Information Fusion 45 (2019) 307–323

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

Multimodal data fusion for sensitive scene localization

Daniel Moreira[a], Sandra Avila[a,b,*], Mauricio Perez[a], Daniel Moraes[a], Vanessa Testoni[c], Eduardo Valle[b], Siome Goldenstein[a], Anderson Rocha[*,a]

[a] Institute of Computing, University of Campinas, Brazil
[b] School of Electrical and Computing Engineering, University of Campinas, Brazil
[c] Samsung Research Institute Brazil, Brazil

ARTICLE INFO

Keywords:
Multimodal data fusion
Sensitive scene localization
Pornography localization
Violence localization

ABSTRACT

The very idea of hiring humans to avoid the indiscriminate spread of inappropriate sensitive content online (e.g., child pornography and violence) is daunting. The inherent data deluge and the tediousness of the task call for more adequate approaches, and set the stage for computer-aided methods. If running in the background, such methods could readily cut the stream flow at the very moment of inadequate content exhibition, being invaluable for protecting unwary spectators. Except for the particular case of violence detection, related work to sensitive video analysis has mostly focused on deciding whether or not a given stream is sensitive, leaving the localization task largely untapped. Identifying when a stream starts and ceases to display inappropriate content is key for live streams and video on demand. In this work, we propose a novel multimodal fusion approach to sensitive scene localization. The solution can be applied to diverse types of sensitive content, without the need for step modifications (general purpose). We leverage the multimodality data nature of videos (e.g., still frames, video space-time, audio stream, etc.) to effectively single out frames of interest. To validate the solution, we perform localization experiments on pornographic and violent video streams, two of the commonest types of sensitive content, and report quantitative and qualitative results. The results show, for instance, that the proposed method only misses about five minutes in every hour of streamed pornographic content. Finally, for the particular task of pornography localization, we also introduce the first frame-level annotated pornographic video dataset to date, which comprises 140 h of video, freely available for downloading.

1. Introduction

We define a *sensitive scene* as a motion picture excerpt whose content may inflict harm (e.g., trauma, shock, or fear) to particular audiences (e.g., children or unwary spectators), due to the inappropriateness of content. Typical representatives include – but are not limited to – scenes depicting pornography and violence (during working time, at school, or in the church, for instance), animal cruelty and child abuse (probably anytime, anywhere), hate speech (depending on the broadcast media), etc.

Due to the recent popularization of mobile amateur live video stream services, which present high pervasiveness and very unpredictable content, sensitive scenes depicting suicide [1,2], murder [3], murder attempt [4], torture [5], rape [6], sexual intercourse of underages [7] – only to name a few – have gone viral over the Internet and social networks. This is alarming as sensitive content may be harmful (e.g., violent media contribute to aggressive behavior and desensitization to brutality in children [8]) and even illegal (e.g., child pornography [9,10]).

In face of the need for moderating the online spread of sensitive scenes, the employment of human regulators for constantly analyzing such streams often leads to stress and trauma [11], justifying the search for computer-aided solutions, to alleviate the job of moderators.

The automatic detection of sensitive content is a challenging and still open problem, mainly due to the subjectivity and to the openness of the concepts that one might want to detect. For instance, depending on sociocultural aspects, nudity may either be a proxy for pornography or simply have artistic or educational purposes. Therefore, relying on skin detectors that operate on still frames of a video might be helpful, but not ultimate for detecting pornography. Complementing the detection with the analysis of the video sound (e.g., looking for moaning sounds) and of the video motion (e.g., looking for repetitive patterns) would improve the accuracy of the detector. In a similar fashion, body-part detectors might be useful for detecting physical violence, but useless for identifying verbal abuse. Sound recognition would thus play an important role. Pixel-color-based blood detectors could be used to gauge

[*] Corresponding authors.
E-mail addresses: daniel.moreira@ic.unicamp.br (D. Moreira), sandra@ic.unicamp.br (S. Avila), anderson.rocha@ic.unicamp.br (A. Rocha).

https://doi.org/10.1016/j.inffus.2018.03.001
Received 1 May 2017; Received in revised form 28 February 2018; Accepted 5 March 2018
Available online 07 March 2018

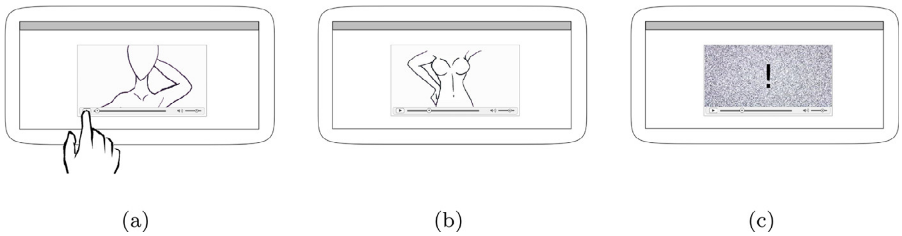


Fig. 1. Application example of sensitive content localization. In (a), the user starts to play a chosen video, within a tablet, through a safe video player. In (b), the video that is being played is about to show sensitive content (pornographic). In (c), the pornographic scenes are properly avoided.

the severity of the violence and complement the analysis.

In this vein, to deal with the subjectivity and openness of the target sensitive concept, it is important to have a way of relying on the combination of multimodal (and complementary) features one can extract from digital videos. Still frames, motion patterns, sound effects, soundtracks, and even subtitles and closed captions – if available – can be used to enhance the detection process.

In addition to the issues of concept subjectivity and openness, prior art of sensitive content analysis usually tackles the matter as a decision problem, seeking to define whether or not a given stream has *any* occurrence of a particular target sensitive concept (a.k.a., sensitive video *classification*). In turn, there exist some works – mostly related to the MediaEval violent scenes detection (VSD) task [12,13] – that take aim at the sensitive scene *localization* problem, i.e., the problem of *finding* the sensitive moments within a video timeline, with proper begin and end times.

Although useful for further web video crawling, the classification approach (decision problem) presents the drawback of having to process the entire video prior to labeling it as sensitive or non-sensitive. The localization approach (search problem), in turn, is more suitable for analyzing live streams and videos on demand. In spite of that, to the best of our knowledge, there are no solutions in the literature that localize sensitive content other than violence.

Fig. 1 depicts a possible application of a sensitive content locator. The action starts in (a), with a person (e.g., a child) playing a chosen video, through a safe video player, which was installed in a personal tablet. In (b), the video content is about to depict sensitive (pornographic) scenes, which are properly prevented in (c), when the pornographic scenes are avoided, according to a sensitive-scene localization process that works in the background.

This paper proposes a novel multimodal fusion pipeline for sensitive scene localization underpinned by the combination of different and independent sensitive snippet[1] classifiers. As each snippet classifier can rely on a particular video data modality (e.g., still frames, audio stream, video motion, etc.), the pipeline has an important multimodal capability. Moreover, the method leverages content of different time-overlapping snippets, to provide a dense sampling and a dense classification of the video timeline. The combination of classifiers is carried out by means of a late fusion of the sensitiveness classification scores that are returned by each snippet classifier. Scores that refer to the same video instant of interest are used to generate a single time-localized fusion feature vector. To create the fusion-vector configurations that better indicate sensitive and non-sensitive video moments, we employ machine-learning techniques. The present pipeline is of general purpose; it can be used – without step modifications – for the detection of diverse sensitive content types (e.g., pornography, violence, gore scenes, child abuse, etc.).

For validation, we perform experiments with both pornographic and violent content localization, two of the commonest types of inappropriate material, specially for their relevance and negative impact on minors [14,15]. For violence localization, we adopt the same infrastructure (dataset and evaluation protocol) provided by the MediaEval VSD task [12,13]. For the pornography localization task, in turn, we employ the Pornography-2k dataset [16] after properly annotating its 140 h of video footage to the frame level, manually. To the best of our knowledge, Pornography-2k becomes the first pornographic dataset in the literature that contains binary annotation (i.e., pornographic vs. non-pornographic) for every single frame. As another contribution of this work, we make the annotated dataset freely available to the scientific community, upon request and the sign of a proper responsibility agreement, due to its sensitive content.

For the sake of information, one additional issue that motivates the present work regards the fact that video watching and online live streaming are performed mostly on mobile handheld devices, practically anywhere, elevating the possibilities of inappropriate sensitive-content disclosure. Therefore, this research is part of a major effort[2] to design ubiquitous and efficient solutions, which can operate on the consumer side – even on devices with limited hardware (e.g., smartphones and tablets, with modest memories and processing power). Such aspect influences the strategies we are combining with the proposed fusion pipeline, in the experimental setup.

We organize the remainder of this paper into five sections. In Section 2, we explore related work to sensitive media analysis. In Section 3, we present the proposed pipeline to localize sensitive scenes through a multimodal fusion of digital video data. In Section 4, we explain the experimental setup while, in Section 5, we discuss the obtained results for both pornography and violence localization. Finally, in Section 6, we conclude the paper and elaborate on possible future work.

2. Related work

The literature of multimedia retrieval reports the importance of combining features from different modalities (e.g., video, audio, and text) to design more effective concept-based video-querying systems [17–19]. As discussed by Snoek et al. [17], fusion strategies can basically operate at either the feature level (a.k.a., early-fusion methods) or at the decision (classification) level (a.k.a., late-fusion methods). What level is better depends on the target concept one wants to retrieve. In a further work, in the occasion of surveying methods of multimedia retrieval, Snoek and Worring [18] tackled again the issue of multimodal information fusion. According to their observations, works that use early-fusion methods must deal with problems such as data synchronization, normalization, and transformation (e.g., feature concatenation), as the features come from different domains. Works relying upon late fusion methods, in turn, have mostly to deal with classification-score normalization, which is usually done by normalizing the values to a range between zero and one. The scores can then be combined in either an unsupervised manner (e.g., through simple averaging score, minimum score, maximum score, etc.), or in a supervised manner (e.g., through Support Vector Machines, SVMs [20], in a meta-recognition fashion). No fusion methodology (either early or late) seems to be consistently better than the other, though. For a more recent survey about data fusion and multimedia retrieval, please refer to [19];

[1] A snippet is any video excerpt.

[2] The present work is part of the project entitled “Sensitive Media Analysis”, sponsored by Samsung. The proposed method is patent pending under the application number US 15/198,626, filled on June 30, 2016.

Table 1
Sensitive video detectors from the literature that have used data fusion. We have not been able to find any pornography localization solutions.

	Reference	Dataset	Visual features	Auditory features	Fusion
Pornography Classification	Jansohn et al. [37]	In-house	DCT; MPEG motion vectors; skin	None	Late linear combination
	Ulges et al. [36]	In-house	DCT; MPEG motion vectors; skin	MFCC	Late linear combination
	Perez et al. [29][a]	Pornography-2k	Raw frames; optic flow; MPEG motion vectors	None	Early in-CNN; mid in-SVM; late linear combination
Violence Classification	Acar et al. [40]	MediaEval 2012	Motion vectors on frame blocks	MFCC	Late linear combination
	Derbas and Quénot [39]	MediaEval 2013	STIP	MFCC	Early feature concatenation
	Mironică et al. [38]	MediaEval 2013	HOG; color histogram	MFCC; rollof; etc.	Late linear combination
Violence Localization	Zhang et al. [42]	MediaEval 2014	SIFT; Dense Traject.	MFCC	Late linear combination
	Lam et al. [45][a]	MediaEval 2014	SIFT; Dense Traject.; raw frames	MFCC	Late linear combination
	Dai et al. [46][a]	MediaEval 2014	STIP; Dense Traject.; raw frames	MFCC	Late linear combination

DCT: discrete cosine transform, "in-CNN": input fed to CNN, "in-SVM": input fed to SVM.
[a] CNN-based.

strategies are not very different from the earlier ones reported in [18], though.

Although being certainly useful for inspiration purposes, fusion solutions of multimedia retrieval do not apply directly to the problem of sensitive content localization. The matter is related to the fact that both tasks (multimedia retrieval vs. content localization) are conceptually different. While multimedia retrieval is essentially a decision problem – i.e., the following question is posed: is concept ω present in shot i? [17]) – content localization is a search problem; rather than answering "yes" or "no", the designed solutions must identify the edges of the sensitive scenes.

In the same direction, related work in sensitive media analysis tackles the matter either as a decision problem (a.k.a., sensitive video classification) or as a search problem (a.k.a., sensitive scene localization). The former aims at deciding if a given stream has sensitive material while the latter aims at returning the sensitive scenes. Regardless of the approach, the typical pipeline for sensitive media analysis can have its operation framed in a three-layered representation. Within it, the (i) low-level layer refers to the video description, where the visual, auditory, and textual streams are directly accessed for the extraction of low-level features. For instance, concerning the visual stream (i.e., video frames), local descriptors such as Scale-Invariant Feature Transforms (SIFT) [21] and Histograms of Oriented Gradients (HOG) [22] can be used to describe perceptual features directly from the frame pixel values. In a similar fashion, the audio stream can be described through Mel-Frequency Cepstral Coefficients (MFCC) [23], and the video space-time can be described with Space-Time Interest Points (STIP) [24], Dense Trajectories [25], or Temporal Robust Features (TRoF) [16]. One level up, the (ii) mid-level layer targets the combination of the low-level features into global video representations, with intermediate complexity, as a way to reduce the semantic gap between the low-level features and the high-level target sensitive concept (e.g., pornography, violence, etc.). Solutions in this line of research vary from the construction of codebooks and Bags of Features (BoF) [26], to Vectors of Linearly Aggregated Descriptors (VLAD) [27] and Fisher Vectors [28], to Deep Learning methods [29]. On top of that, the (iii) high-level layer deals with the challenge of learning and predicting the classes of the global video representations. This is often accomplished by means of Support Vector Machines (SVM) [20] and Naïve Bayes Classifiers [30], among others.

As one might expect, works on sensitive video analysis that rely on the three-level pipeline are abundant, ranging from nudity classification [31], to pornography classification [32–37], to violence classification [38–41], and to violence localization [42]. Notwithstanding, more recently, some works on sensitive content detection have been replacing the first two levels, or even the entire pipeline, with Convolutional Neural Networks (CNN) [43]. That is the case of pornography classification [29,44], and of violence localization [45,46].

The literature of sensitive video analysis tackles mostly nudity, pornography, and violence detection. In the particular case of nudity and pornography, to the best of our knowledge, there is only research regarding video classification. Despite the relevance and utility, there is a lack of solutions and datasets for pornographic scene localization. In the particular case of violence, thanks to the MediaEval benchmark initiative, the scientific community can count on proper datasets, common groundtruth, and standard evaluation protocols for violence classification [47] and for violence localization [48] tasks.

In face of the related work, this research takes aim at sensitive scene localization for *both* pornography (for the first time, to our best knowledge) and violence concepts. Moreover, it aims at combining multimodal features (visual and auditory) that can be extracted from the video stream, with the intent to improve the performance of sensitive scene localization. With such strategy, we reached second place in the 2014 MediaEval VSD task competition, regarding the localization of violent scenes within webvideos (a.k.a., generalization task). Official results are reported in [49].

Table 1 puts together related work that have relied upon more than one feature for sensitive video analysis thus far. As one might observe, independently of the target problem (pornography classification, violence classification, or violence localization), the fusion strategies do not diverge too much from late linear combination (i.e., a late weighted sum of the classification scores), except for the works of Perez et al. [29] and of Derbas and Quénot [39].

Perez et al. [29] tackled the problem of pornography classification, in which the system must decide if a given footage has any occurrence of pornographic content. For that, they proposed feeding pre-trained and fine-tuned CNN with samples of three data types: (i) either raw frames, or (ii) vertical and horizontal components of optical flow vectors, or (iii) vertical and horizontal components of MPEG motion vectors. In some situations, the authors suggested to add the three types into a multichannel image, before feeding it to a CNN; in such cases, the fusion was considered an early one. In other situations, the authors proposed having one particular CNN for each one of the three data types. For obtaining what they called a mid-level fusion, they proposed concatenating the outputs of the last hidden layers of each CNN, prior to using it to train a single SVM. For obtaining a late-fusion alternative, in the other hand, they suggested training one SVM for each CNN, based on the respective outputs of the last hidden layers. In this case, a linear combination should be used for combining the SVM classification scores. According to their experiments, the late fusion strategy delivered better results.

Derbas and Quénot [39], in turn, aimed at violence classification, in which the system must decide if a given footage is violent or not. They proposed the use of Histograms of Optical Flow (HOF) [50] for describing STIP-detected space-temporal interest points, and MFCC for describing the audio stream. The most evident particularity of their approach relied on the early fusion of the low-level features, which were concatenated according to a randomly selected subset of all possible combinations, within a given video shot. By interpreting such concatenations as joint audio-visual features, the authors constructed codebooks with them, and established bags of audio-visual features, per shot, which were fed to SVM classifiers.

Having the task of sensitive scene localization in mind, the present work is directly comparable only to the last set of publications (violence localization) listed in the bottom of Table 1. Roughly speaking, Zhang et al. [42] inherited the violence classification idea of segmenting the target streams into shots. For that, they employed a third-party shot boundary detection method. In the mid-level, for each type of feature (e.g., SIFT on regular grids, Dense Trajectories, and MFCC), they represented each shot by a proper Fisher Vector (equivalent to the notion of a bag). In the high-level, each set of feature-related Fisher Vectors was fed to a particular SVM classifier (i.e., they trained one SVM per feature type). Then, a weighted sum of classification scores was used for the final shot classification (late linear combination). Given that the labeled shots did not present time overlaps, Zhang et al. simplified the fusion of discrete bag scores. Their system just returned a time-sorted concatenation of the shot violence scores, when in test execution.

Contrary to Zhang et al. [42], Lam et al. [45] opted for dividing the streams into non-overlapping five-second snippets. In the mid-level, for each type of feature (e.g., SIFT on regular grids, Dense Trajectories, and MFCC), each snippet was encoded as a Fisher vector, and as a bag of features. Besides that, the authors fed keyframes to a CNN, for obtaining a third alternative of mid-level representation (a further improvement on their original task attendance [51]). In face of plenty of mid-level representations (Fisher vectors, BOF, and CNN outputs), one SVM classifier was trained for each feature type. To combine everything, a late linear combination of classification scores was performed, for the final snippet classification. In the end, in the online snippet score fusion, Lam et al. [45] proceeded as Zhang et al. [42], configuring their solution to return a concatenation of the adjacent snippet violence scores.

Dai et al. [46], in turn, divided the target streams into non-overlapping fixed-length three-second snippets. In the mid-level, for some features (e.g., Dense Trajectories), they represented each snippet by a Fisher Vector. For other features (e.g., STIP and MFCC), they established conventional BOF, one for each snippet. In face of such diversity of representations, they trained one SVM classifier for each feature type. Additionally, they fed some of the features to a CNN, that worked as a high-level classifier, equivalent to SVM. Once more, a late linear combination of classification scores was performed, for the final snippet classification. In contrast to the previous solutions, Dai et al. suggested a more complex strategy for the online bag score fusion. Snippet classification scores were first smoothed by a proper function. Then, each snippet received a label (violent or non-violent), according to a threshold on the smoothed scores. In the end, adjacent snippets with the same label were merged into a single segment, whose final violence score was set as the average of the merged scores. More recently, Dai et al. extended the previous work in [46] to [52], by employing recurrent neural networks. Results are not comparable, though, since the used datasets and protocols are not the same.

Contrary to these works, this paper introduces a novel meta-learning late fusion solution that is of general purpose: it can be used for the analysis of varied sensitive contents (either pornography or violence or other tasks), and diverse video types (either amateur or professionally edited), as we show through experiments. In addition, it allows the combination of time-overlapping snippets, as an effort to densely sample and classify the video content. To our best knowledge, no work has tried that before.

3. Proposed solution

In a typical sensitive video classification problem, the solutions are supposed to attribute a label to an entire well-defined *video unit* (for instance, a label for an entire video *shot*, or a label for an entire video *file*). That makes the application of BoF-based approaches straightforward: just establish a bag per video unit of interest, for a further label-prediction learning (while training), or for a further discrete classification (while testing).

However, for the sensitive scene localization problem, in which the solutions are supposed to point out when a stream starts and ceases to display inappropriate content, there is no clear definition of a video unit of interest to be labeled. In face of such absence, how could one still benefit from the use of methods such as bags of features for description, for instance? Given the many possibilities of video segmentation (e.g., frames, shots, scenes, etc.), it is not clear in which unit one should pool mid-level features to provide bag labels that are more supportive of the task of content localization.

As we are looking for designing a more general-purpose solution, we do not assume anything about the target video stream, regarding number of camera sources, presence of scene cuts, amateurishness, or studio film grammar. We tackle the video segmentation problem by pooling and normalizing consecutive features, as long as they belong to a same fixed-length video segment (a.k.a., a *snippet*). The inherent idea is to primarily classify such snippets; the resulting classification scores are further combined through the fusion method we are proposing.

As a consequence of the decision of using video snippets, we can admit that we have available various *sensitive snippet classifiers*. Each snippet classifier can rely on a particular data modality (e.g., video frames, audio stream, video motion, etc.). In addition, each one can be seen as an expert in predicting the sensitiveness of Δt-second-sized snippets. The value of Δt may vary from a single frame to the entire footage, depending on the type of sensitive content (e.g., either pornography, or violence, or hate speech, etc.), and on the type of analyzed media (e.g., either video motion, or audio, or still video frames, etc.). For instance, previous experience has shown us that five seconds is enough for capturing either violence- or pornography-related events (e.g., punches, kicks, slaps, kisses); such value might not be true for localizing other contents, though. Moreover, the literature has consistently verified that temporal information is fundamental for detecting sensitive contents [16,53]; thus, having snippets with the size of a single frame might not allow the proper capture of motion. Anyhow, one important aspect of the proposed fusion method is that it can deal with any type of snippet classifiers, considering both data modality (i.e., the solution is multimodal) and length of snippets.

Lastly, we establish snippets that systematically overlap in time, as an effort to let the recorded sensitive events be entirely enclosed by at least one bag, in spite of eventually being split among the others. Fig. 2 illustrates the advantage of segmenting videos into time-overlapping snippets. Images 1–6 depict the frames of a sample video sequence, whose frames 2–4 capture a sensitive event (actually a violent event, regarding a slap on the face). The labeled rectangles that are positioned below represent possible video snippets, which one might use for establishing bags of features. In the case of a non-redundant-content segmentation strategy, such as the ones used in [42,45,46], snippets might at most be extracted consecutively, as illustrated through the horizontally aligned white rectangles. As one might observe, due to the non-overlapping nature of the snippets, the violent motion is split and, therefore, entirely represented by none of the two possible bags. The gray rectangles, in turn, represent a segmentation provided by a time-overlapping strategy. The additional *Snippets C* and *D* increase the chances of the sensitive event be entirely analyzed.

Fig. 3 depicts a flowchart overview of the proposed method for sensitive scene localization. Each rounded rectangular box denotes an activity while the solid arrows represent the precedence of activities. Dashed arrows denote data flow. Ultimately, we aggregate the snippet classifications through late fusion.

As pointed out by Atrey et al. [54], late fusion strategies have the advantage of offering easier scalability, regarding the addition or subtraction of classifiers, when compared to early fusion solutions. Besides that, early fusion strategies present the drawback of having to combine low-level features from different modalities (e.g., visual and auditory), which certainly present distinct types of representation (for instance, in

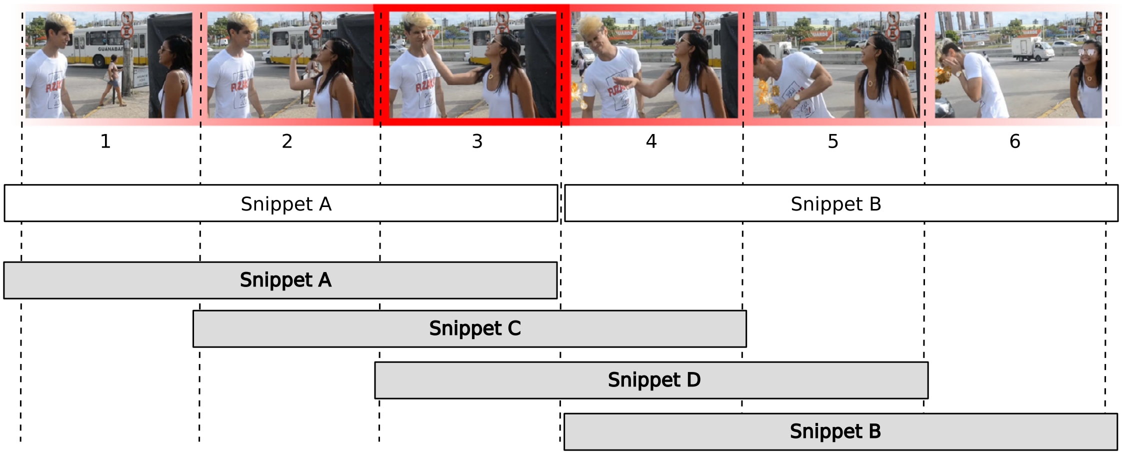


Fig. 2. Video snippet segmentation. Images 1–6 depict frames of a video sequence of interest. (Available at http://www.youtube.com/watch?v=S4PlfblnIws under Creative Commons license.) Frames 2–4 depict a violent event (slap on the face). Labeled rectangles represent possible snippets. White rectangles illustrate the segmentation provided by a non-redundant-content strategy. Due to its non-overlapping nature, the violent event is improperly split and spread between the two consecutive *Snippets A* and *B*. Gray rectangles, in turn, refer to the segmentation provided by a time-overlapping strategy. In such case, in spite of *Snippets A* and *B* still be splitting the violent event, the overlapping *Snippets C* and *D* increase the chance of the sensitive event be entirely described.

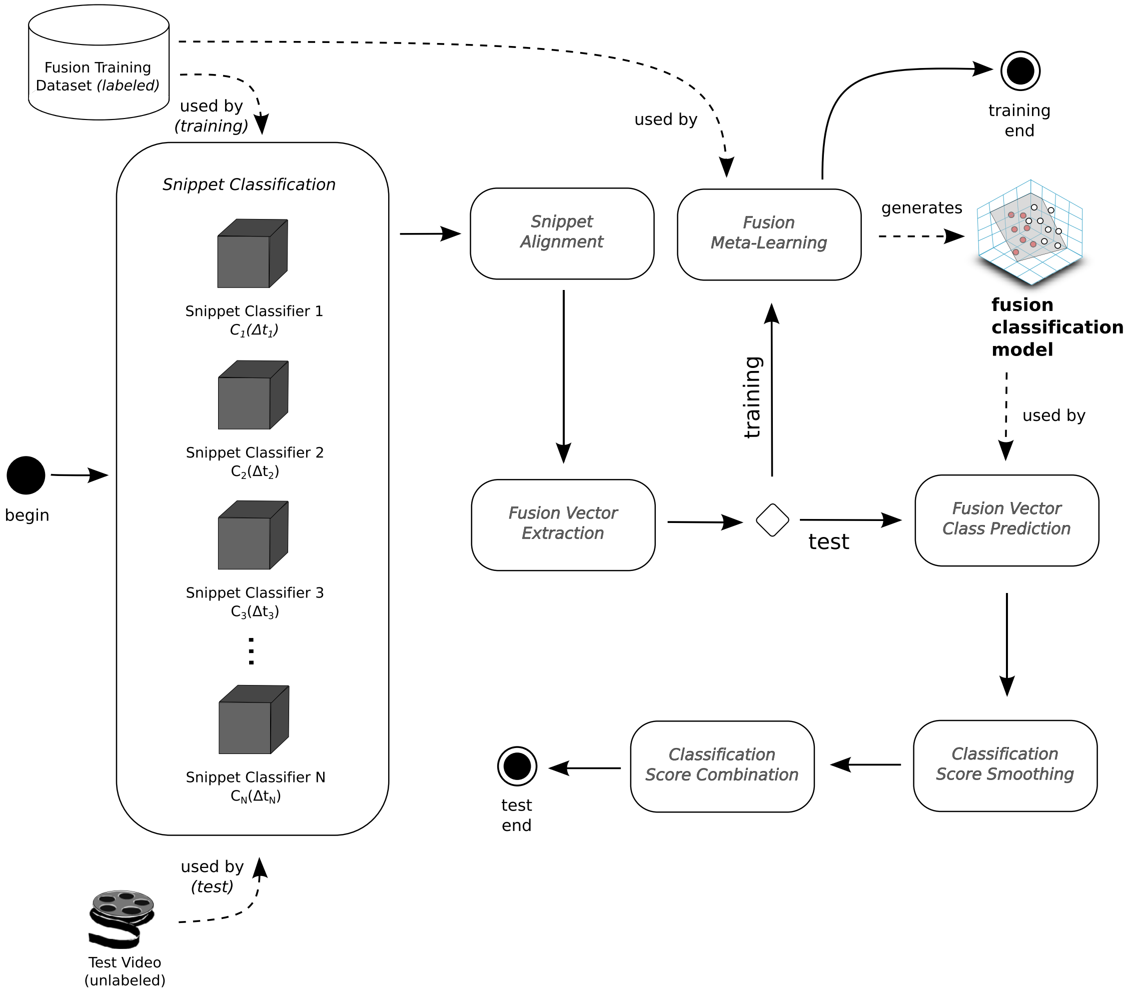


Fig. 3. Sensitive scene localization method overview. Rounded rectangular boxes denote activities, solid arrows represent their precedence, and dashed arrows denote data flow. Depending on the operation mode (training or test), the activity sequence may reach either the *training* or the *test end*. The *Snippet Classification* activity is detailed, to depict the use of *N* different *Snippet Classifiers*, as initial resources, which are properly represented as black boxes. Each snippet classifier $C_i(\Delta t_i)$, with $i \in [1, \ldots, N]$, is an expert in predicting the sensitiveness of Δt_i-second-sized snippets.

terms of dimension, scale, data type, etc.). In opposition, late-fusion solutions combine decisions at the semantic level, hence dealing with the same type of representation (e.g., classification or confidence scores, distances to decision hyperplanes, probabilities, etc.). Due to the data manipulation in more akin domains, late-fusion alternatives are usually more straightforward to be performed.

More specifically, we propose a machine-learning solution that aims at finding the best strategies for ultimately combining the outputs of N snippet classifiers (i.e., we propose a meta-learning strategy). Again, each snippet classifier $C_i(\Delta t_i)$, with $i \in [1.N]$, is an expert in predicting the sensitiveness of Δt_i-second-sized snippets. The sensitiveness, in turn, can be given through confidence scores, or distances to decision hyperplanes, or integer labels (e.g., + 1 for *sensitive*, − 1 for *non-sensitive*), etc., depending on the system settings. From now on, we will simply refer to such values as snippet *classification scores*.

As expected from most of the machine-learning techniques, the resulting fusing system may operate in one of two modes, namely training and testing (see Fig. 3). Depending on the type of system operation, the activity sequence may reach either the *training end*, or the *test end*. In Section 3.1, we detail the training activity sequence (*Snippet Classification, Snippet Alignment, Fusion Vector Extraction*, and *Fusion Meta-Learning*), in which the desired system content-localization behavior is learned from the labeled *Fusion Training Dataset*. In Section 3.2, in turn, we explain the test activity sequence (*Snippet Classification, Snippet Alignment, Fusion Vector Extraction, Fusion Vector Class Prediction, Classification Score Smoothing*, and *Classification Score Combination*), in which an arbitrary unlabeled *Test Video* is presented to and analyzed by the system.

3.1. Training activity sequence

Fig. 4 depicts the training activity sequence of the proposed fusion solution, by means of an illustrative toy case, with $N = 2$ snippet classifiers, and a *Fusion Training Dataset* that contains three videos (*Videos A, B,* and *C*, in the related diagram). Nevertheless, in spite of the quantity of snippet classifiers and of training videos, the aimed operation is always divided into four steps.

3.1.1. Snippet classification

Step 1 refers to the *Snippet Classification* activity, in which the *Fusion Training Dataset* – represented by a hollow cylinder – is submitted to the snippet classifiers. The training dataset must be annotated at frame level, with the indications of the start and end times of the sensitive and non-sensitive sequences. The snippet classifiers, in turn, are represented by *black boxes*, in the sense that it is not important how they operate, considering the execution of the proposed fusion method. In fact, what they really need to do is to return a set of classified snippets, which are grouped per classifier (and thus per length Δt_i), and per training video. In the chosen notation, we represent each snippet by a hollow rectangle, containing the resulting classification score in the center, and a small chronometer on the lower right corner, to highlight their temporal nature. The widths of these rectangles are supposed to indicate their duration, which means that – for the sake of illustration – *Snippet Classifier 1* ($C_1(\Delta t_1)$) is able to classify snippets that are twice as long as the snippets analyzed by *Snippet Classifier 2* ($C_2(\Delta t_2)$).

3.1.2. Snippet alignment

Step 2 refers to the *Snippet Alignment* activity, which is performed per training video: at such point, snippets coming from different streams are not ready to be combined yet. As one might observe in Fig. 4, as the snippets are defined by a start and an end time, it is possible to align them along the video timeline, in order to reveal their coincidences.

The *Snippet Alignment* activity works as follows. For each classifier, the respective snippets are sorted according to their start times, leading to one sorted list of snippets per classifier. These lists shall be used later on by a query function $q(t)$, which retrieves all the snippets, within all the lists, that coincide at a given instant of interest t. This query is accomplished by means of a binary search over each sorted list, which compares the instant of interest, and the bounds (start and end times) of the snippets.

3.1.3. Fusion vector extraction

Step 3, in turn, refers to the *Fusion Vector Extraction* activity. At this point, we want to generate a finite number of fusion vectors, which tie together the classification scores that were previously returned by the various snippet classifiers. For that, we sample the snippet alignments at every d seconds of video. Each second leads to an instant of interest t, which is fed to the aforementioned query function $q(t)$, and retrieves all the snippets that coincide at t.

Fig. 5 depicts the combination of fusion vectors, within the *Fusion Vector Extraction* activity, for a particular case of combining four snippet classifiers. As one might observe, for each video instant of interest (which is obtained in accordance to d), a fusion vector is extracted, containing the classification scores of coincident snippets. The coincident snippets must be sorted by source classifier and start time, according to a predefined order of snippet classifiers. As a matter of fact, any order is acceptable, as long as it is repeated in the test system operation. In Fig. 5, the colors of the fusion vector components indicate the snippet classifiers they are linked to, and therefore they reveal the fusion order.

The length l of every fusion vector is given by

$$l = \sum_{i=1}^{N} \left\lceil \frac{\Delta t_i}{s_i} \right\rceil, \tag{1}$$

where N is the number of fused snippet classifiers, Δt_i is the length, in seconds, of the snippets for which classifier C_i is an expert in predicting, and s_i is the step, in seconds, used to start a new snippet that is supposed to be analyzed by classifier C_i. For the sake of illustration, Eq. (2) calculates the length of the fusion vectors that are depicted in Fig. 5, where $N = 4$:

$$\begin{aligned} l_{sample} &= \left\lceil \frac{\Delta t_1}{s_1} \right\rceil + \left\lceil \frac{\Delta t_2}{s_2} \right\rceil + \left\lceil \frac{\Delta t_3}{s_3} \right\rceil + \left\lceil \frac{\Delta t_4}{s_4} \right\rceil \therefore \\ &= \left\lceil \frac{5}{2} \right\rceil + \left\lceil \frac{3}{2} \right\rceil + \left\lceil \frac{10}{7} \right\rceil + \left\lceil \frac{5}{5} \right\rceil = 8. \end{aligned} \tag{2}$$

On the occasion of creating the fusion vectors, in the case of missing snippets (and thus missing classification scores), the respective vector components may be assumed as a value of complete uncertainty (e.g., 0.5, in the case of a normalized confidence score, which varies from zero — i.e., no confidence at all — to one — i.e., total confidence), or they can be interpolated. Missing vector components are represented by ϵ, in Fig. 5.

3.1.4. Fusion meta-learning

Back to Fig. 4, each discrete fusion vector obtained in *Step 3* is linked to an instant of interest, within the target video timeline. As one might observe, the labels of such vectors are deductible from the training dataset groundtruth, being either depicted in red, if the vector concerns a sensitive instant, or in white, if the vector lies within a non-sensitive segment. In the sequence, the *Fusion Meta-Learning* activity (*Step 4*) refers to the application of a machine-learning technique for generating a mathematical model that is able to predict the labels of unknown fusion vectors. As these fusion vectors are, by themselves, generated from previously machine-learned classification scores, we may say this is a meta-learning stage of the joint behavior of such scores.

For this particular task, we exploit three implementation alternatives for the *Fusion Meta-Learning* activity: (i) score thresholding, as a baseline, (ii) Naïve Bayes Classifier [30], as a representative of generative strategies, and (iii) SVM [20], as a representative of

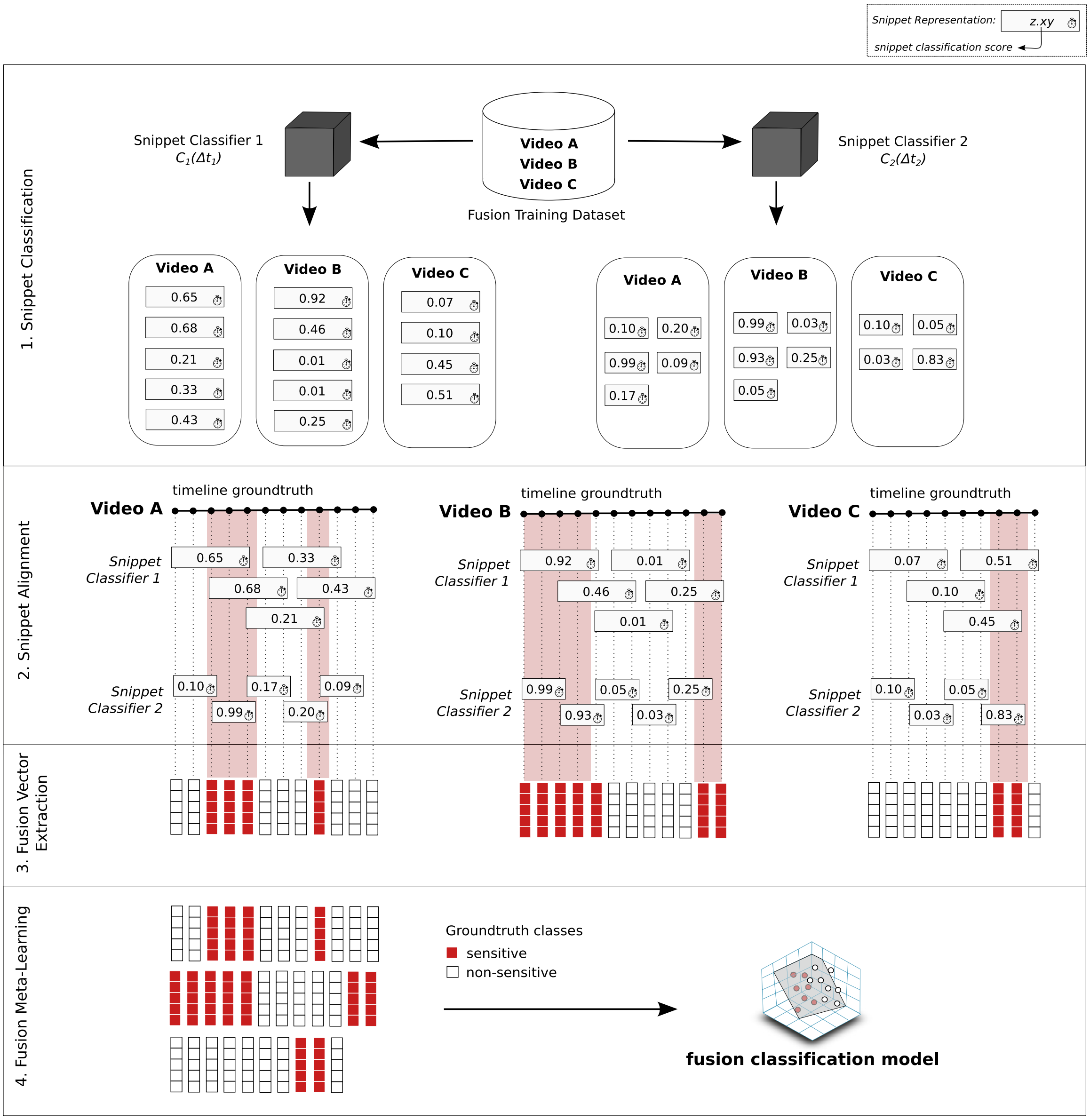

Snippet Representation: z.xy
snippet classification score
1. Snippet Classification
Snippet Classifier 1
C1(Δt1)
Video A
Video B
Video C
Fusion Training Dataset
Snippet Classifier 2
C2(Δt2)
Video A 0.65 0.68 0.21 0.33 0.43
Video B 0.92 0.46 0.01 0.01 0.25
Video C 0.07 0.10 0.45 0.51
Video A 0.10 0.20 0.99 0.09 0.17
Video B 0.99 0.03 0.93 0.25 0.05
Video C 0.10 0.05 0.03 0.83
2. Snippet Alignment
timeline groundtruth
Video A
Video B
Video C
Snippet Classifier 1
Snippet Classifier 2
3. Fusion Vector Extraction
4. Fusion Meta-Learning
Groundtruth classes
sensitive
non-sensitive
fusion classification model

Fig. 4. Toy case instantiation of the proposed fusion training pipeline. The method starts with the *Fusion Training Dataset* (*Videos A, B, and C*), which is submitted to the different snippet classifiers that need to be fused: classifiers $C_1(\Delta t_1)$ and $C_2(\Delta t_2)$. The training dataset sensitiveness must be annotated at frame level. The method ends with a meta-learned classification model (*fusion classification model*), which must be stored for further use, during the test system operation. The size of the training dataset, and the quantity of combined snippet classifiers, can be larger than the given example, with no changes on the order of the depicted steps. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

discriminative strategies. In addition, all of them are conceived to return a confidence score, in the real interval [0, ...,1], when classifying each fusion vector, which we refer to as *fusion score*. In the following, we detail each one of the adopted fusion meta-learning methods.

Score thresholding. Different from learning strategies, the score thresholding solution does not learn a mathematical model from the training dataset. In fact, one can admit that the model is known in advance, from the following and reasonable common sense: the ultimate label of the fusion vector is supposed to be the one that was detected with the highest confidence, over the coincidental snippet classifiers.

For that, we average the confidence scores that lie within each fusion vector component. Let $v[i]$ be the i-th snippet classification score, within a target fusion vector v whose length is l (i.e., $i \in [1, \ldots, l]$). The resulting fusion score of v is given by

$$fusionscore(v) = \frac{\sum_{i=1}^{l} v[i]}{l}, \tag{3}$$

where l (the size of fusion vectors) is given by Eq. (1). With such fusion score, we define the label of v as being

$$label(v) = \begin{cases} (+)\ positive, & \text{if } fusionscore(v) \geq \tau; \\ (-)\ negative, & \text{otherwise}, \end{cases} \tag{4}$$

where τ is the intended decision threshold.

Naïve Bayes classifier. As explained in [55], generative strategies for data learning usually establish a model of the joint probability of observations and labels, which are generalized by means of the Bayes theorem, for predicting the most likely label of an arbitrary unknown observation. In this work, we experiment with a simplified discrete naïve Bayes strategy [30].

We start with the binarization of the training fusion vectors, through

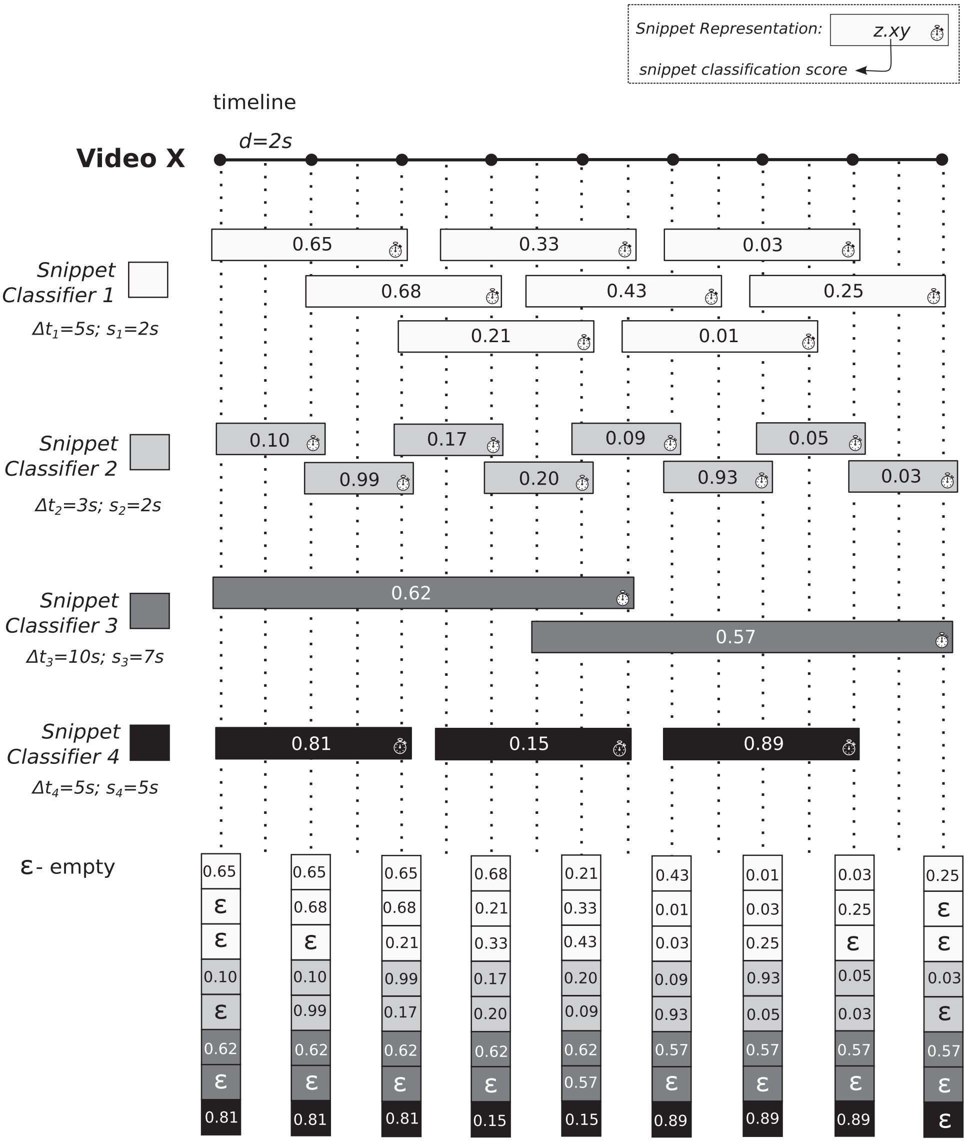


Fig. 5. Extracting the combined confidence vectors for later fusion. In this example, four snippet classifiers are being combined, regarding the content of *Video X*. Fusion vectors are extracted every *d* seconds of video, and are filled with snippet classification scores. Missing values are indicated by ϵ. The different vector component colors indicate the source snippet classifier.

the application of a threshold τ over each vector component. Let $v[i]$ be the i-th snippet classification score, within a target fusion vector v whose length is l (i.e., $i \in [1, \ldots, l]$). The binary value $b(v, i)$ that is respective to $v[i]$ is given by

$$b(v, i) = \begin{cases} 1, & \text{if } v[i] \geq \tau; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The score binarization reduces the fusion vector space to a finite number of 2^l possibilities, where l is the size of the fusion vectors. In face of such limited number of possible l-sized binarized fusion vectors (which are the observations), we adopt a frequentist approach to estimate the probabilities of each possible combination occur in the training set. In other words, we count, over the training dataset, how many positive and how many negative samples occur for each l-sized observation b_j, with $j \in [1, \ldots, 2^l]$, according to the training groundtruth. This procedure allows us to calculate the (i) prior probabilities $p(b_j)$ of all observations; (ii) the prior probability of finding a positive sample, $p(positive)$; and (iii) the conditional probabilities $p(b_j|positive)$ – i.e., the probability of an observation b_j be positive – only by relying upon the frequencies of the observations.

The mentioned prior and conditional probabilities (i, ii, and iii) constitute the Bayesian *fusion classification model* (please refer to Figs. 4 and 6). For predicting the probability of an arbitrary l-sized binarized vector b_j being positive, we apply the Bayes theorem

$$p(positive|b_j) = \frac{p(positive) \times p(b_j|positive)}{p(b_j)}, \tag{6}$$

where $j \in [1, \ldots, 2^l]$.

Complementarily, we calculate the probability of b_j being negative as $1.0 - p(positive|b_j)$. To determine the snippet's label (positive or negative), we pick the most probable one (i.e., positive, if $p(positive|b_j) \geq 0.5$, or negative, otherwise). The resulting fusion score is given by $p(positive|b_j)$.

Support Vector Machine. In contrast to the generative strategies, discriminative strategies focus on directly modeling the posterior probability of an observation belong to a target class [55]. Typical representatives include the solutions that aim at establishing the boundaries that better separate elements from different problem classes. The posterior probability, thus, can be estimated as a function

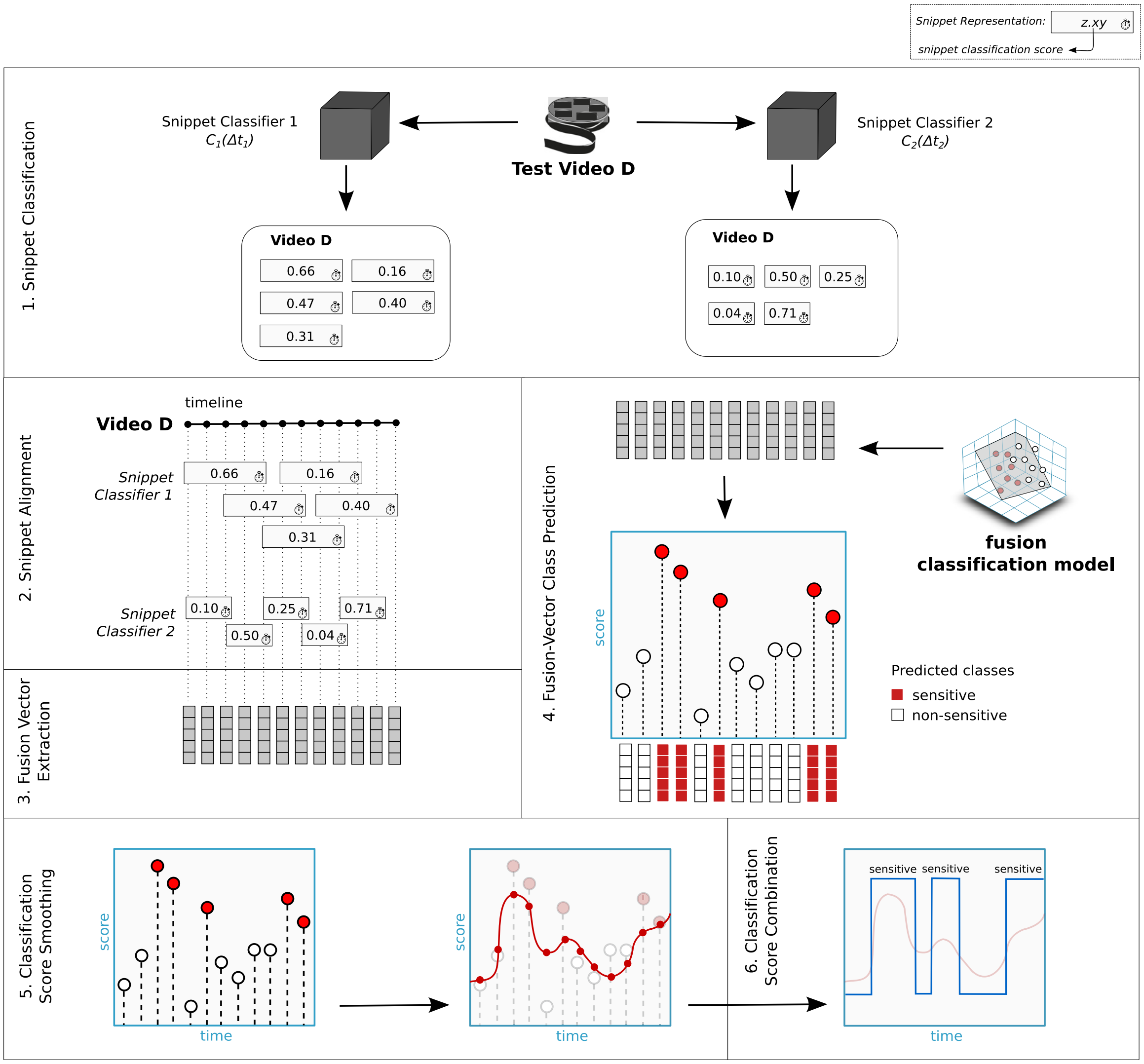


Fig. 6. Toy case instantiation of the proposed fusion test pipeline. The method starts with the unlabeled sample *Test Video D*, which is submitted to the different snippet classifiers that ultimately will be combined: classifiers $C_1(\Delta t_1)$ and $C_2(\Delta t_2)$. The method ends returning the instants when *Test Video D* starts and ceases to display sensitive content, optionally enriched by confidence scores. The quantity of combined snippet classifiers can be larger than the given example, with no changes on the order of the depicted steps. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

of the element distance to the boundary. The farther away an element is from the boundary within the side of class x, the stronger the evidence of belonging to x.

SVMs [20] are popular representatives of discriminative strategies. They comprise supervised-learning methods that compute the optimal hyperplane that better separates a feature space into two classes. In addition, it is possible to transform the original feature space into another, in which the computed separation hyperplane is more effective for class separation. This is done implicitly, by means of a kernel function, which algebraically operates over the elements of the original feature space, to find their representatives into the new better-separable higher-order feature space.

In this work, we apply an SVM with a Radial Basis Function (RBF) kernel, for learning how to separate the l-sized training fusion vectors v into positive and negative samples. With the learned hyperplane-separation model, we decide the label of arbitrary fusion vectors according to the class side (either positive or negative) they fall into. As pointed out in [56], RBF is a reasonable choice for SVM kernel, because it nonlinearly maps samples onto a higher dimensional space, so that, in the case of class elements being nonlinearly separable, the nonlinearity is handled. For reporting the fusion score (i.e., the SVM prediction confidence), we employ the standard Platt normalization [57], which calibrates the element distances to the decision hyperplane, conveniently returning a value in the real interval $[0, \ldots, 1]$.

3.2. Test activity sequence

Fig. 6 depicts the test activity sequence of the proposed fusion solution, by means of an illustrative toy case, with $N = 2$ snippet classifiers. In spite of the quantity of snippet classifiers, the aimed operation always starts with an arbitrary video (*Test Video D*), and it is always divided into six steps. The initial three activities (represented by *Snippet Classification, Snippet Alignment*, and *Fusion Vector Extraction*) are the same from the training sequence. The only difference relies on the absence of timeline groundtruths – in the test case – with no impact on the refereed activities. In the following, we detail the three remaining activities (*Fusion Vector Class Prediction, Classification Score Smoothing*, and *Classification Score Combination*), which are test-exclusive.

3.2.1. Fusion Vector Class Prediction

Prior to this step, the target video (properly represented by *Test Video D*, in Fig. 6) is supposed to have been segmented into snippets, which must have been classified during the *Snippet Classification* activity previously described. In addition, the classified snippets must have been aligned along the video timeline (during the *Snippet Alignment* activity), and thereafter combined into fusion vectors (during the *Fusion Vector Extraction* activity). In the particular case of the *Fusion Vector Extraction* activity, it is important to mention that the order in which the snippet classification scores are combined – for generating the fusion vectors – must be consistent with the one adopted in the training system operation (c.f., Fig. 5, for more details).

As one might observe, in the beginning of *Step 4*, in Fig. 6, the labels of the fusion vectors are unknown (what is represented by their gray colors). Hence, the system relies upon the *fusion classification model* to predict the labels of each fusion vector, with a proper confidence score. That justifies their red and white colors, in the end of *Step 4*. As a result, considering each fusion vector represents a discrete instant of interest within the target video timeline, the predicted labels actually classify the sensitiveness of every video instant of interest.

Strategies to perform the class prediction are a consequence of the chosen *Fusion Meta-Learning* solution (*Step 4*, in Fig. 4), which – as already mentioned – may comprise SVM [20], Naïve Bayes Classifiers [30], etc.

3.2.2. Classification Score Smoothing

Obtaining a classification confidence score for every video instant of interest may generate a noisy answer in time, with interleaving positive and negative segments at an unsound rate, which may change too much and too fast, regarding the actual occurrence of enduring and relevant sensitive events. Hence, in the *Classification Score Smoothing* activity, we can use a *denoising* function to smooth the classification scores along the video timeline.

We propose the use of a unidimensional Gaussian blurring function, with standard deviation σ, which is convolved with the time-sorted sequence of classification scores. This leads to a more well-behaved sequence of scores, besides offering the opportunity of eliminating eventually incorrect predictions, according to time-surrounding evidence. Dai et al. [46] report to adopt a similar solution, which relies upon a score-averaging convolution filter, instead of a Gaussian one.

3.2.3. Classification Score Combination

Finally, the *Classification Score Combination* aims at combining the discrete scores of adjacent video instants of interest that belong to the same sensitive class, according to decision thresholds. The inherent idea is to replace the sequences of diverse scores by a single, time-continuous, and representative one, which may persist for a longer time, better characterizing the sensitive and non-sensitive video moments. Strategies to accomplish this may comprise (but are not limited to) assuming a score threshold t, and then substituting all the time-adjacent scores equal to, or greater than t, by their average value (which is certainly not smaller than t). Complementarily, all the time-adjacent scores smaller than t shall be replaced by their average value, which, in turn, is certainly smaller than t. We ultimately end up with a continuous answer, which discriminates the instants the target video starts and ceases to disclose sensitive content.

In the next two sections, we detail the experimental setup and validate the proposed pipeline for both pornography and violence localization. For that, we rely upon visual and auditory features, which are obtained with diverse low-level video descriptors.

4. Experimental setup

In this section, we present the experimental setup for validating the proposed solution by detailing the used datasets, experimental protocols, and evaluation metrics, as well as the selected snippet classifiers, parametrization, and implementation details. Section 4.1 brings information regarding the datasets and the respective experimental protocols and evaluation metrics. In the sequence, Section 4.2 details the combined visual and auditory snippet classifiers, while Section 4.3 explains the investigated late multimodal fusion alternatives.

4.1. Datasets, experimental protocols, and evaluation metrics

As one might expect, we have different datasets for pornography and for violence localization. Depending on the dataset, we have a particular experimental protocol (with a particular data folding strategy, for instance), and a respective set of evaluation metrics. Therefore, Section 4.1.1 details the pornography localization task and its experimental particularities. Section 4.1.2, in turn, pays attention to the violence localization task and its nuances.

4.1.1. Pornography task setup

To the best of our knowledge, there is no video dataset in the literature with frame-level annotation for supporting the task of sensitive scene localization. To address this problem, we manually annotated every frame of the Pornography-2k dataset [16]. The Pornography-2k dataset has a total of 2000 videos, of which 1000 contain pornographic scenes (positive videos), and 1000 are free of pornographic content (negative videos). The samples were collected from the Internet and range from six seconds to 33 min. The content is very assorted, including both professional and amateur production. Pornographic samples depict several genres, varying from cartoon to live action, with diverse behavior and ethnicity.

The annotation process for the 1000 negative videos was straightforward: they were automatically and entirely marked as negative frame sequences as they do not contain any pornography grammar. To support the task of annotating the 1000 remaining positive videos, we developed a tool to extract every frame of a given video, and show the images in a time-sorted and keyboard-controlled way to a user. By inspecting the frames one-by-one, and pressing the correct keys, we were able to easily annotate parts of the streams as positive or negative.

For the annotation process, we recruited four of our authors (three men and one woman), which are in the range of 25 to 30 years old, all raised in the western culture. Each one was initially responsible for 250 videos, which were randomly distributed. All annotators adopted the concept of pornography as being “any explicit sexual matter with the purpose of eliciting arousal” [14] to equalize the situations one should consider positive. Aiming at calibrating the opinions, five videos were chosen at random and, prior to the actual annotation process, all four members dedicated some time to annotate these samples for further discussion. In spite of that, there were some videos whose annotation revealed itself as being slightly unclear (namely *medium cases*) or very dubious (namely *hard cases*), specially in the transitions from positive to negative scenes, and vice-versa. In such cases (around 9% concerning medium cases, and 3% concerning hard cases), the first annotation was further revised by the entire team together, leading to alterations whenever the group unanimously found it necessary.

Table 2 brings the statistics of the annotated videos. As one might observe, the Pornography-2k dataset has a total of almost 140 video hours out of which 91 h43 min (65.54%) refer to pornographic content.

We apply a $k \times 2$-fold cross-validation protocol [58] for data folding and validation, with $k = 3$, which we refer to as 3×2-fold protocol. In face of the 140-h Pornography-2k dataset and due to the extent of experiments carried out, this choice was the minimum number of repetitions for meaningful statistical tests later on. The 3×2-fold protocol consists of randomly splitting the dataset into two same-sized

Table 2
Time statistics on the annotated pornographic videos. As one might expect, negative videos do not have positive sequences, only negative. Positive videos, in turn, might have non-pornographic frame intervals.

	Non-porn scenes	Porn scenes	Total
Non-porn videos	40 h 25 min	00h 00 min	40h 25 min
Porn videos	07h 49 min	91h 43 min	99h 32 min
Total	48h 14 min	91h 43 min	139h 57 min

class-balanced folds, three times. Each time, training and test sets are switched, leading to six independent experiments, for each evaluated solution. In addition, to enable paired statistical tests, we submit the exact six folds to each pornography locator considered in this work. Therefore, whenever it is convenient to compare different locators with some statistical confidence, we employ the non-parametric pairwise Wilcoxon signed-rank test, with Bonferroni's *p*-correction [59].

Lastly, given the nature of our pipeline – in which we have two moments of data learning, (i) one related to the snippet classification learning, and (ii) another related to the fusion meta-learning – we further divide the training datasets into two disjoint parts: 60% for snippet classification learning, and 40% for fusion meta-learning. Using disjoint parts allows us to, during the fusion meta-learning stage, present a set of samples that was not previously used for training snippet classifiers. As a consequence, meta-learning classifiers are fed, during their training step, with more realistic outcomes from the snippet classifiers (since data is intentionally unknown). With the correct groundtruth, we can then give to the fusion meta-learning stage an opportunity to circumvent eventually mislabeled snippets, thus increasing the overall system robustness and generalizability.

For assessing the performance of the pornography locators, we report the *normalized classification accuracy* rate (ACC), and the *F_2 measure* (F_2). Prior to explaining ACC, we need to define *recall* and *specificity* from the point of view of pornography localization. Recall expresses the ability of a locator to identify truly pornographic video seconds as sensitive. For instance, if a given locator presents a recall of 75%, it is able to correctly recognize three in each four seconds of pornographic content. Specificity, in turn, measures the capacity of a locator to correctly identify truly negative video seconds as so. A specificity of only 50%, for example, means the system mislabels one in every two seconds of non-pornographic content, wrongly identifying it as sensitive. In this vein, ACC is the mean of recall and specificity. A higher accuracy indicates a higher capability of separating pornographic video seconds from non-pornographic ones.

F_2 measure, in turn, is a more complex metric that depends also on the concept of *precision*. From the point of view of pornography localization, precision expresses how many seconds are truly relevant (i.e., pornographic), among all the ones that a locator identifies as such. Therefore, F_2 is the weighted harmonic mean of recall and precision, which gives twice more weight to recall than to precision, by means of a $\beta = 2$ parameter. Eq. (7) depicts the original F_β formula

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall}, \tag{7}$$

in which we use $\beta = 2$. In doing so, F_2 lets us pay more attention to the recall of the solutions, rather than to their precision. This is useful because, in the case of pornography filtering, false-negative answers are worse than the false-positive ones. It is less prejudicial to wrongly deny the access to non-pornographic content, than to wrongly disclose pornographic content. Hence, we can consider that a solution with higher F_2 measure is better, because it cares more about how many pornographic video seconds are really being filtered out (recall), instead of how many "supposedly" positive seconds are indeed pornographic (precision).

4.1.2. Violence task setup

For the violence task, we adopt the same groundtruth and standard evaluation protocol provided by the MediaEval benchmark for conducting experiments. The MediaEval 2014 violent scenes detection (VSD) dataset [48] is an extension of the 2013 dataset [47] and comprises 31 Hollywood movie titles of diverse genres, from extremely violent (e.g., Pulp Fiction) to musical (e.g., The Wizard of Oz). Due to copyright issues, competitors and other interested people are supposed to purchase such titles at their own expenses. The MediaEval initiative provides only the annotations, which come separated into a training set, comprising 24 titles, and a test set with seven titles. In addition to the 31 titles, there is also a second minor dataset, which contains 86 *YouTube* clips that may last from six seconds to six minutes. In this particular case, these clips are provided within the dataset for free, since they are licensed under Creative Commons regulation.

With the intent of challenging participants to perform violent scene localization, the 2014 edition counts on frame-level annotations of all violent video segments, which are individually identified by their start and end frame numbers. These annotations had been carried out by several human assessors, in a hierarchical bottom-up revision manner [48]. For the annotators, violent segments were considered to be the ones a person would not let an eight-year-old child watch, due to physical violence [48]. In summary, nearly 12% of the training scenes contain violent content, while 17% of the test scenes are violent.

The VSD task motivation is the development of systems that may help users to choose suitable titles for their children, by retrieving the most violent movie parts for parental preview [47]. As a consequence, competitors' solutions are compared from the perspective of retrieval: the top-performing systems are the ones that return the largest number of violent scenes, at the first positions of the top-*k* retrieved scenes, properly ranked by violence confidence. Therefore, the MediaEval initiative suggests using the Mean Average Precision (MAP) metric for evaluation.

In the particular case of the 2014 edition, participants can provide any segmentation of the target video stream (in terms of segment sizes and positions), for attributing labels and confidence scores to each segment. As a consequence, competitors' segments may coincide only partially with the groundtruth scenes, and the tested systems may also provide various small segments that fit and intersects with an eventual larger scene from the groundtruth. To deal with these situations, MediaEval organizers propose a variation on the calculation of the hits (and thus of the precision), when measuring MAP. They only consider a segment prediction as a hit, if it overlaps with the corresponding groundtruth segment by more than 50%. In addition, to deal with the situation of evaluating many small segments, several hits on the same groundtruth scene only count as one true positive. The other hits are ignored, to avoid raising the value of MAP inappropriately. VSD organizers refer such variation of MAP calculation as MAP2014 [48].

Relying upon the MAP2014 metric, the MediaEval 2014 VSD task adopts a straightforward protocol. Participants must report results over the seven-title test dataset, which must not be used in any system training step. Solutions must contain a proper segmentation of the target stream, and each segment must receive a label (violent or non-violent), and a confidence violence score. The best solutions are the ones that report the highest values of MAP2014. For assessing the MAP2014, the MediaEval initiative provides a Perl script for free, which we use in our experiments.

Finally, given the nature of our approach – in which we have two moments of data learning, (i) one related to the snippet classification learning, and (ii) the other related to the fusion meta-learning – we separate the seven movies that had belonged to the 2013 test set [47], and 26 clips from the *YouTube* set, for exclusively use in the fusion meta-learning step. In a similar fashion to the pornographic setup, with such split, we aim at training the fusion meta-learning classifiers with more realistic outcomes from the snippet classifiers, which are formerly asked to label confidently unknown data.

4.2. Multimodal snippet classifiers

We evaluate the proposed fusion pipeline through different combinations of four distinct snippet classifiers. Two of these classifiers rely upon auditory features, namely MFCC [23] and prosodic features (fundamental frequency, voicing probability, and loudness contours). The remaining two rely upon visual features, of which one is representative of still image descriptors (namely, HOG [22]), and the other is representative of recently-proposed space-temporal descriptors (namely TRoF [53]).

MFCC features are used primarily for speech description [60], and a great deal of works in the literature have been using it for violent video content detection [38–40,42,45,46]. In this work, we use MFCC as the basis of the first available snippet classifier, through the OpenSmile library [60]. We therefore obtain 39-dimensional low-level auditory features in every 40 ms of video audio, without overlap.

In addition to MFCC, we extract prosodic features (PROS) to describe the audio and to support the second available snippet classifier. Similar to MFCC, we employ the OpenSmile library [60] to obtain three-dimensional features (fundamental frequency, voicing probability, and loudness contours) in every 40 ms of audio, without overlap.

To provide a visual descriptor that relies solely on static images, we employ HOG [22] as the basis of the third available snippet classifier. For the sake of processing time, we extract two frames per second from the video footage. HOG descriptions are then extracted on a dense spatial grid, at five scales, in the same manner as described in [53], leading to 128-dimensional low-level features.

Lastly, to capture video space-time properties, we employ TRoF [53] as the basis of the fourth available snippet classifier. The application of TRoF is made exactly in the same manner as described in [53], leading to 192-dimensional low-level descriptions.

As we have mentioned in Section 1, all these four chosen snippet classifiers regard solutions amenable to deployment on mobile devices: they present low-memory footprint and small processing time. Furthermore, all of them are trained to label snippets that are five-second long. In preliminary experiments with other data and not reported here, such duration showed a good tradeoff between the quantity of described snippets and the amount of information that constitutes each snippet. In the training phase of all classifiers, we consider a snippet negative if it falls entirely out of sensitive scenes. Similarly, we consider a snippet positive if it is at least 80% coincident with sensitive scenes. In the test phase, we describe one five-second long snippet in every second of a video sequence. That allows us to constitute one fusion vector per second, over the test dataset.

Regardless of the used low-level features, we employ Fisher Vectors [28] – one of the best mid-level representations [61] – to aggregate the low-level descriptions, within all the snippet classifiers. The codebooks are based on Gaussian Mixture Models (GMMs), each of which is estimated over one million randomly sampled low-level features (with 500,000 coming from the training sensitive scenes, and 500,000 coming from the training non-sensitive scenes). Moreover, each GMM contains 256 Gaussians, as suggested in [28].

Prior to the Fisher Vector encoding, we apply PCA over the low-level features, for either whitening or reducing their dimensionality, as suggested in [28]. MFCC descriptions are thus reduced to 24 dimensions (as recommended in [51]), while PROS features are whitened (i.e., we maintain their three dimensions), due to their original small size. HOG and TRoF descriptions, in turn, are reduced by half, as suggested in [53].

In the high level, we apply linear SVM classifiers, as suggested in [28], by means of the LIBLINEAR library [62]. We apply grid search to find the best c-SVM parameter, during the snippet classification training. Concerning the test phase, we obtain the confidence scores of each class prediction, which are normalized in the real interval $[0, \ldots, 1]$: the closer to one, the higher the certainty about the classification.

4.3. Late-fusion setup

As explained in Section 3, we investigate three meta-learning solutions for the late fusion of multimodal snippet classifiers: (i) score thresholding (THR), (ii) Naïve Bayes Classifier (NBC), as a representative of generative strategies, and (iii) SVM, as a representative of discriminative strategies. All of them are conceived to return a confidence score, in the real interval [0.1], when classifying each fusion vector, which we refer to as *fusion score*.

Regardless of the used fusion meta-learning method, in the test system operation, we always convolve a Gaussian window with standard deviation $\sigma = 5s$ (the size of each snippet) over the temporal sequence of obtained fusion scores, for smoothing (related to the *Classification Score Smoothing* task, which is explained in Section 3.2.2). In the end, the *Classification Score Combination* task (c.f., Section 3.2.3) takes place as previously described: by assuming a fusion score threshold $t = 0.5$, we substitute all the time-adjacent scores equal to, or greater than $t = 0.5$, by their average value. Complementarily, all the time-adjacent scores smaller than $t = 0.5$ are replaced by their own average value. In the case of eventually missing snippets – which are related to the ϵ value, in Fig. 5 – the empty fusion vector components are filled with a linear interpolation of the present ones.

In both THR and NBC solutions, we use threshold $\tau = 0.5$ (c.f., Eqs. (4) and (5)). Regarding the SVM solution, we employ the LIBSVM API [63] for training fusion vector classifiers, and for predicting the class of arbitrary fusion vectors. To find the parameters that lead to the best RBF kernel, we perform a grid-search with five-fold cross validation over the training dataset, as suggested in [56].

5. Experiments and validation

We present results on pornography localization in Section 5.1, while in Section 5.2, we report the ones for violence localization.

5.1. Pornography localization

Table 3 puts together all the results we have obtained for pornography localization over the Pornography-2k dataset. As explained in Section 4, we report values of normalized accuracy rate (ACC) and F_2 measure (F_2).

In Table 3(a), we present the individual results of the snippet classifiers, without combinations. As one might observe, visual features are more suitable for the task, with static and space-temporal approaches showing close performance. Indeed, a pairwise comparison of TRoF and HOG snippet classifiers shows that they are not statistically different

Table 3
Pornography localization result over the Pornography-2k dataset. We report the average performance over the 3 × 2 cross-validation folds. In all experiments, the standard deviation is lower than 0.04. In (a), results were obtained without fusion of snippet classifiers. In (b), results refer to the fusion of snippet classifiers that rely upon features of the same nature (auditory or visual). In (c), results refer to the multimodal fusion of snippet classifiers. The best results are highlighted in bold.

(a) No fusion				(b) Same-nature fusion				(c) Multimodal fusion			
		ACC (%)	F_2 (%)			ACC (%)	F_2 (%)			ACC (%)	F_2 (%)
Audio	PROS	76.31	77.22	THR	PROS + MFCC	82.79	84.21	THR	MFCC + TRoF	90.08	92.76
	MFCC	79.72	80.98		HOG + TRoF	90.74	**93.92**		ALL	**90.75**	93.53
Image	HOG	87.25	89.65	NBC	PROS + MFCC	81.33	82.56	NBC	MFCC + TRoF	89.52	91.61
	TRoF	86.47	89.89		HOG + TRoF	90.07	91.42		ALL	90.18	92.04
				SVM	PROS + MFCC	82.12	83.59	SVM	MFCC + TRoF	90.01	91.47
					HOG + TRoF	90.29	91.33		ALL	90.72	91.93

with respect to ACC. Besides that, PROS is the worst solution with 95% confidence, being statistically different even to MFCC, which presents the second worst results. Notwithstanding, if we take solely PROS into consideration, it is able to correctly classify three in every four seconds of video (ACC = 76.31%), starting with only three feature values in the low-level video description (due to the prosodic features). This shows a promising suitability for describing video in mobile devices, and for dealing with the tradeoff between efficiency and effectiveness.

In Table 3(b), we present the results of combining same-nature solutions (i.e., PROS with MFCC, for being auditory, and HOG with TRoF, for being visual). Regardless of the type of fusion meta-learning (THR, NBC, or SVM), the combined visual features once again outperform the combined auditory features, as expected. Indeed, the single visual solutions (HOG and TRoF) are better than any combination of auditory features (PROS + MFCC). For instance, TRoF is statistically better than THR-PROS + MFCC, NBC-PROS + MFCC, and SVM-PROS + MFCC, in terms of ACC, with 95% confidence. More importantly, however, the fusion of specific features always result in better values for ACC and F_2 measure, when compared to the isolated use of these same features. This hints at the expected complementarity of the features, even though, at this point, they are still of similar nature. For example, in the case of visual features (HOG and TRoF), the baseline THR fusion yields an error reduction – regarding ACC – of about 27%[3] and 31%,[4] when compared to the solely HOG- and TRoF-based solutions, respectively.

In Table 3(c), we present the results of combining snippet classifiers that rely upon features of different nature (e.g., auditory and visual, a.k.a., multimodal solutions). We evaluate the combination of the best auditory feature with the space-temporal one (MFCC + TRoF), and alternatively, we evaluate a complete fusion, with all the four available snippet classifiers (referred to as ALL, therefore combining PROS, MFCC, HOG and TRoF). In all cases, the multimodal combinations are not statistically different to the solutions that exclusively combine visual features (HOG + TRoF solutions). It indicates that the audio-based snippet classifiers do not produce hits on the occasions in which the visual classifiers miss, and vice-versa. Hence, they may not be complementary in the particular case of the Pornography-2k dataset. The possible reasons for the not so impressive performance of the audio-based snippet classifiers may rely on the samples of such dataset: many of them depict amateur content, with amateur editing. Hence, it is common to find sexual footage whose moaning sounds are further covered with electronic music, for not exposing ashamed spectators.

Finally, concerning the different types of fusion meta-learning (THR, NBC, or SVM), in the particular case of pornography, we notice that equivalent solutions (e.g., THR-HOG + TRoF, NBC-HOG + TRoF, and SVM-HOG + TRoF) are not statistically different, with respect to either ACC or F_2 measure.

Fig. 7 depicts the quality of pornography localization over a 1.5-min long video footage, which was sampled from the Pornography-2k dataset. As each row refers to the same footage, they individually represent the same timeline. Red and white areas depict the localization groundtruth: red for positive, and white for negative. As expected, these areas do not change along the boxes. Black dots, in turn, represent mislocalization: the lesser the quantity of black dots, the better the quality of a solution. Moreover, some video segments are labeled with capital letters (*A, B*, and *C*), for further reference.

In Fig. 7(a–d), we show the localization quality of each single solution, with no fusion of features. As one might observe, contrary to the general results, the PROS-based strategy provides a good answer, except for some mislocalization in the points of transition, where the stream changes its sensitiveness (e.g., from segment *A* to *B*, and from *B* to *C*), and for some false negatives in the one-minute long positive segment *B*. The MFCC- and HOG-based ones, in turn, result in some additional false positives within the 23-s long negative segment *A*, while the TRoF-based alternative presents mislocalization only in the points of transition. Regarding Fig. 7(e) – which depicts the localization quality of the NBC-based late fusion of all available snippet classifiers – the respective answer presents better quality, when compared to the single solutions (they clearly present less black dots), as expected.

5.2. *Violence localization*

Table 4 puts together the obtained results for violence localization over the MediaEval 2014 VSD dataset. We report the MAP 2014, which is the official metric [48].

As one might observe in Table 4(a), in the particular case of violence localization, and in opposition to pornography localization (c.f., Section 5.1), auditory and visual features are equally suitable for the task, with the PROS-based alternative presenting the highest MAP2014. This may be related to the high sound edition quality of Hollywood movies, which also follow a well-established grammar for affecting spectators. Moreover, we also verify that motion is an important feature for violence detection. While in the pornographic case, the HOG- and TRoF-based solutions are equivalently good, for the violence localization case, the still-image HOG-based solution presents a much inferior result, when compared to the space-temporal TRoF-based one.

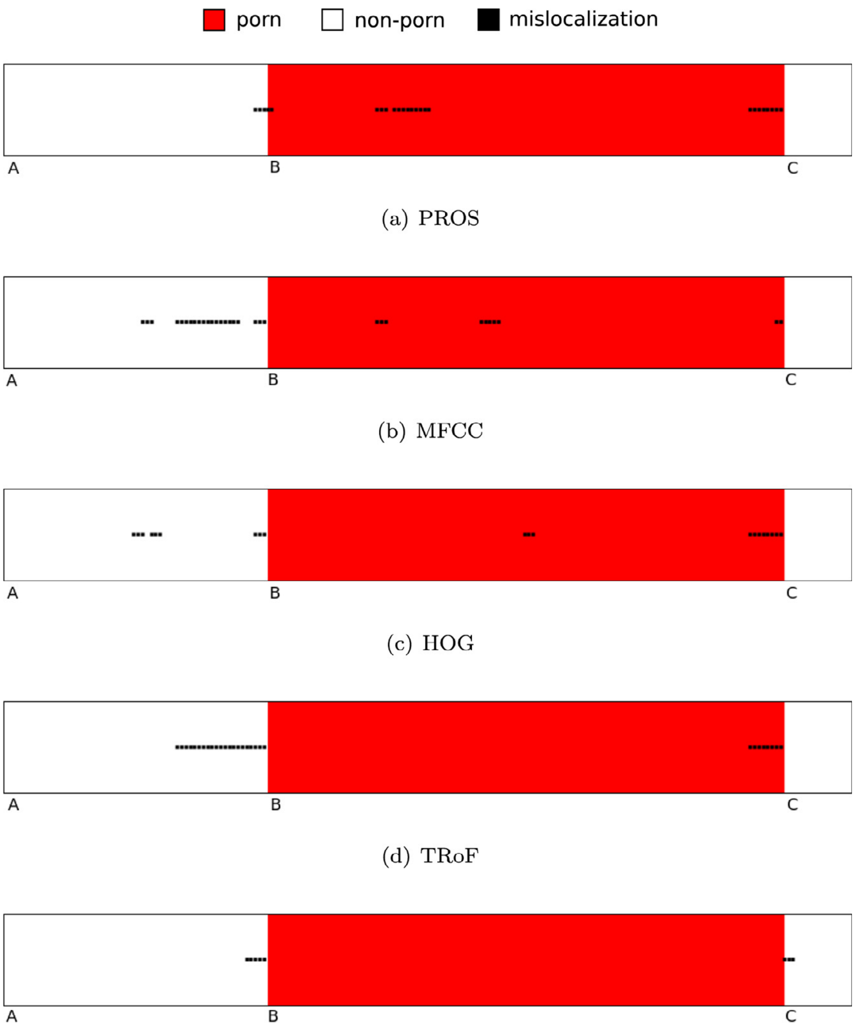
In Table 4(b), we present the results of combining same-nature solutions. Contrary to the cases of pornography localization, for violence localization, the THR fusion of same-nature features does not perform well. It leads to worse results of MAP2014, when compared to the best counterpart single solutions. Nevertheless, the NBC- and SVM-based fusions of features lead to better results, specially in the NBC case. For instance, in the case of visual features (HOG and TRoF), the NBC fusion yields an improvement in MAP2014 of around 58% and 18%, when compared to the solely HOG- and TRoF-based solutions, respectively.

In Table 4(c), we present the results of combining snippet classifiers that rely upon features of different nature. We investigate the combination of the best auditory feature with the best visual one (PROS + TRoF). Alternatively, we also exploit a complete fusion setup, with all the four available snippet classifiers (referred to as ALL). When combining all the features, the fusions are not clearly better than exclusively combining only auditory (PROS + MFCC), or only visual features (HOG + TRoF solutions), either with NBC or with SVM. More importantly, though, the multimodal combination of PROS (auditory) and TRoF (visual) leads to the best solution. The SVM-PROS + TRoF combination provides an MAP2014 of 0.502. It indicates the auditory PROS-based snippet classifier produces hits on the occasions for which the visual TRoF-based one misses, and vice-versa.

Finally, in Table 4(d), we report three works from the literature, which also adopt the MediaEval VSD dataset. As one might observe, we report a more modest value for the official competition metric, although not very different from the mentioned publications. Nevertheless, in face of such numbers, there are some considerations that we deem important to take into account, when analyzing such performances. First and foremost, all three works make use of more than one combination of diverse content classifiers, that rely upon several auditory and visual features. Within those features, the use of time consuming space-temporal approaches is prime for obtaining a high effectiveness, specially regarding Dense Trajectories [25]. Additionally, the works of Lam et al. [45], and of Dai et al. [46] also rely upon the deployment of complex deep neural networks, for obtaining the reported results. The SVM-PROS + TRoF solution, on the contrary, relies upon the use of only two classifiers, which individually present low-memory footprint and small processing time. While TRoF was conceived targeting efficient video description (c.f., [53]), prosody is an auditory feature that presents the impressive characteristic of delivering only three values for each low-level feature vector. To the best of our knowledge, no other low-level descriptor presents such a low-memory footprint.

[3] ACC error reduction from 12.75% (100% − 87.25%) to 9.26% (100% − 90.74%).

[4] ACC error reduction from 13.53% (100% − 86.47%) to 9.26% (100% − 90.74%).



(a) PROS

(b) MFCC

(c) HOG

(d) TRoF

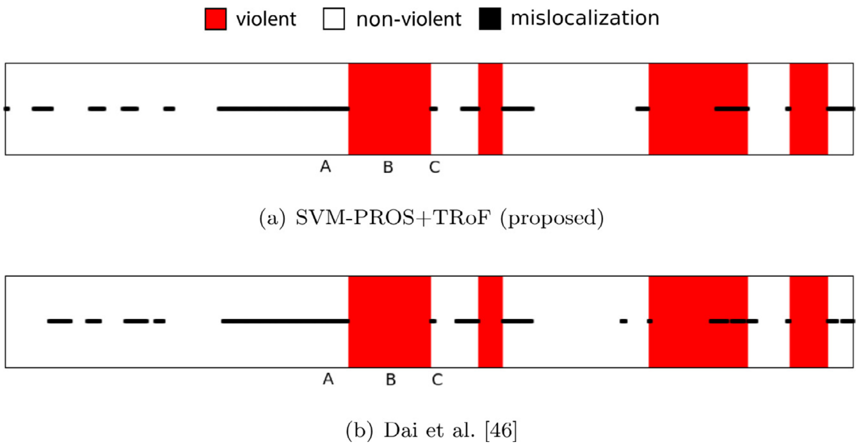
(e) ALL (NBC)

Fig. 7. Localization quality over the same 1.5-min long Pornography-2k video sample. Red and white areas depict the localization groundtruth: red for positive, and white for negative. Black dots represent the mislocalization of each technique: the lesser the quantity of black dots, the better the result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Results of violence localization over the MediaEval 2014 VSD dataset. We report the MAP2014 official metric over the official seven-title test set. In (a), results were obtained without fusion of snippet classifiers. In (b), results denote the fusion of snippet classifiers that rely upon features of the same nature (auditory or visual). In (c), results refer to the multimodal fusion of snippet classifiers. In (d), we report the best results from the literature. The best results are highlighted in bold.

(a) No fusion			(b) Same-nature fusion			(c) Multimodal fusion			(d) Literature	
		MAP2014			MAP2014			MAP2014		MAP2014
Audio	PROS	0.402	THR	PROS + MFCC	0.374	THR	PROS + TRoF	0.460	Dai et al. [46]	0.630
	MFCC	0.288		HOG + TRoF	0.324		ALL	0.406	Zhang et al. [42]	0.566
Image	HOG	0.299	NBC	PROS + MFCC	0.453	NBC	PROS + TRoF	0.488	Lam et al. [45]	0.564
	TRoF	0.401		HOG + TRoF	0.473		ALL	0.476		
			SVM	PROS + MFCC	0.419	SVM	PROS + TRoF	**0.502**		
				HOG + TRoF	0.406		ALL	0.397		

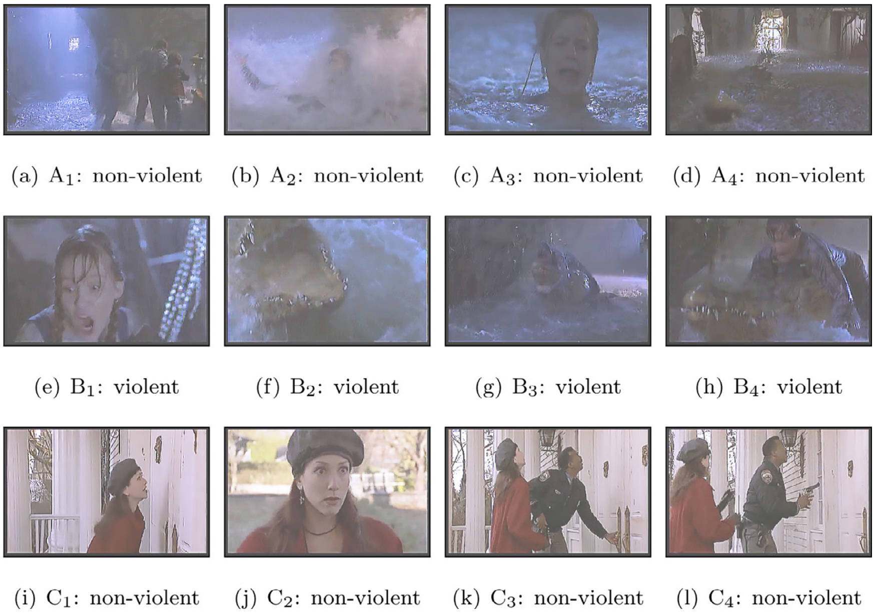
For the sake of illustration, we present a qualitative evaluation of violence localization, over ten minutes that were selected from the *Jumanji* movie title, using the best-performing multimodal solution (SVM-PROS + TRoF) and the best 2014 MediaEval VSD task competitors' solution (Dai et al. [46]). As one might observe, results are qualitatively similar. Fig. 8 depicts the ten-minute timeline, with groundtruth and system answer, in a similar fashion to Fig. 7. For both cases, along the observed ten minutes of video footage, we have many occurrences of false positives (black dots lying within the white regions), and of false negatives (black dots lying within red regions). Moreover, except for the first quarter of the footage at hand – which presents arbitrary false positives – the mislocalizations are concentrated around the regions of label transition (i.e., the instants when the scene changes from positive to negative, or vice-versa). To understand the eventual difficulties faced by the proposed solution over transition regions, we focus on a particular sequence of the footage, which is related



(a) SVM-PROS+TRoF (proposed)

(b) Dai et al. [46]

Fig. 8. Localization quality over a ten-minute long footage that was sampled from the *Jumanji* movie title. In (a), we depict the localization provided by the proposed multimodal SVM-PROS + TRoF solution. In (b), we depict the localization provided by Dai et al. [46], the best 2014 MediaEval VSD task competitors. Results are qualitatively similar. Red and white areas depict the localization groundtruth: red for positive, and white for negative. Black dots represent the mislocalization. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) A_1: non-violent (b) A_2: non-violent (c) A_3: non-violent (d) A_4: non-violent

(e) B_1: violent (f) B_2: violent (g) B_3: violent (h) B_4: violent

(i) C_1: non-violent (j) C_2: non-violent (k) C_3: non-violent (l) C_4: non-violent

Fig. 9. Frames sampled from the *Jumanji* movie title. In (a–d), we have a prior sequence of false positive frames that were sampled from segment *A*, within Fig. 8. In (e–h), we have a middle sequence of true positive frames that were sampled from segment *B*. In (i–l), we have a posterior sequence of false positive frames that were sampled from segment *C*. All images are copyrighted and therefore belong to Sony/Columbia.

to the segments *A, B*, and *C*, and the transitions thereof.

Fig. 9 depicts some frames that comprise such segments. In Fig. 9(a–d), we have the frames related to segment *A*, which is non-violent, although such frames are mistakenly labeled as positive. In Fig. 9(e–h), in turn, we have the frames related to segment *B*, which is violent, and whose frames are correctly identified as such. Finally, in Fig. 9(i–l), we have the frames related to segment *C*, which is non-violent, in spite of such frames being labeled as positive.

The violent scene – which is correctly detected and is related to Fig. 9(e–h) and to segment *B* – depicts a situation with panicked people, who are being attacked by an alligator. Prior to that, segment *A* – represented by Fig. 9(a–b) – depicts a scene with the same studio setup of segment *B*. Although the groundtruth tells the opposite, the action already shows a flooded room, with apprehensive players and motion on water. In such context, one might argue that the scene is already tense, indeed indicating a difficult transition. Posterior to the violent scene, the studio setup changes completely, becoming outdoor (see Fig. 9(i–l), which refers to segment *C*). However, we point out some elements that may turn such transition also difficult to cope with. First, the players are clearly tense, what might be captured by the PROS descriptor. Second, the police officer is holding a gun – see Fig. 9(l) – which is an action that is present in many positive scenes throughout the dataset. Although this is a small example, considering the size of the MediaEval 2014 VSD dataset, it hints at how difficult the localization task is.

6. Conclusions and future work

In this work, we tackled the problem of localizing sensitive content, in the sense of pointing out when a scene starts and ceases to display sensitive material. We proposed a late-fusion pipeline that is able to combine diverse snippet classifiers, even if they rely on different data modalities (e.g., audio, video, etc.). Moreover, the pipeline is of general purpose, as it can be easily adapted for various types of sensitive content.

For validation, we analyzed the localization capability of the pipeline for pornography and for violence, two of the commonest types of sensitive content. From the experiments, we verified important differences between the two concepts. For pornography localization, audio is negligible, and space-temporal features perform as well as still-image features. As discussed in the text, the audio aspect might be related to the abundance of pornographic amateur content in the used dataset, whose audio streams have nothing to do with the visual content, due to poor edition, compression, or stealth purposes. As for using space-temporal vs. still-image features, the best approach actually refers to a combined use of both, as they seem to be complementary, in the pornographic case. For violence localization, in turn, audio is key for improving effectiveness, and space-temporal approaches strongly outperform still-image solutions. In this case, it is worth mentioning the adopted violence dataset is mostly composed of Hollywood titles, which present professional special sound effects, and controlled camera pace rates. The datasets for pornographic and for violent content localization are thus really distinct, not only in content, but also in film grammar (studio vs. amateur).

In any case, the fusion pipeline could be nicely adapted for each situation, while still relying upon the classification and fusion of multimodal time-overlapping video snippets. Nevertheless, more than pornography and violence, the representatives of sensitive content are untold, including – only to name a few – child abuse, upskirt filming, elder abuse, child pornography, cruelty to animals, humiliation, murder, etc. All the remaining sensitive concepts are out there to be analyzed, and we firmly believe the new proposed fusion formulation for sensitive media analysis will be highly useful in several applications.

In addition, with prior and present works, most takes on the sensitive content analysis problem adopt one of two fronts: (i) as a decision problem; or (ii) as a search problem, which is related to the task of sensitive scene localization. In this vein, a third front could be devoted to treating the problem as an optimization one, whereby one might want to localize not *any occurrence* of sensitive content, within a target video stream, but rather the occurrence of a *particular* one, which minimizes the cost, or maximizes the gain of a problem-dependent objective function. That is useful, for instance, in forensic setups, in which one might want to track the behavior of a particular person, which had been previously identified as a criminal or as a highly important suspect to understanding a specific event of interest. Another application is in the movie industry and entertainment, whereby an enthusiastic or pundit might want to see only the scenes from a target stream in which a specific actor or actress appears.

Finally, given that our initial grand objective regarded designing non-GPU-based solutions that are amenable to deployment on hardware-constrained mobile devices (e.g., tablets and smartphones), we focused mostly on employing solutions with low-memory footprint and small runtime. However, taking into consideration the current popularization and impressive results of deep neural networks, it is worth considering — as future work — putting them in perspective with the solutions proposed herein, as well as investigating appropriate forms of combining them and exploring their complementarity, if existent.

Acknowledgments

We thank the financial support of the Brazilian Council for Scientific and Technological Development – CNPq (Grants #477662/2013-7 and #304472/2015-8), the São Paulo Research Foundation – FAPESP (DéjàVu Grant #2017/12646-3), and the Coordination for the Improvement of Higher Level Education Personnel – CAPES (DeepEyes project). Finally, part of the results presented in this paper was obtained through the project “Sensitive Media Analysis”, sponsored by Samsung Eletrônica da Amazônia Ltda., in the framework of law No. 8248/91.

References

[1] J. Wilkinson, A 12-year-old Girl Live-streamed Her Suicide. It Took Two Weeks for Facebook to Take the Video Down, 2017, https://www.washingtonpost.com/news/the-intersect/wp/2017/01/15/a-12-year-old-girl (Accessed 13 February 2017).
[2] C. Miller, A. Burch, Another Girl Hangs Herself whizle Streaming it Live – This Time in Miami, 2017, http://www.miamiherald.com/news/local/article128563889.html (Accessed 13 February 2017).
[3] CNN, Man Shot, Killed While Live-streaming, 2016, (http://www.cnn.com/videos/us/2016/06/17/man-shot-killed-while-live-streaming-orig-vstan-dlewis.cnn (Accessed 13 February 2017)).
[4] S. Almasy, S. Essaid, Norfolk Men Shot While Streaming Video on Facebook Live, 2016, http://www.cnn.com/2016/07/13/us/norfolk-facebook-live-shooting/ (Accessed 13 February 2017).
[5] E. Grinberg, Chicago Torture: Facebook Live Video Leads to 4 Arrests, 2017, http://www.cnn.com/2017/01/04/us/chicago-facebook-live-beating/ (Accessed 13 February 2017).
[6] M. McPhate, Teenager is Accused of Live-Streaming a Friends Rape on Periscope, 2016, https://www.nytimes.com/2016/04/19/us/periscope-rape-case-columbus-ohio-video-livestreaming.html (Accessed 13 February 2017).
[7] K. McLaughlin, High Schoolers, 14 and 15, Live Streamed Themselves Having a Threesome on Facebook While their Friends Watched at School, 2016, http://www.dailymail.co.uk/news/article-3589385/High-schoolers-14-15 (Accessed 13 February 2017).
[8] Council on Communications and Media, Policy statement – media violence, AAP Pediatr. 124 (5) (2009) 1495–1503.
[9] P. Vitorino, S. Avila, M. Perez, A. Rocha, Leveraging deep neural networks to fight child pornography in the age of social media, J. Vis. Commun. Image Represent. 50 (2018) 303–313.
[10] International Centre for Missing & Exploited Children, Child Pornography: Model Legislation & Global Review, 2016 http://www.icmec.org/wp-content/uploads/2016/02/Child-Pornography-Model-Law-8th-Ed-Final-linked.pdf (Accessed 13 February 2017).
[11] E. Barnett, I. Hollingshead, 2012. The Dark Side of Facebook, http://www.telegraph.co.uk/technology/facebook/9118778/The-dark-side-of-Facebook.html (Accessed 13 February 2017).
[12] M. Sjöberg, B. Ionescu, Y.-G. Jiang, V.L. Quang, M. Schedl, C.-H. Demarty, The MediaEval 2014 affect task: violent scenes detection, Proceedings of the MediaEval Workshop, Working Notes, (2014), pp. 1–2.
[13] C.-H. Demarty, C. Penet, M. Schedl, B. Ionescu, V.L. Quang, Y.-G. Jiang, The MediaEval 2013 affect task: violent scenes detection, Proceedings of the MediaEval Workshop, Working Notes, (2013), pp. 1–2.
[14] M. Short, L. Black, A. Smith, C. Wetterneck, D. Wells, A review of internet pornography use research: methodology and content from the past 10 years, Cyberpsychol. Behav. Soc. Netw. 15 (1) (2012) 13–23.
[15] World Health Organization, WHA49.25 – Prevention of Violence: A Public Health Priority, 1996, http://www.who.int/violence_injury_prevention/resources/publications/en/WHA4925_eng.pdf (Accessed 13 February 2017).
[16] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, Pornography classification: the hidden clues in video space-time, Forensic Sci. Int. 268 (2016) 46–61.
[17] C. Snoek, M. Worring, A. Smeulders, Early versus late fusion in semantic video analysis, Proceeedings of ACM International Conference on Multimedia, (2005), pp. 399–402.
[18] C. Snoek, M. Worring, Concept-based video retrieval, Found. Trends Inf. Retr. 2 (4) (2009) 215–322.
[19] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, M. Shah, High-level event recognition in unconstrained videos, Int. J. Multimed. Inf. Retr. 2 (2) (2013) 73–101.
[20] V. Vapnik, Statistical Learning Theory, 1 Wiley, 1998.
[21] D. Lowe, Object recognition from local scale-invariant features, Proceeedings of IEEE International Conference on Computer Vision (ICCV), 2 (1999), pp. 1150–1157.
[22] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, Proceeedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2005), pp. 886–893.
[23] S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Acoust. Speech Signal Process. 1 (1) (1980) 357–366.
[24] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2) (2005) 107–123.
[25] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (1) (2013) 60–79.
[26] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, Proceeedings of International Conference on Computer Vision (ICCV), 2 (2003), pp. 1470–1477.
[27] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, Proceeedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2010), pp. 3304–3311.
[28] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher kernel for large-scale image classification, Proceeedings of European Conference on Computer Vision (ECCV), (2010), pp. 143–156.
[29] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, Video pornography detection through deep learning techniques and motion information, Neurocomputing 230 (2017) 279–293.
[30] S. Russel, P. Norvig, Artificial Intelligence: A Modern Approach, third ed., 1 Prentice Hall, 2010.
[31] A. Lopes, S. Avila, A. Peixoto, R. Oliveira, M. Coelho, A. Araújo, Nude detection in video using bag-of-visual-features, Proceeedings of IEEE Conference on Graphics, Patterns and Images (SIBGRAPI), (2009), pp. 224–231.
[32] C. Caetano, S. Avila, W. Schwartz, S. Guimarães, A. de A. Araújo, A mid-level video representation based on binary descriptors: a case study for pornography detection, Neurocomputing 213 (2016) 102–114.
[33] C. Caetano, S. Avila, S. Guimarães, A. Araújo, Pornography detection using bossanova video descriptor, Proceeedings of IEEE European Signal Processing Conference (EUSIPCO), (2014), pp. 1681–1685.
[34] S. Avila, N. Thome, M. Cord, E. Valle, A. Araújo, Pooling in image representation: the visual codeword point of view, Comput. Vis. Image Underst. 117 (5) (2013) 453–465.
[35] F. Souza, E. Valle, G. Cámara-Chávez, A. Araújo, An evaluation on color invariant based local spatiotemporal features for action recognition, Proceeedings of IEEE Conference on Graphics, Patterns and Images (SIBGRAPI), (2012), pp. 31–36.
[36] A. Ulges, C. Schulze, D. Borth, A. Stahl, Pornography detection in video benefits (a lot) from a multi-modal approach, Proceeedings of ACM International Workshop on Audio and Multimedia Methods for Large-scale Video Analysis (AMVA), (2012), pp. 21–26.
[37] C. Jansohn, A. Ulges, T. Breuel, Detecting pornographic video content by combining image features with motion information, Proceeedings of ACM International Conference on Multimedia (MM), (2009), pp. 601–604.
[38] I. Mironică, I. Duţă, B. Ionescu, N. Sebe, A modified vector of locally aggregated

descriptors approach for fast video classification, Multimed. Tools Appl. 1 (1) (2015) 1–28.

[39] N. Derbas, G. Quénot, Joint audio-visual words for violent scenes detection in movies, Proceeedings of ACM International Conference on Multimedia Retrieval (ICMR), (2014), pp. 1–4.

[40] E. Acar, F. Hopfgartner, S. Albayrak, Violence detection in hollywood movies by the fusion of visual and mid-level audio cues, Proceeedings of ACM International Conference on Multimedia (MM), (2013), pp. 717–720.

[41] E. Bermejo, O. Deniz, G. Bueno, R. Sukthankar, Violence detection in video using computer vision techniques, Proceeedings of Computer Analysis of Images and Patterns, (2011), pp. 332–339.

[42] B. Zhang, Y. Yi, H. Wang, J. Yu, MIC-TJU at MediaEval Violent Scenes Detection (VSD) 2014, Proceedings of the MediaEval Workshop, Working Notes, (2014), pp. 1–2.

[43] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, Proceeedings of Advances in Neural Information Processing Systems (NIPS), (2012), pp. 1097–1105.

[44] M. Moustafa, Applying deep learning to classify pornographic images and videos, Proceeedings of Pacific Rim Symposium on Image and Video Technology (PSIVT), (2015), pp. 1–10.

[45] V. Lam, S. Phan, D.-D. Le, D. Duong, S. Satoh, Evaluation of multiple features for violent scenes detection, Multimed. Tools Appl. 1 (1) (2016) 1–25.

[46] Q. Dai, Z. Wu, Y.-G. Jiang, X. Xue, J. Tang, Fudan-NJUST at mediaeval 2014: violent scenes detection using deep neural networks, Proceedings of the MediaEval Workshop, Working Notes, (2014), pp. 1–2.

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

[48] M. Schedl, M. Sjöberg, I. Mironică, B. Ionescu, V. Lam, Y.-G. Jiang, C.-H. Demarty, VSD2014: a dataset for violent scenes detection in hollywood movies and web videos, Proceeedings of IEEE International Workshop on Content-Based Multimedia Indexing (CBMI), (2015), pp. 1–6.

[49] 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, Proceedings of the MediaEval Workshop, Working Notes, (2014), pp. 1–2.

[50] I. Laptev, M. Marszałek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, Proceeedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2008), pp. 1–8.

[51] V. Lam, D.-D. Le, S. Phan, S. Satoh, D. Duong, NII-UIT at mediaeval 2014: violent scenes detection affect task, Proceedings of the MediaEval Workshop, Working Notes, (2014), pp. 1–2.

[52] 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, Proceedings of the MediaEval Workshop, Working Notes, (2015), pp. 1–2.

[53] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, Temporal robust features for violence detection, Proceeedings of IEEE Winter Conference on Applications of Computer Vision (WACV), (2017), pp. 392–399.

[54] P. Atrey, A. Hossain, A.E. Saddik, M. Kankanhalli, Multimodal fusion for multimedia analysis: a survey, Multimed. Syst. 16 (6) (2010) 345–379.

[55] R. Raina, Y. Shen, A. Mccallum, A. Ng, Classification with hybrid generative/discriminative models, Proceeedings of Advances in Neural Information Processing Systems (NIPS), (2003), pp. 1–8.

[56] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A Practical Guide to Support Vector Classification, 2003, (https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (accessed June 6, 2016)).

[57] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, MIT Adv. Large Margin Classif. 10 (3) (1999) 61–74.

[58] T. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, ACM Neural Comput. 10 (7) (1998) 1895–1923.

[59] J. Demšar, Statistical comparisons of classifiers over multiple data sets, ACM J. Mach. Learn. Res. 7 (2006) 1–30.

[60] F. Eyben, M. Wöllmer, B. Schuller, Opensmile – the Munich versatile and fast open-source audio feature extractor, Proceeedings of ACM International Conference on Multimedia (MM), (2010), pp. 1459–1462.

[61] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the Fisher vector: theory and practice, Int. J. Comput. Vis. 105 (3) (2013) 222–245.

[62] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: a library for large linear classification, J. Mach. Learn. Res. 9 (1) (2008) 1871–1874.

[63] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27.