and designate violent scenes as those containing fights (i.e., aggressive human actions), regardless of the context and the number of people involved. Their SVM approach is based on the use of Bag-of-Words (BoW), where local Spatial-Temporal Interest Point Features (STIP) are used as feature representations. They compare the performance of STIP-based BoW with SIFT-based BoW on their own dataset, which contains 400 videos (200 violent and 200 non-violent videos). Hassner et al. [9] present a method for real-time detection of breaking violence in crowded scenes. They define violence as sudden changes in motion in a video footage. The method considers statistics of magnitude changes of flow-vectors over time using the Violent Flows (ViF) descriptor. ViF descriptors are then classified as either violent or non-violent using a linear SVM. In [10], Gong et al. propose a three-stage method. In the first stage, they apply a semi-supervised cross-feature learning algorithm [11] on the extracted audio-visual features for the selection of candidate violent video shots. In the second stage, high-level audio events (e.g.,

screaming, gun shots, explosions) are detected via SVM training for each audio event. In the third stage, the outputs of the classifiers generated in the previous two stages are linearly weighted for final decision. Although not explicitly stated, the authors define violent scenes as those which contain action and violence-related concepts such as gunshots, explosions and screams. Chen et al. [12] proposed a two-phase solution. According to their violence definition, a violent scene is a scene that contains action and blood. In the first phase, where average motion, camera motion, and average shot length are used for scene representation and SVM for classification, video scenes are classified into action and non-action. In the second phase, faces are detected in each keyframe of action scenes and the presence of blood pixels near detected faces is checked using color information. Aiming at improving SVM-based classification, Wang et al. [4] apply Multiple Instance Learning (MIL; MI-SVM [13]) using audio-visual features in order to detect horror. The authors do not explicitly state their definition of horror. Therefore, assessing the performance of their method and identifying the situations on which it properly works is difficult. Video scenes are divided into video shots, where each scene is formulated as a bag and each shot as an instance inside the bag for MIL. In [14], Goto and Aoki propose a violence detection method which is based on the combination of visual and audio features extracted at the segment level using multiple kernel learning.

Next to SVM-based solutions, approaches which make use of other types of learning-based classifiers exist. Yan et al. [15] adopt a Multi-task Dictionary Learning approach to complex event detection in videos. Based on the observation that complex events are made of several concepts, certain concepts useful for particular events are selected by means of combination of text and visual information. Subsequently, an event oriented dictionary is learnt. The experiments are conducted on the TRECVid Multimedia Event Detection dataset. The same authors have experimented Multi-task Learning in other situations. For instance in [16], Yan et al. employ a variant of Linear Discrimant Analysis (LDA – used to find a linear combination of features capable of characterizing or discriminating several classes) called Multi-task LDA to perform multi-view action recognition based on temporal self-similarity matrices. More recently, Yan et al. [17] have developed a Multi-task Learning approach for head-pose estimation in a multi-camera environment under target motion. Giannakopoulos et al. [5], in an attempt to extend their approach based solely on audio cues [7], propose to use a multi-modal two-stage approach based on k nearest neighbors (k-NN). In the first step, they perform audio and visual analysis of segments of one second duration. In the audio analysis part, audio features are used to classify scenes into one of seven

classes (violent ones including shots, fights and screams). In the visual analysis part, motion features are used to classify segments as having either high or low activity. The classifications obtained in this first step are then used to train the k-NN classifier. Another work based on k-NN is the one by Derbas and Quénot [18], where they explore the joint dependence of audio and visual features for violent scene detection (they actually compare k-NN and SVMs and report superior results for k-NN). They first combine the audio and the visual features and then determine statistically joint multi-modal patterns. The proposed method mainly relies on an audio-visual BoW representation. The experiments are performed in the context of the MediaEval 2013 VSD task. The obtained results show the potential of the proposed approach in comparison to methods which use audio and visual features separately, and to other fusion methods such as early and late fusion. In [19], Ionescu et al. address the detection of objective violence in Hollywood movies using Neural Networks, where *objective violence* is defined as "physical violence or accident resulting in human injury or pain" in [2]. The method relies on fusing mid-level violence-related concept predictions inferred from low-level features. The authors employ a bank of multi-layer perceptrons featuring a dropout training scheme in order to construct 10 violence-related concept classifiers. The predictions of these concept classifiers are then merged to construct the final violence classifier. The method is tested on the dataset of the MediaEval 2012 VSD task and ranked first among 34 other submissions, in terms of precision and F-measure. Using Bayesian networks, Penet et al. [20] propose to exploit temporal and multi-modal information for objective violence detection at video shot level. In order to model violence, different kinds of Bayesian network structure learning algorithms are investigated. The proposed method is tested on the dataset of the MediaEval 2011 VSD Task. Experiments demonstrate that both multimodality and temporality add valuable information into the system and improve the performance in terms of the MediaEval cost function [21]. Lin and Wang [22] train separate classifiers for audio and visual analysis and combine these classifiers by co-training. Probabilistic latent semantic analysis is applied in the audio classification part. Audio clips of one second length are represented with mid-level audio features with a technique derived from text analysis. In the visual classification part, the degree of violence of a video shot is determined by using motion intensity, the (non-)existence of flame, explosion and blood appearing in the video shot. Violence-related concepts studied in this work are fights, murders, gunshots and explosions. Ding et al. [23] observe that most existing methods identify horror scenes only from independent frames, ignoring the context cues among frames in a video scene. In order to consider contextual cues in horror

scene recognition, they propose a Multi-view MIL (M^2IL) model based on a joint sparse coding technique which simultaneously takes into account the bag of instances from the independent view and from the contextual view. Their definition of violence is very similar to the definition in [4]. They perform experiments on a horror video dataset collected from the Internet and the results demonstrate that the performance of the proposed method is superior to other existing well-known MIL algorithms.

2.2. Addressed Research Questions and Contributions of this Paper

Based on an in-depth analysis of the literature, we identified two research questions (RQs) that we aim to address with the work presented in this paper.

(RQ1) – The first question is how to model the concept of violence given the data. This question has been qualified as an interesting research direction by the organizers of the VSD challenge [24]. We remarked that, in nearly all of the works mentioned above, a single model is employed to model violence using audio or visual features. In other words, the samples constituting the training set are taken as a whole to train a unique "violence" classifier. Two types of improvements can be envisaged to better model violence. Both are based on the fact that violence can be expressed in very diverse manners. For instance, two distinct events (e.g., "explosion" and "fight") may be both intensely violent in the eyes of a consumer and, nevertheless, be located in different regions of the feature space. In addition, two events of the same type (e.g., "fight") might be characterized by distinct audio or visual features (fight between two individuals *vs.* brawl of 50 people). The first type of improvement is using manually designed multiple violence models, i.e., one model for each possible type of violence (e.g., one for "explosion", one for "fight"). The work by Ionescu et al. [19] is an example. However, such approaches do not solve the latter problem and are hardwired to the violence concept. The second type is deriving subconcepts from the data to build multiple models. Partitioning the feature space, where each "partition" (i.e., cluster) is used to build a separate model, is an illustration. It enables deriving multiple violence models, and can be extended to other concepts.

To the best of our knowledge, the sole work on violent scene detection performing feature space partitioning is the one by Goto and Aoki [14]. Feature space partitioning is achieved through mid-level violence clustering in order to implicitly learn mid-level concepts. However, their work is limited in two aspects. First, they cluster violent samples only. The inclusion of non-violent samples in the training process is done by a random selection. Such an approach

presents the drawback of not taking into account the proximity of some violent and non-violent samples. For instance, if a violent sample and a non-violent one are closely located in the feature space, this is an indication that they are difficult to discriminate. Therefore, in order to obtain optimal classification boundaries, such particularities should be considered when building the models. Second, they use motion features, which are computationally expensive. This does not pose a problem for training. However, such a solution might introduce scalability issues, and might hinder the execution of violence detection in a real-world environment.

(RQ2) – The second question is how to efficiently use powerful motion features. In many of the existing works, next to audio or static visual cues, motion information is also used in the detection of violent scenes. Employed motion features range from simplistic features such as motion changes, shot length, camera motion or motion intensity [5, 12, 20, 22], to more elaborated descriptors such as STIP, ViF [8, 9] or dense trajectories, which have recently enjoyed great popularity. Dense trajectory features [25] have indeed received attention even among the VSD participants (e.g., [26, 27, 28]). Both types of motion approaches have drawbacks and advantages. Simplistic ones do not induce heavy computations but are likely to fail when it comes to efficacy; elaborated ones constitute powerful representations but result from computationally expensive processes. To cope with the heavy computations induced by the use of motion features, we perform a coarse-to-fine analysis, starting with coarse-level analysis with time efficient audio features and pursuing with fine-level analysis with advanced features when necessary. To the extent of our knowledge, none of the studies addressing the detection of violent scenes in videos solve the issue of computational expense using such a staged approach.

To sum up, the contributions of this paper can be summarized as follows: (1) a modeling of violence with feature space partitioning, which reliably models violence in the feature space without extensive supervision, and can be easily transposed for the detection of other concepts; and (2) a coarse-to-fine analysis approach which paves the road for time efficient and scalable implementations.

3. The Proposed Method

In this paper, we address the problem of violence detection at the segment level. This means that no video shot boundaries are available for the videos that we analyze. Therefore, we start our analysis by partitioning all videos into fixed-size

segments of a length of 0.6 second. This length of video segments is determined according to the pre-evaluation runs.

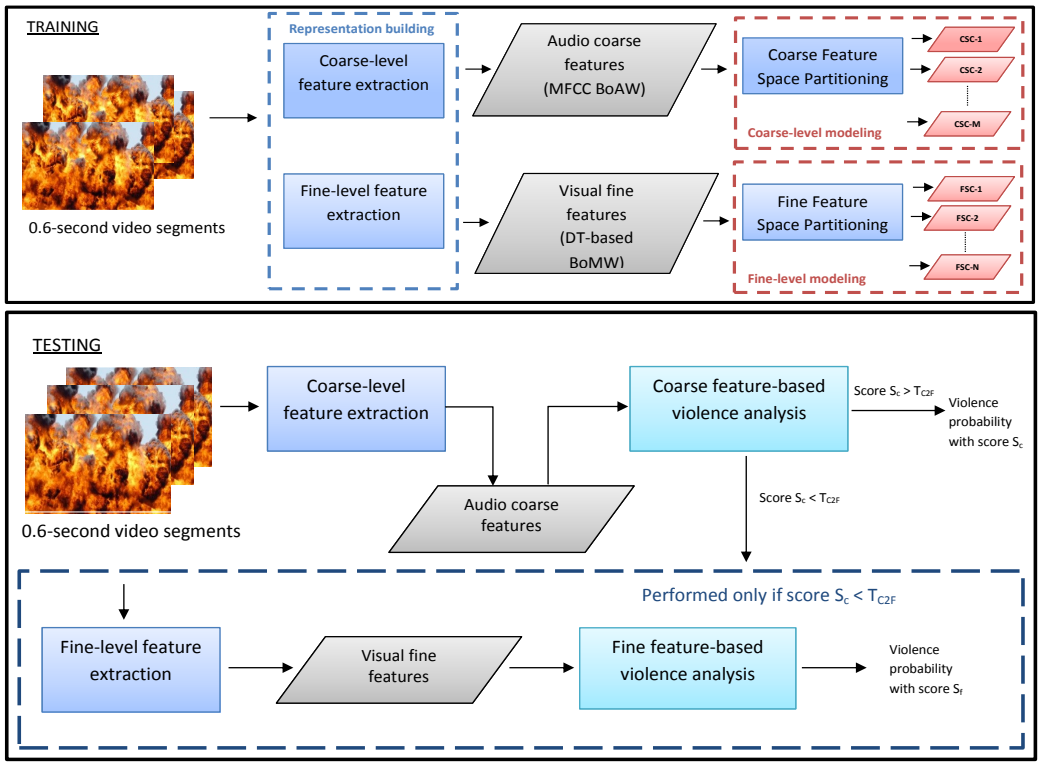


Figure 1: The general overview of our approach illustrating the two main phases of the system. The upper part of the figure gives the main steps performed during training (i.e., coarse and fine-level model generation), while the lower part shows the main steps of testing (i.e., execution). (DT: Dense Trajectory, BoAW: Bag-of-Audio-Words, BoMW: Bag-of-Motion-Words, CSC: Coarse Sub-Concept, FSC: Fine Sub-Concept).

As can be seen in Figure 1, our violence detection approach consists of training and testing phases. The training task involves the extraction of features from the raw video data, which are used to build audio and visual representations, the set of which constitutes a feature space. As indicated earlier, we do not wish to construct a single model obtained with these features, but a plurality of models (i.e., plurality of "subconcepts"), as we argue that "violence" might not be expressed in a "compact" manner in the feature space.

For most machine learning methods, testing would follow training, i.e., it would involve extraction of both audio and visual features followed by classifi-

cation. However, because the extraction of motion features is computationally heavy, our testing does not exactly follow training. Instead, for the execution of our system, a *coarse-to-fine approach* is adopted, which explains the parallel construction of the testing scheme in Figure 1, where coarse-level violence detection is always performed while fine-level analysis is optional.

Therefore, in order to present in more detail each of these steps, the remainder of this section is organized as follows. Section 3.1 deals with video representation. Feature space partitioning is explained in Section 3.2. Section 3.3 details the generation of the subconcepts subsequent to partitioning. The audio or visual violence detections are based on combining the outputs of the models, which is presented in Section 3.4. Temporal smoothing and merging which are designed to further enhance performance are introduced in Section 3.5. Section 3.6 presents the coarse-to-fine analysis.

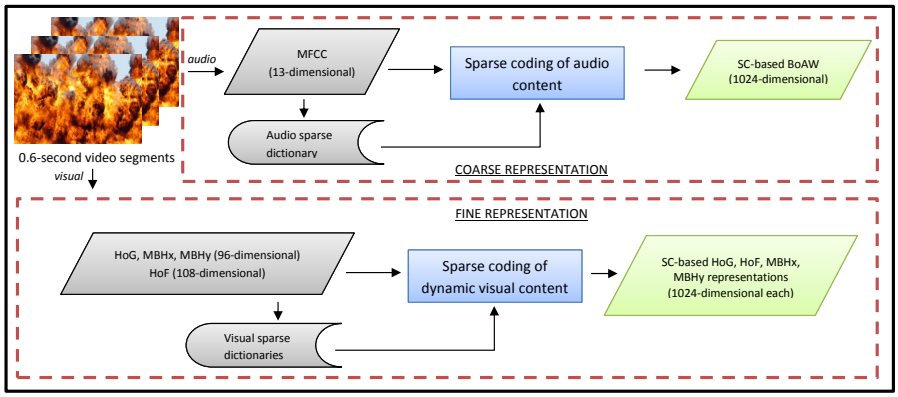


Figure 2: The generation process of audio and visual representations for video segments (upper part: coarse-level analysis features, lower part: fine-level analysis features). Each video segment is of length 0.6 second. Separate dictionaries are constructed and used for MFCC, HoG, HoF, MBHx and MBHy to generate 1024-dimensional representations. Each HoG, MBHx and MBHy descriptor is 96-dimensional, whereas the HoF descriptor is 108-dimensional. (SC: Sparse Coding, BoAW: Bag-of-Audio-Words, BoMW: Bag-of-Motion-Words).

3.1. Representation of Video Segments

In this section, we start introducing our framework by outlining the representation of audio-visual content of videos and present features that we use for *coarse-level* (Section 3.1.1) and *fine-level* (Section 3.1.2) video content analysis.

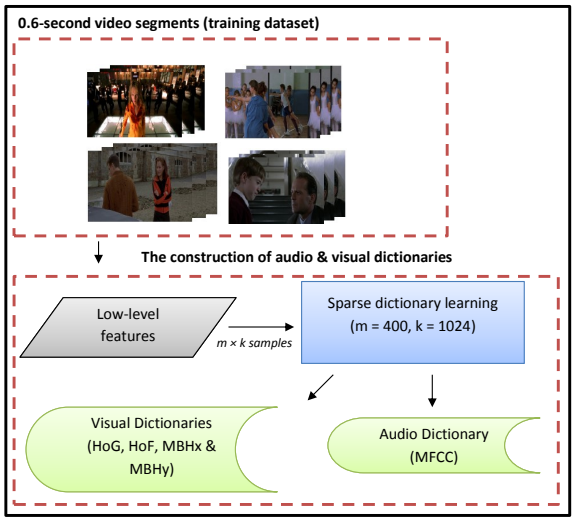


Figure 3: The generation of audio and visual dictionaries with sparse coding. Each video segment is of length 0.6 second. Low-level features are MFCC and dense trajectory features (HoG, HoF, MBHx and MBHy descriptors).

3.1.1. Mid-level Audio Representations

Sound effects and background music in movies are essential for stimulating people's perception [29]. Therefore, the audio signals are important for the representation of videos. Mid-level audio representations are based on MFCC features extracted from the audio signals of the video segments of 0.6 second length as illustrated in Figure 2. We experimentally verified that the 0.6 second time window (i.e., 15 visual frames) was short enough to be computationally efficient and long enough to retain sufficient relevant information. In order to generate the mid-level representations for audio segments, we apply an abstraction process which uses an MFCC-based Bag-of-Audio Words (BoAW) approach with sparse coding (SC) as the coding scheme. We prefer SC over vector quantization in the context of this research as SC was shown to provide more discriminative representations [30]. We employ the dictionary learning technique presented in [31]. The advantage of this technique is its scalability to very large datasets with millions of training samples which makes the technique well suited for our work. In order to learn the dictionary of size k (k = 1024 in this work) for sparse coding, $400 \times k$ MFCC feature vectors are sampled from the development data (experimentally determined figure). The construction of the SC-based audio dictionary is illustrated in Figure 3, where the steps are illustrated. In the coding phase, we construct the sparse

representations of audio signals by using the LARS algorithm [32] due to its efficiency in preliminary evaluations. Given an audio signal and a dictionary, the LARS algorithm returns sparse representations for MFCC feature vectors. In order to generate the final sparse representation of audio segments which are a set of MFCC feature vectors, we apply the *max-pooling* technique.

3.1.2. Mid-level Dynamic Visual Representations

Dynamic visual content of videos provides complementary information for the detection of violence in videos. The importance of motion in edited videos (e.g., movies and Web videos) motivated us to incorporate motion information to our framework. To this end, we adopt the work of Wang et al. on dense trajectories [25]. Improved dense trajectories are dynamic visual features which are derived from tracking densely sampled feature points in multiple spatial scales. Although initially used for unconstrained video action recognition [25], dense trajectories constitute a powerful tool for motion or video description, and, hence, are not limited to action recognition.

Our dynamic visual representation works as follows. First, dense trajectories [25] of length 15 frames are extracted from each video segment. The sampling stride, which corresponds to the distance by which extracted feature points are spaced, is set to 20 pixels due to time efficiency concerns. Dense trajectories are subsequently represented by a histogram of oriented gradients (HoG), histogram of oriented optical flow (HoF) and motion boundary histograms in the x and y directions (MBHx and MBHy). The sparse dictionary learning and coding phases are performed similarly to the audio case. For each dense trajectory descriptor (i.e., each one of HoG, HoF, MBHx and MBHy), we learn a separate dictionary of size k (k = 1024 in this work) by sampling 400 $\times$ k feature vectors from the development data. Finally, sparse representations are constructed using the LARS and *max-pooling* algorithms. The construction of the SC-based motion dictionaries is also summarized in Figure 3.

3.2. Feature Space Partitioning

As discussed before, "violence" is a concept which can be expressed in diverse manners. For instance, in a dataset, both explosions and fight scenes are labeled as violent according to the definition that we adopted. However, these scenes might highly differ from each other in terms of audio-visual appearance depending on their characteristics of violence. Instead of learning a unique model for violence detection, learning multiple models constitutes a more judicious choice. Therefore, in the learning phase, a "divide-and-conquer" (*"divide-et-impera"*) approach

is applied by performing feature space partitioning. The first step of learning multiple violence subconcept models is to partition the feature space into smaller portions. We perform partitioning by clustering the set of video segments of 0.6 second length in our development dataset. Moreover, we employ the Approximate Nearest Neighbor (ANN) k-means algorithm [33] which is a variant of Lloyd's algorithm [34] particularly suited for very large clustering problems [35]. The ANN algorithm computes approximated nearest neighbors to speed up the instance-to-cluster-center comparisons. We use the Euclidean metric for distance computations, initialize cluster centers with the *k*-means++ algorithm [36] and repeat *k*-means clustering several times before determining data clusters. The number of iterations represents a trade-off between clustering quality and time efficiency. We observed that after 8 iterations stable clusters were obtained; increasing this number did not improve significantly the quality of the obtained clusters. By applying feature space partitioning, we infer (sub)concepts in a data-driven manner as opposed to approaches (e.g., [19]) which use violence-related concepts as a mid-level step.

3.3. Model Generation for Subconcepts

For the generation of subconcept models, we apply the following procedure. After clusters are generated (Section 3.2), they are distinguished as *atomic* and *non-atomic* clusters as in [37]. A cluster is defined as atomic, if it contains patterns of the same type, i.e., patterns which are all either "violent" or "non-violent". No model generation is realized for atomic clusters and their class labels are stored for further use in the test phase. Non-atomic clusters are clusters which include patterns from both the violent and non-violent classes. For those non-atomic clusters, a different model is built for each violence subconcept (i.e., cluster). As the base classifier in order to learn violence models, we use a two-class SVM. An overview of the generation of the violence models is presented in Figure 4.

In the learning step, the main issue of the two-class SVM is the problem of imbalanced data. This is caused by the fact that the number of non-violent video segments is much higher than the number of violent ones in the development dataset. This phenomenon causes the learned boundary being too close to the violent instances. Consequently, the SVM has a natural tendency towards classifying every sample as non-violent. In order to cope with this bias, different strategies to "push" this decision boundary towards the non-violent samples exist. Although more sophisticated methods dealing with the imbalanced data issue have been proposed in the literature (see [38] for a comprehensive survey), we choose, in the current framework, to perform random undersampling to balance the number of

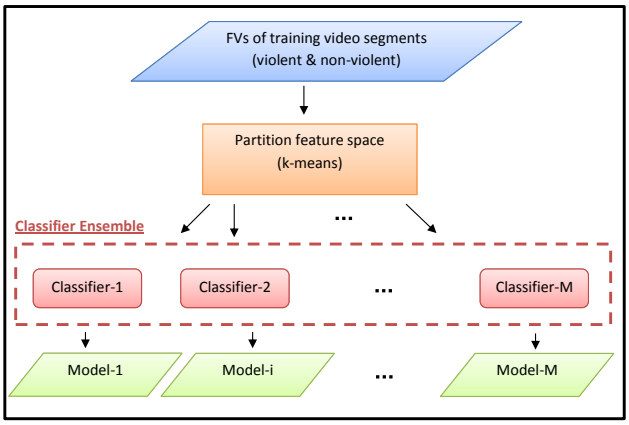


Figure 4: The generation of violence detection models with feature space partitioning through k-means clustering. Each video segment is of length 0.6 second. Feature vectors (i.e., either coarse or fine features) of the training segments are given as input to the combination process. (FV: Feature Vector).

violent and non-violent samples (with a balance ratio of 1:2). This method proposed by Akbani et al. [39] appears to be particularly adapted to the application context of our work. In [39], different under and oversampling strategies are compared. According to the results, SVM with the undersampling strategy provides the most significant performance gain over standard two-class SVMs. In addition, the efficiency of the training process is improved as a result of the reduced development data and, hence, training is easily scalable to large datasets similar to the ones used in the context of our work.

3.4. Combining Predictions of Models

In the test phase, the main challenge is to combine the classification results (i.e., probability outputs) of the violence models. In order to achieve the combination of classification results, we perform one of two different combination methods, namely *classifier selection* and *classifier fusion*, which are alternative solutions. Normally, in a basic SVM, only class labels or scores are output. The class label results from thresholding the score, which is not a probability measure. The scores output by the SVM are converted into probability estimates using the method explained in [40].

An overview of the combination methods is presented in Figure 5. Both methods follow the main canvas of Figure 5, i.e., they get feature vectors as input and return a violence score; the dotted frames highlight the specificities of each of

them. In order to determine the clip-level violence score of a video sample (i.e., to assign one violence score per video), we use the maximum violence score of the video segments in the sample.

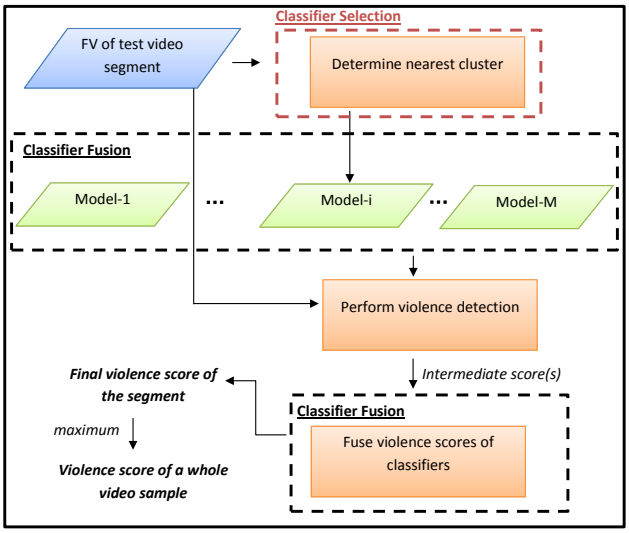


Figure 5: An overview of the classifier decision combination phase of our method. Each video segment is of length 0.6 second. Feature vector (i.e., either coarse or fine features) of the test segment is given as input to the combination process. (FV: Feature Vector).

3.4.1. Classifier Selection

In the *classifier selection* method, we first determine the nearest cluster to a test video segment using the Euclidean distance measure, i.e., the cluster minimizing the following Euclidean distance is identified:

$$d(c_i, x_j) = \|c_i - x_j\| \tag{1}$$

where c_i represents a given cluster center, and x_j a video segment. If the nearest cluster is an atomic cluster, then the test video segment is labeled with the unique label of the cluster and the probability measure is set to 1.0. For the non-atomic cluster case, once the best fitting classifier for the video sample is determined, the probability output of the corresponding model is used as the final prediction for that video sample.

3.4.2. Classifier Fusion

In the *classifier fusion* method, we combine the probability outputs of all cluster models (i.e., both atomic and non-atomic clusters). As in the classifier selection

method, the probability measure is set to 1.0 for atomic clusters. The classifiers that we adopt are all SVMs. Hence, we are in the presence of homogeneous "learners" (i.e., all of the same type) according to the terminology of [6]. In such a situation, it is advised to directly fuse the violence probabilities ($h_i(x_j)$) generated by each of the classifiers (i.e., "learners") using the *weighted soft voting* technique [6]:

$$H(x_j) = \sum_{i=1}^{T} w_{ij} h_i(x_j) \tag{2}$$

As shown in Equation 2, a classifier-specific weight (w_{ij}) is dynamically assigned to each classifier for each test video segment (x_j) using the Euclidean distance of the segment to the cluster centers. The weights assigned to the clusters are determined such that they always sum up to 1.

3.5. Temporal Smoothing and Merging

As mentioned earlier, we split videos into small segments of length 0.6 second. However, a violent scene may be composed of several continuous segments. While some of these segments can easily be identified as violent, others might be more challenging to detect. Previous findings support that, if a segment contains violence, its neighboring segments are likely to contain violence as well and that, consequently, temporal score smoothing is likely to boost the performance. Therefore, we perform the post-processing steps of (1) temporal smoothing and (2) segment merging, in order to further improve the performance of the system.

The temporal smoothing technique we adopt consists in applying a simple yet efficient score smoothing, where the smoothed violence prediction score of a segment is determined as the average of the scores in a window of three consecutive segments.

Our segment merging method is based on the use of a threshold value ($T_{violence}$) which is determined experimentally on the development dataset. We merge two neighboring segments, if they are both identified as violent or non-violent (i.e., both of their violence scores are above or below $T_{violence}$) and set the new violence prediction score as the average of the related segments.

3.6. Coarse-to-Fine Violence Analysis

The inclusion of a coarse-to-fine analysis in the whole process originated from the observation that the extraction of dense trajectory features [25] is a computationally expensive process, as it involves computing and tracking features over several frames.

Various precautions could be taken to cope with this issue. Some straightforward solutions include for instance: Resizing frames; tuning parameters, i.e. using an increased step size; considering only a subset of the dense trajectory features. However, the gain in computation time obtained through such measures comes at the expense of decreased accuracy. We therefore developed an alternative solution in the form of coarse-to-fine classification.

We observed that, for the task of violence detection, audio is an extremely discriminative feature. A violence detection approach for video analysis based solely on audio features (e.g., MFCC) will normally fail only if the video contains no sound or if the volume is low. When, according to audio features, a segment is classified as violent, we can realistically assume that this "violent" label is correct. However, if it is classified as non-violent, then a verification by the use of "advanced" visual features (i.e., dense trajectory based Bag-of-Motion-Words) would be necessary to confirm the absence of violence.

From a practical point of view, the implementation of violence detection (i.e., execution of the system during test) follows the scheme under the lower part of Figure 1 (testing). First of all, coarse detection based on audio features is performed. MFCC features are extracted and converted to mid-level features as described under Section 3.1. Coarse-level analysis is further performed in line with the teachings of Sections 3.4 and 3.5. During coarse analysis, a segment is assigned a score (S_c) which is compared to a threshold T_{C2F} (coarse-to-fine analysis threshold). If S_c exceeds this threshold T_{C2F}, the segment is labeled as violent with the score S_c. If not, the fine analysis based on advanced visual features is initiated. The fine-level visual features are extracted and converted to mid-level features and visual feature analysis is run. The outcome is a score S_f, which is compared to the threshold $T_{violence}$ (threshold mentioned in Section 3.5) to determine if the segment is violent.

4. Performance Evaluation

The experiments presented in this section aim at comparing the discriminative power of our method based on feature space partitioning (referred hereafter as "FSP") against the method employing a unique model and also at highlighting the advantages brought by the coarse-to-fine analysis. We also evaluate the performance of audio and visual features at the different levels (coarse and fine) described in Section 3.1. Because of potential differences in the definition of "violence" adopted in published works discussed in Section 2, a direct comparison of our results with those works is not always meaningful. Nevertheless, we can

compare our approach with the methods which took part to the MediaEval 2014 and 2015 VSD tasks (segment-level detection for 2014, and clip-level detection for 2015), where the same "violence" definition and datasets are employed.

Using the evaluation framework provided by the MediaEval VSD task is an opportunity to test our algorithms in a standardized manner on a standardized data corpus. Although running since 2011, the MediaEval VSD task reached a certain level of maturity only in 2014, when the organizers opted for the subjective definition of violence, and for the use of the mean average precision metric. The same definition and evaluation metric were kept for the 2015 edition. For these reasons, we show our results on the latest two editions of the the MediaEval VSD task (2014 and 2015). The 2015 dataset differs from the 2014 data in difficulty, as can be seen from the results (see below).

The MediaEval 2014 VSD challenge is structured as two separate tasks: A *main task* which consists in training on Hollywood movies only and testing on Hollywood movies only; and a *generalization task* which consists in training on Hollywood movies only and testing on Web videos only. The MediaEval 2015 VSD structure is slightly different than the task of 2014 in terms of dataset and consists of only one task. In addition, violence detection is performed at the clip-level (i.e., only one violence score is assigned for each video in the dataset).

4.1. Dataset and Ground Truth

For the evaluation within the context of the **MediaEval 2014 VSD task**, we used two different types of dataset in our experiments: (1) a set of 31 movies which were the movies of the MediaEval 2014 VSD task (referred hereafter as the "Hollywood movie dataset"), and (2) a set of 86 short YouTube Web videos under Creative Commons (CC) licenses which were the short Web videos of the MediaEval 2014 VSD task (referred hereafter as the "Web video dataset").

A total of 24 movies from the Hollywood set are dedicated to the development process: *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, The Bourne Identity, The Wizard of Oz, Dead Poets Society, Fight Club, Independence Day, Fantastic Four 1, Fargo, Forrest Gump, Legally Blond, Pulp Fiction, The God Father 1* and *The Pianist*. The remaining 7 movies – *8 Mile, Brave Heart, Desperado, Ghost in the Shell, Jumanji, Terminator 2* and *V for Vendetta* – and the Web video dataset serve as the test set for the main and generalization task, respectively.

Each movie and short Web video is split in a multitude of video segments, where each video segment is of length 0.6 second as exposed in Section 3. The

development set (24 movies) consists of 289,699 video segments, whereas the movie test set (7 movies) consists of 83,350 video segments and the Web video dataset consists of 15,747 such short video segments, where each segment is labeled as *violent* or *nonviolent*. The movies of the development and test sets were selected in such a manner that both development and test data contain movies of variable violence levels (ranging from extremely violent movies to movies without violence) from different genres and production years ranging from 1939 (*The Wizard of Oz*) to 2007 (*I am Legend*). On average, around 14.45% of segments are annotated as violent in the whole dataset (i.e., the Hollywood movie and Web video datasets).

The ground truth of the Hollywood dataset was generated by 9 human assessors, partly by developers, partly by potential users. Violent movie and Web video segments are annotated at the frame level. For the generalization task, the ground truth was created by several human assessors[2] who followed the subjective definition of violence as explained in [2]. A detailed description of the Hollywood and the Web video datasets, and the ground truth generation are given in [3].

In addition to the datasets provided by the MediaEval 2014 VSD task, we also performed our evaluations on the dataset provided by the **MediaEval 2015 VSD task** (referred hereafter as the "VSD 2015 movie dataset"). The development and test sets are completely different from the dataset of the MediaEval 2014 VSD task. The VSD 2015 movie dataset consists of 10,900 video clips (each clip lasting from 8 to 12 seconds approximately) extracted from about 199 movies, both professionally edited and amateur movies. The movies in the dataset are shared under CC licenses that allow redistribution. The VSD 2015 development dataset contains 6,144 video clips, whereas the test set has 4,756 clips. As in the MediaEval 2014 VSD evaluation, each movie clip is split in a multitude of video segments, where each video segment is of length 0.6 second. On average, around 4.61% of segments are annotated as violent in the whole dataset (i.e., development and test sets). The ground-truth generation process of the VSD 2015 movie dataset is similar to the VSD task of 2014. The details of the ground-truth generation are explained in [41].

[2]Annotations were made available by *Fudan University*, *Vietnam University of Science*, and *Technicolor*.

4.2. Experimental Setup

We employed the MIR Toolbox v1.6.1[3] to extract the MFCC features (13-dimensional). Frame sizes of 25 ms with a step size of 10 ms are used. Audio-visual features are extracted as explained in Section 3.

The SPAMS toolbox[4] is employed in order to compute sparse codes which are used for the generation of the mid-level audio and dynamic visual representations. The VLFeat[5] open source library is used to perform k-means clustering (k ranging from 5 to 40 in this work).

The two-class SVMs were trained with an RBF kernel using libsvm[6] as the SVM implementation. Training was performed using audio and visual features extracted at the video segment level. SVM parameters were optimized by 5-fold cross-validation on the development data. All video segments belonging to a specific video were always either in the training set or in the test set during cross-validation. Zero mean unit variance normalization was applied on feature vectors of development dataset.

We used the *average precision (AP)* as the evaluation metric which is the official metric of the MediaEval 2014 and 2015 VSD tasks. The mean of AP (MAP) values on the MediaEval 2014 and 2015 datasets are provided in the results.

4.3. Results and Discussion

The evaluation of our approach is achieved from different perspectives: (***i***) comparison to unique concept modeling; (***ii***) comparison to MediaEval 2014 participants; (***iii***) comparison to MediaEval 2015 participants; and (***iv***) added-value of the coarse-to-fine analysis. The experiments (***i***) and (***iv***) pertain to the research questions (RQ1, RQ2) identified in Section 2 (assessment of FSP and coarse-to-fine analysis, respectively), while (***ii***) and (***iii***) provide a benchmark of our violence detection framework against existing solutions.

Experiment (*i*) reviews the gain in classification performance brought by FSP. In this case, the baseline method for comparison is the approach using single a SVM classifier trained with the same data, i.e., a single model for violence. The results are summarized in Tables 1, 2 and 3 which provide a comparison of the

[3] `https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox`

[4] `http://spams-devel.gforge.inria.fr/`

[5] `http://www.vlfeat.org/`

[6] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

FSP method against the unique violence detection model (no FSP) in terms of MAP metric on the Hollywood movie dataset (Table 1), on the Web video dataset (Table 2) and on the VSD 2015 movie dataset (Table 3), respectively. For the FSP method, evaluated k values for the MediaEval 2014 datasets are 10, 20 and 40, whereas these values range from 5 to 20 for the MediaEval 2015 dataset. The conclusions which can be drawn from Table 1 are as follows:

- The highest MAP value of 0.52 is achieved with a visual-analysis method for the situation where the feature space is split in $k = 20$ clusters and classifier scores are fused.

- Another observation is that the MAP values of the FSP method are usually higher, when classifier fusion is the score combination method.

- One final observation about the results is about the choice of the number of clusters for the FSP method. When considering classifier fusion, reducing the number of clusters from 20 to 10 reduces performance. This shows that a lower number of clusters does not provide a faithful representation of the development dataset. Increasing the number of clusters to 40 does not help in obtaining a better accuracy either. This is an indication that an excessive number of clusters results in clusters containing a limited number of training samples, which prevents them from correctly representing the data.

The important observations from Table 2 are summarized as follows:

- We observe that generally the results show little variability. For Web videos, FSP methods outperform unique SVM-based solutions, when comparing approaches using the same modality (i.e., audio or visual).

- More importantly, for Web videos, selection based combinations perform better than fusion based combinations, in general. This differs from the results obtained on Hollywood movies (Table 1), where fusion based combinations provided better outcomes. A plausible explanation is that, in movies, violent scenes are long, and often correspond to several subconcepts, while user-generated Web videos are short, and, therefore, are likely to correspond to a single subconcept.

The conclusions which can be drawn from Table 3 are as follows:

Table 1: The MAP of the FSP method with coarse and fine representations, k clusters (k = 10, 20 and 40) and different classifier combination methods (classifier selection and classifier fusion) and an SVM-based unique violence detection model (without feature space partitioning) on the **Hollywood movie dataset**. (MAP: mean average precision).

Method	**MAP**
FSP method (visual, fusion, k = 10)	0.40
FSP method (visual, fusion, k = 20)	**0.52**
FSP method (visual, fusion, k = 40)	0.39
FSP method (audio, fusion, k = 10)	0.36
FSP method (audio, fusion, k = 20)	0.43
FSP method (audio, fusion, k = 40)	0.34
FSP method (visual, selection, k = 10)	0.42
FSP method (visual, selection, k = 20)	0.31
FSP method (visual, selection, k = 40)	0.29
FSP method (audio, selection, k = 10)	0.36
FSP method (audio, selection, k = 20)	0.29
FSP method (audio, selection, k = 40)	0.28
Unique model (visual)	0.32
Unique model (audio)	0.29

- The highest MAP value of 0.2947 is achieved with a visual-analysis method for the situation where the feature space is split in $k = 10$ clusters and classifier scores are fused.

- Globally, for the videos in the VSD 2015 movie dataset, selection-based combinations perform better than fusion-based combinations as in the Web videos (Table 2). This differs from the results obtained on Hollywood movies (Table 1). A possible explanation is that, in the VSD 2015 movies, violent scenes correspond to single events (e.g., a fight) rather than complicated events, and, therefore, are likely to correspond to a single subconcept.

- When compared to the MAP values reported in Table 1 and 2, the MAP values are lower. This characteristic is independent from our approach. While the best runs of the VSD 2014 participants could reach a MAP of 0.6, those of 2015 could not exceed 0.3. The 2015 dataset is, hence, more difficult. We think that the extra difficulty of the 2015 dataset is caused by two factors. First, the VSD 2015 movie dataset is not originally selected

Table 2: The MAP of the FSP method with coarse and fine representations, k clusters (k = 10, 20 and 40) and different classifier combination methods (classifier selection and classifier fusion) and an SVM-based unique violence detection model (without feature space partitioning) on the **Web video dataset**. (MAP: mean average precision).

Method	**MAP**
FSP method (visual, fusion, k = 10)	0.61
FSP method (visual, fusion, k = 20)	0.62
FSP method (visual, fusion, k = 40)	0.61
FSP method (audio, fusion, k = 10)	0.63
FSP method (audio, fusion, k = 20)	0.59
FSP method (audio, fusion, k = 40)	0.58
FSP method (visual, selection, k = 10)	**0.64**
FSP method (visual, selection, k = 20)	**0.64**
FSP method (visual, selection, k = 40)	0.63
FSP method (audio, selection, k = 10)	**0.64**
FSP method (audio, selection, k = 20)	0.61
FSP method (audio, selection, k = 40)	0.56
Unique model (visual)	0.56
Unique model (audio)	0.42

for violence concept analysis; the violence level of the dataset is, therefore, quite low (around 4% of the clips). Second, the violence concept is less "emphasized" by the film editing rules which are usually used in the 2014 Hollywood movies.

Experiment (*ii*) aims at comparing our approach with the MediaEval submissions of 2014 [42]. Table 4 provides a comparison of our best performing FSP method with the best run of participating teams (in terms of MAP) in the MediaEval 2014 VSD task. If we look at the results, we notice that there is a pattern: all the solutions perform better for the generalization task, except Fudan-NJUST. For a fair evaluation, we compare our results only against the teams who submitted results for both tasks; in this case, the best systems were the ones from Fudan-NJUST and FAR. Fudan-NJUST and FAR ranked first for the main task (with a MAP of 0.63) and for the generalization task (with a MAP of 0.664), respectively.

For the main task, our FSP solution (visual with classifier fusion and $k = 20$) achieves results close to *run4* of Fudan-NJUST, i.e., our results are competing with state of the art results. However, a closer look at the short paper describing the Fudan-NJUST system [48] reveals that the Fudan-NJUST team used more

Table 3: The MAP of the FSP method with coarse and fine representations, k clusters (k = 5, 10 and 20) and different classifier combination methods (classifier selection and classifier fusion) and an SVM-based unique violence detection model (without feature space partitioning) on the **VSD 2015 movie dataset**. (MAP: mean average precision).

Method	**MAP**
FSP method (visual, fusion, k = 5)	0.1114
FSP method (visual, fusion, k = 10)	0.1832
FSP method (visual, fusion, k = 20)	0.0916
FSP method (audio, fusion, k = 5)	0.1121
FSP method (audio, fusion, k = 10)	0.1036
FSP method (audio, fusion, k = 20)	0.0897
FSP method (visual, selection, k = 5)	0.2496
FSP method (visual, selection, k = 10)	**0.2947**
FSP method (visual, selection, k = 20)	0.1236
FSP method (audio, selection, k = 5)	0.1605
FSP method (audio, selection, k = 10)	0.1913
FSP method (audio, selection, k = 20)	0.0932
Unique model (visual)	0.2068
Unique model (audio)	0.1868

Table 4: The MAP for the best run of teams in the MediaEval 2014 VSD Task [42] and our best performing method on the Hollywood movie and Web video datasets. * = at least one participant member of the MediaEval VSD organizing team. ** = provided by the organizing team. (NA: Not Available)

Team	**MAP - Movies**	**MAP - Web**
Fudan-NJUST [26]*	**0.63** (*run4*)	0.604 (*run2*)
NII-UIT [28]*	0.559	NA
FSP (visual, fusion, k = 20)	**0.52**	**0.62**
FAR [43]*	0.451 (*run1*)	**0.664** (*run3*)
FSP (visual, selection, k = 10)	**0.42**	**0.64**
MIC-TJU [44]	0.446	0.566
RECOD [27]	0.376	0.618
ViVoLab-CVLab [45]	0.178	0.43
TUB-IRML [46]	0.172	0.517
*Random run (baseline) **	0.061	0.364
MTM [47]	0.026	NA

features than our system: not only the 4 visual and 1 audio features that we used here, but also 6 additional ones (STIP; Fisher-encodings of HoG, HoF, MBHx, MBHy and trajectory shape). Hence, we argue that the performance difference can be explained by the inclusion of more features, which also results in larger complexity; there also seems to be strong evidence in the literature supporting our belief that the performance difference is caused by the different feature set. In a paper by members of the Fudan-NJUST team [49], the accuracy obtained with larger feature sets is reported to be better. In addition, the MAP of 0.63 of *run4* was achieved by fusing SVM with deep learning methods (the one-classifier-type *runs 1* and *2* were below 0.454). In contrast, we used only one type of classifiers (SVM).

For the generalization task, our solutions (visual with classifier selection and $k = 10$ or 20, audio with classifier selection and $k = 10$) achieve results extremely close to *run3* of the FAR team, which ranked first. We achieved 0.64 while their *run3* achieved 0.664.

The comparison to other MediaEval participants outlined above clearly demonstrates excellent results already. However, when considering the **aggregated** results of a given run (i.e., the MAP on the main task and the MAP on the generalization task **for a given run**), our FSP solution with visual analysis, fusion and $k = 20$ (0.52 - 0.62) outperforms *runs 2* (0.454 - 0.604) and *4* (0.63 - 0.5) of Fudan-NJUST and *runs 1* (0.45 - 0.578) and *3* (0.25 - 0.664) of FAR. The conclusion which can be drawn from this aggregated comparison is that our best performing setup (visual analysis, fusion and $k = 20$) provides more stable results, i.e., the performance in real-world scenarios will be more predictable.

Experiment (*iii*) aims at comparing our approach with the MediaEval submissions of 2015 [50]. Table 5 provides a comparison of our best performing FSP method with the best run of participating teams (in terms of MAP) in the MediaEval 2015 VSD task (i.e., Affective Impact of Movies – including Violence – task). The following conclusions can be drawn from the results in Table 5.

- Our FSP solution (visual with classifier selection and $k = 10$) achieves results on par with *run5* of Fudan-Huawei, i.e., our results are state of the art level. The Fudan-Huawei team employs learned spatio-temporal and violence-specific representations using convolutional neural network architectures.

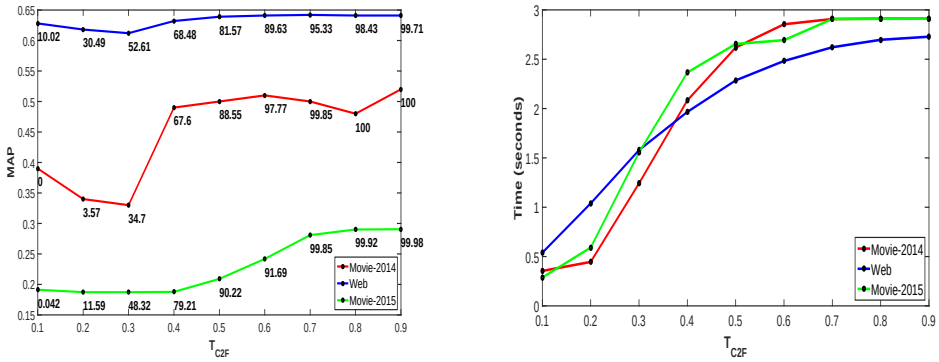
- The FSP solution where we employ only MFCC-based audio representa-

tions achieves also very promising results (with a MAP of 0.1913) compared to the participating teams of the MediaEval 2015 VSD task. In addition, all solutions are above the baseline methods (i.e., random and trivial) provided by the MediaEval VSD organizing team in terms of MAP.

Table 5: The MAP for the best run of teams in the MediaEval 2015 VSD Task [50] and our best performing method using visual and audio representations on the VSD 2015 movie dataset. * = at least one participant member of the MediaEval VSD organizing team. ** = provided by the organizing team.

Team	**MAP**
Fudan-Huawei [51]	**0.2959** (*run5*)
FSP method (visual, selection, k = 10)	**0.2947**
MIC-TJU [52]*	0.2848 (*run1*)
NII-UIT [53]*	0.2684 (*run5*)
RUCMM [54]	0.2162 (*run4*)
FSP method (audio, selection, k = 10)	**0.1913**
ICL-TUM-PASSAU [55]	0.1487 (*run4*)
RFA [56]*	0.1419 (*run4*)
KIT [57]	0.1294 (*run5*)
RECOD [58]	0.1143 (*run1*)
UMons [59]	0.0967 (*run1*)
TCS-ILAB [60]	0.0638 (*run2*)
*Random (baseline)***	0.0511
*Trivial (baseline)***	0.0486

Experiment (*iv*) provides results for the coarse-to-fine analysis on the MediaEval datasets of 2014 and 2015 (Figure 6). As mentioned in detail in Section 3.6, coarse-level analysis is run for a given segment, and if the violence score for that segment is below a threshold (T_{C2F}), fine-level analysis is also run. We repeat the experiment with different values for the threshold, ranging from 0.1 to 0.9 (a threshold of 0 is equivalent to coarse-level analysis only, while a threshold of 1 is equivalent to fine-level analysis only). We select the best detectors according to experiment (*i*), where the coarse detector is audio-based and the fine detector is visual-based. For higher threshold values, visual analysis is run on a higher number of segments. Conversely, for lower threshold values, visual analysis is run on a lower number of segments. We drew the following conclusions on the coarse-to-fine analysis on different MediaEval datasets.



(a) (b)

Figure 6: **(a)** Plot of the coarse-to-fine analysis threshold (T_{C2F}) vs MAP. The numbers indicated next to the points in the graph correspond to the percentage of segments for which visual analysis is performed. **(b)** Average computational time per video segment of coarse-to-fine analysis with respect to the threshold (T_{C2F}), where the computational time includes raw feature extraction, feature encoding and classification. (Movie-14: the Hollywood movie dataset, Web: the Web video dataset, Movie-15: the VSD 2015 movie dataset).

- On the Hollywood movie dataset, we see that setting a threshold value equal to 0.4 provides a MAP of 0.49. In other words, running coarse audio analysis with a threshold of 0.4, helps in drastically reducing the total computations for visual analysis; for such a threshold, we indeed observe that visual analysis is executed on only 67.6% of the segments. Such a result means that running audio and visual analysis on 67.6% of the segments, provides results which are almost as good as visual analysis running independently. Threshold values between 0.5 and 0.7 also return very accurate classifications, but for these values the gain in computation time is rather limited.

- For Web videos, the gain is even more pronounced: From Figure 6(a), we also observe that setting a threshold value equal to 0.1 provides a MAP of 0.628 on the Web video dataset, while the visual analysis is performed only on 10.02% of the segments.

- Concerning the results on the VSD 2015 movie dataset, as shown in Figure 6(a), in order to achieve MAP values closer to the highest MAP value of the proposed framework (i.e., 0.2947), we need to run visual analysis in addition to the audio analysis on more than 90% of the segments. We can realistically assume that this is due to the expression of violence in the

movies of the VSD 2015 dataset which is mainly in terms of visual features other than special audio effects which are usually used in the Hollywood movies.

- Finally, we observe from Figure 6(b) that, on average, coarse-to-fine analysis can provide a significant gain in execution time, especially for Web videos (with a threshold of 0.1, 5 times faster and quasi-identical performance, when compared to fine analysis).

4.4. Computational Complexity

In this section, we provide an evaluation of the time complexity and computational time of our system. We present measures for the two main components of the system: (1) feature generation, and (2) model learning. All the timing evaluations were performed with a machine with 2.40 GHz CPU and 8 GB RAM. For feature generation, Figure 7 presents average computational times calculated on the MediaEval 2014 and 2015 development datasets for raw feature extraction, sparse dictionary learning and feature encoding. The raw feature extraction part is the most time-consuming part of the whole component. Especially, the extraction time of dense trajectory feature descriptors is 14 times higher than that of MFCC descriptors.

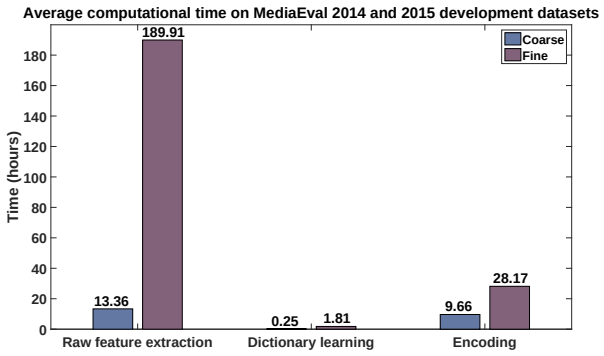


Figure 7: Average computation times (in hours) of coarse-level (i.e., MFCC-based BoAW) and fine-level (i.e., DT-based BoMW) features on the development datasets of MediaEval VSD 2014 and 2015. *Raw feature extraction*: extraction of raw descriptors from audio and visual signals. *Dictionary learning*: the generation of sparse dictionaries. *Encoding*: the feature encoding phases for coarse and fine features (BoAW: Bag-of-Audio-Words, BoMW: Bag-of-Motion-Words).

Concerning model learning, Figure 8 provides a computational time comparison of the FSP method against unique modeling whose classification perfor-

mances are discussed in detail in the previous section (Section 4.3). For the MediaEval 2014 dataset, the k value used for FSP model generation is 20, whereas k is 10 for the MediaEval 2015 dataset. As presented in Figure 8, the model generation time is drastically reduced by using the FSP method. In simple words, this shows that training multiple SVMs (with RBF kernels) using different parts of the training data is faster than training a single one with the whole data. In conclusion, besides improved accuracies, FSP provides an advantage in training time.

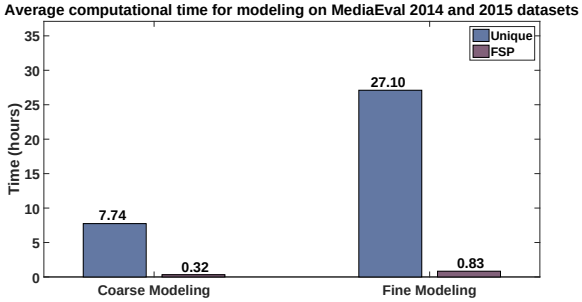


Figure 8: Average computation times (in hours) of coarse-level and fine-level model generation with unique concept modeling and FSP method, using the development datasets of MediaEval VSD 2014 and 2015. (FSP: Feature Space Partitioning)

5. Conclusions and Future Work

In this paper, we introduced an efficient approach for the detection of violent contents in movies and Web videos. We adopted mid-level audio and visual features in a Bag-of-Words fashion. To boost performance, feature space partitioning was included. This was applied on audio and on visual feature spaces, for different number of clusters, and each cluster was used to build a model designed to detect a particular violence subconcept. The combination of the classification results was realized under the selection and fusion mechanisms.

The results obtained on the standardized MediaEval dataset of 2014 demonstrated that our system competes with state of the art detectors for the main and generalization tasks taken separately. When evaluating a given solution on both tasks simultaneously, we outperform state of the art detectors, which implies that our solutions constitute more stable violence detectors (i.e., their performance can be better predicted in real world situations). The results obtained on the latest MediaEval dataset of 2015 also demonstrated that our approach is on par with the state-of-the-art.

Finally, in an attempt to develop a solution which can execute promptly, a coarse-to-fine analysis was introduced. This has shown that an important gain in computation time can be achieved, without sacrificing accuracy.

We plan to further investigate the feature space partitioning and coarse-to-fine components to enhance the classification performance. An interesting research question is to assess whether augmenting the coarse analysis feature set by including computationally efficient audio-visual representations brings a further gain in computation time while keeping comparable accuracy. Another research direction is automating the selection of the number of clusters in the modeling based on feature space partitioning. Finally, we intend to evaluate the performance of classifiers other than SVM as base classifiers.

6. Acknowledgments

The research which lead to these results has received funding from the European Communitys FP7 under grant agreement number 261743 (NoE VideoSense).

7. References

[1] B. Bushman, L. Huesmann, Short-term and long-term effects of violent media on aggression in children and adults, Archives of Pediatrics & Adolescent Medicine 160 (4) (2006) 348.

[2] C.-H. Demarty, B. Ionescu, Y.-G. Jiang, V. L. Quang, M. Schedl, C. Penet, Benchmarking violent scenes detection in movies, in: IEEE International Workshop on Content-Based Multimedia Indexing (CBMI), 2014, pp. 1–6.

[3] M. Sjöberg, B. Ionescu, Y.-G. Jiang, V. L. Quang, M. Schedl, C.-H. Demarty, The MediaEval 2014 Affect Task: Violent Scenes Detection, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[4] L.-H. Chen, H.-W. Hsu, L.-Y. Wang, C.-W. Su, Horror video scene recognition via multiple-instance learning, in: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 1325–1328.

[5] T. Giannakopoulos, A. Makris, D. Kosmopoulos, S. Perantonis, S. Theodoridis, Audio-visual fusion for detecting violent scenes in videos, Artificial Intelligence: Theories, Models and Applications (2010) 91–100.

[6] Z.-H. Zhou, Ensemble methods: foundations and algorithms, CRC Press, 2012.

[7] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, S. Theodoridis, Violence content classification using audio features, Advances in Artificial Intelligence (2006) 502–507.

[8] F. D. de Souza, G. C. Chávez, E. A. do Valle Jr., A. de A. Araújo, Violence detection in video using spatio-temporal features, in: SIBGRAPI Conference on Graphics, Patterns and Images, 2010, pp. 224–230.

[9] T. Hassner, Y. Itcher, O. Kliper-Gross, Violent flows: Real-time detection of violent crowd behavior, in: IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, pp. 1–6.

[10] Y. Gong, W. Wang, S. Jiang, Q. Huang, W. Gao, Detecting violent scenes in movies by auditory and visual cues, Advances in Multimedia Information Processing-PCM (2008) 317–326.

[11] R. Yan, M. Naphade, Semi-supervised cross feature learning for semantic concept detection in videos, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2005, pp. 657–663.

[12] L.-H. Chen, H.-W. Hsu, L.-Y. Wang, C.-W. Su, Violence detection in movies, in: International Conference on Computer Graphics, Imaging and Visualization (CGIV), 2011, pp. 119–124.

[13] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, Advances in Neural Information Processing Systems 15 (2002) 561–568.

[14] S. Goto, T. Aoki, Violent scenes detection using mid-level violence clustering, in: International Conference on Computer Science & Information Technology (CCSIT), 2014.

[15] Y. Yan, Y. Yang, D. Meng, G. Liu, W. Tong, A. G. Hauptmann, N. Sebe, Event oriented dictionary learning for complex event detection, IEEE Trans. Image Processing 24 (6) (2015) 1867–1878.

[16] Y. Yan, E. Ricci, R. Subramanian, G. Liu, N. Sebe, Multitask linear discriminant analysis for view invariant action recognition, IEEE Trans. Image Processing 23 (12) (2014) 5599–5611.

[17] Y. Yan, E. Ricci, R. Subramanian, G. Liu, O. Lanz, , N. Sebe, A multi-task learning framework for head pose estimation under target motion, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).

[18] N. Derbas, G. Quénot, Joint audio-visual words for violent scenes detection in movies, in: ACM International Conference on Multimedia Retrieval (ICMR), 2014, pp. 483–486.

[19] B. Ionescu, J. Schlüter, I. Mironica, M. Schedl, A naive mid-level concept-based fusion approach to violence detection in hollywood movies, in: ACM International Conference on Multimedia Retrieval (ICMR), ACM, 2013, pp. 215–222.

[20] C. Penet, C.-H. Demarty, G. Gravier, P. Gros, Multimodal information fusion and temporal integration for violence detection in movies, in: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 2393–2396.

[21] C.-H. Demarty, C. Penet, G. Gravier, M. Soleymani, The MediaEval 2012 Affect Task: Violent Scenes Detection, in: MediaEval Workshop, 2012.

[22] J. Lin, W. Wang, Weakly-supervised violence detection in movies with audio and video based co-training, Advances in Multimedia Information Processing-PCM (2009) 930–935.

[23] X. Ding, B. Li, W. Hu, W. Xiong, Z. Wang, Horror video scene recognition based on multi-view multi-instance learning, in: Computer Vision–ACCV 2012, Springer, 2013, pp. 599–610.

[24] C.-H. Demarty, C. Penet, B. Ionescu, G. Gravier, M. Soleymani, Multimodal violence detection in hollywood movies: State-of-the-art and benchmarking, in: Fusion in Computer Vision, Springer, 2014, pp. 185–208.

[25] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3169–3176.

[26] Q. Dai, Z. Wu, Y.-G. Jiang, X. Xue, J. Tang, Fudan-njust at mediaeval 2014: Violent scenes detection using deep neural networks, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[27] S. Avila, D. Moreira, M. Perez, D. Moraes, I. Cota, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, Recod at mediaeval 2014: Violent scenes detection task, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[28] V. Lam, D.-D. Le, S. Phan, S. Satoh, D. A. Duong, Nii-uit at mediaeval 2014 violent scenes detection affect task, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[29] H. L. Wang, L.-F. Cheong, Affective understanding in film, IEEE Transactions on Circuits and Systems for Video Technology 16 (6) (2006) 689–704.

[30] E. Acar, F. Hopfgartner, S. Albayrak, Detecting violent content in Hollywood movies by mid-level audio representations, in: IEEE International Workshop on Content-Based Multimedia Indexing (CBMI), 2013, pp. 73–78.

[31] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, The Journal of Machine Learning Research 11 (2010) 19–60.

[32] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, The Annals of statistics 32 (2) (2004) 407–499.

[33] M. Muja, D. G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP) 2.

[34] S. P. Lloyd, Least squares quantization in pcm, IEEE Transactions on Information Theory 28 (2) (1982) 129–137.

[35] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, in: ACM International Conference on Multimedia (ACMMM), 2010, pp. 1469–1472.

[36] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 1027–1035.

[37] A. Rahman, B. Verma, Novel layered clustering-based approach for generating ensemble of classifiers, IEEE Transactions on Neural Networks 22 (5) (2011) 781–792.

[38] H. He, E. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[39] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: Machine Learning: ECML 2004, Springer, 2004, pp. 39–50.

[40] T.-F. Wu, C.-J. Lin, R. C. Weng, Probability estimates for multi-class classification by pairwise coupling, The Journal of Machine Learning Research 5 (2004) 975–1005.

[41] M. Sjöberg, Y. Baveye, H. Wang, V. L. Quang, B. Ionescu, E. Dellandréa, M. Schedl, C.-H. Demarty, L. Chen, The MediaEval 2015 Affective Impact of Movies Task, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[42] MediaEval Violent Scenes Detection (VSD) Task Proceedings, 2014, http://ceur-ws.org/Vol-1263/.

[43] M. Sjöberg, I. Mironica, M. Schedl, B. Ionescu, Far at mediaeval 2014 violent scenes detection: A concept-based fusion approach, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[44] B. Zhang, Y. Yi, H. Wang, J. Yu, Mic-tju at mediaeval violent scenes detection (vsd) 2014, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[45] D. Castán, M. Rodrıguez, A. Ortega, C. Orrite, E. Lleida, Vivolab and cvlab-mediaeval 2014: Violent scenes detection affect task, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[46] E. Acar, S. Albayrak, Tub-irml at mediaeval 2014 violent scenes detection task: Violence modeling through feature space partitioning, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[47] B. do Nascimento Teixeira, Mtm at mediaeval 2014 violence detection task, in: Working Notes Proceedings of the MediaEval Workshop, 2014.

[48] MediaEval Violent Scenes Detection (VSD) Task Proceedings, 2014, http://ceur-ws.org/Vol-1263/.

[49] J. Tu, Z. Wu, Q. Dai, Y. Jiang, X. Xue, Challenge huawei challenge: Fusing multimodal features with deep neural networks for mobile video annotation, in: IEEE International Conference on Multimedia and Expo (ICME) Workshops, 2014, pp. 1–6.

[50] MediaEval Affective Impact of Movies (including Violence) Task Proceedings, 2015, http://ceur-ws.org/Vol-1436/.

[51] Q. Dai, R.-W. Zhao, Z. Wu, X. Wang, Z. Gu, W. Wu, Y.-G. Jiang, Fudan-huawei at mediaeval 2015: Detecting violent scenes and affective impact in movies with deep learning, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[52] Y. Yi, H. Wang, B. Zhang, J. Yu, Mic-tju in mediaeval 2015 affective impact of movies task, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[53] V. Lam, S. Phan, D.-D. Le, S. Satoh, D. A. Duong, Nii-uit at mediaeval 2015 affective impact of movies task, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[54] Q. Jin, X. Li, H. Cao, Y. Huo, S. Liao, G. Yang, J. Xu, Rucmm at mediaeval 2015 affective impact of movies task: Fusion of audio and visual cues, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[55] G. Trigeorgis, E. Coutinho, F. Ringeval, E. Marchi, S. Zafeiriou, B. Schuller, The icl-tum-passau approach for the mediaeval 2015 affective impact of movies task, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[56] I. Mironica, B. Ionescu, M. Sjöberg, M. Schedl, M. Skowron, Rfa at mediaeval 2015 affective impact of movies task: A multimodal approach, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[57] P. Marin Vlastelica, S. Hayrapetyan, M. Tapaswi, R. Stiefelhagen, Kit at mediaeval 2015–evaluating visual cues for affective impact of movies task, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[58] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, Recod at mediaeval 2015: Affective impact of movies task, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[59] O. Seddati, E. Kulah, G. Pironkov, S. Dupont, S. Mahmoudi, T. Dutoit, Umons at mediaeval 2015 affective impact of movies task including violent scenes detection, in: Working Notes Proceedings of the MediaEval Workshop, 2015.

[60] R. Chakraborty, A. K. Maurya, M. Pandharipande, E. Hassan, H. Ghosh, S. K. Kopparapu, Tcs-ilab-mediaeval 2015: Affective impact of movies and violent scene detection, in: Working Notes Proceedings of the MediaEval Workshop, 2015.