Temporal Robust Features for Violence Detection

Daniel Moreira[1] Sandra Avila[1] Mauricio Perez[1] Daniel Moraes[1]
Vanessa Testoni[2] Eduardo Valle[1] Siome Goldenstein[1] Anderson Rocha[1,*]
[1]University of Campinas, Brazil [2]Samsung Research Institute, Brazil
[*]anderson.rocha@ic.unicamp.br

Abstract

Automatically detecting violence in videos is paramount for enforcing the law and providing the society with better policies for safer public places. In addition, it may be essential for protecting minors from accessing inappropriate contents on-line, and for helping parents choose suitable movie titles for their children. However, this is an open problem as the very definition of violence is subjective and may vary from one society to another. Detecting such nuances from video footages with no human supervision is very challenging. Clearly, when designing a computer-aided solution to this problem, we need to think of efficient (quickly harness large troves of data) and effective detection methods (robustly filter what needs special attention and further analysis). In this vein, we explore a content description method for violence detection founded upon temporal robust features that quickly grasp video sequences, automatically classifying violent videos. The used method also holds promise for fast and effective classification of other recognition tasks (e.g., pornography and other inappropriate material). When compared to more complex counterparts for violence detection, the method shows similar classification quality while being several times more efficient in terms of runtime and memory footprint.

1. Introduction

Violence is a worldwide public health problem, which constantly demands efforts from authorities to provide the population with safer public places [38]. As part of these efforts, experts have been inspecting solutions for performing computer-aided violence detection in camera footage. Such solutions can be useful for supporting crime solving (e.g., suspect identification) — in Forensic scenarios — while alleviating the job of officers.

Regarding the entertainment industry, the exposure to violence in the media (e.g., television and movies) represents a risk to the health of children, contributing to episodes of aggressive behavior and desensitization to violence [12]. In this direction, researchers have also been investigating different forms of providing automated content filtering, of movies and on-line streams, with the aim of supporting video rating and recommendation.

In the last few years, progress in violence detection has been quantified mainly due to the MediaEval Violent Scenes Detection (VSD) task [16]. Among the proposed solutions, the typical approach relies upon spatio-temporal video characterization, since it has long been proven that spatio-temporal descriptors, such as space-time interest points (STIP) [27] and dense trajectories [35], improve the effectiveness of violence classifiers [6, 32].

Nevertheless, the literature of violence detection, in general, lacks of proper performance evaluation. We show that existing spatio-temporal video descriptors normally demand high computational power, thus impairing the final system performance, specially in terms of runtime and memory footprint.

The fast detection of violent content is important in surveillance scenarios (in which, for instance, the real-time identification of violent events shall be determinant for saving lives), and in Forensic scenarios (in which the fast identification of violent content among millions of files shall allow law enforcers to catch red-handed criminals). Moreover, if automated violence detection is transparently performed in low-memory devices, such as smart-phones and tablets, it shall ubiquitously protect audiences without harming the user experience.

This paper explores a fast end-to-end Bag-of-Visual-Words (BoVW)-based framework for violence classification. We adapt Temporal Robust Features (TRoF) [29], a fast spatio-temporal interest point detector and descriptor, which is custom-tailored for inappropriate content detection, such as violence.

This paper is organized as follows. Section 2 discusses the related work, while Sections 3 and 4 presents the framework for violence detection and the custom-tailored spatio-temporal detector and descriptor, respectively. Section 5 details the experimental setup, while Section 6 reports the obtained results. Finally, Section 7 concludes the paper and elaborates on possible future work.

2. Related Work

There is a well-known auditory and visual film grammar of the movie industry, which has been systematically explored by researchers. Pioneer movie-aimed works employed at least one of the following aspects for inferring scene nature: sound effects (e.g., gunshots, explosions, screams), visual effects (e.g., fire and blood), scene and soundtrack pace rates, as indicators of frantic moments [10, 21, 22, 30]. Such publications tested their approaches on Hollywood action movies, but they did not report the same metrics over the same titles. Moreover, in face of the current easiness of recording videos, and considering the growing offer of on-line amateur content, these solutions may fail due to the heterogeneity of material (e.g., illumination conditions, low video quality, and absence of special effects).

Some works in the literature have taken advantage of the BoVW approach for providing more general solutions. For instance, through the use of motion patterns with a BoVW-based approach, Souza et al. [32] addressed the problem of detecting physical violence such as fighting. Similarly, Bermejo et al. [6] employed the BoVW framework with motion scale-invariant feature transform (MoSIFT [11]) for detecting violence in ice hockey clips. However, the clear drawback of these works resides in the fact that they were developed for a very specific type of violence; consequently, the results are not directly comparable.

Aware of the absence of a standard benchmark for violence detection, the MediaEval Benchmarking Initiative[1] provided, in the occasion of proposing the VSD task, a common ground-truth and standard evaluation protocols. Since then, myriad of works were proposed in the literature, aiming at attending the task.

Most VSD task participants relied upon a three-step pipeline for violence detection [1, 3, 14, 18, 26, 28, 39]. The typical configuration consists of: (i) low-level feature extraction from visual, auditory, or textual modalities, (ii) mid-level feature extraction using BoVW-based representations, and (iii) high-level supervised classification.

In the low level, most of the approaches explored both auditory (e.g., mel-frequency cepstral coefficients, a.k.a., MFCC [1, 14, 18, 26, 28, 39]) and visual information (e.g., scale-invariant feature transform, a.k.a., SIFT [26, 39], STIP [18], dense trajectories [14, 26, 39]). Avila et al. [3] additionally incorporated textual features extracted from movie subtitles. In the mid level, the low-level features were frequently encoded using the BoVW approach [1, 18] or the Fisher Vector representation [14, 26, 39]. Finally, in the high level, support vector machines (SVMs) were the most used alternative for classification [1, 14, 18, 26, 28, 39]. All the mentioned solutions performed multimodal fusion of classifiers at the decision level, except for the work of Derbas and Quénot [18], who provided an early combination of visual and auditory features.

None of the mentioned publications, however, assessed performance in terms of memory footprint and processing time. Indeed, aiming at performing fast shot classification, Mironică et al. [28] even gave up using spatio-temporal features, when conducting experiments on the VSD dataset. Taking a different strategy, we still rely upon the spatio-temporal characterization of motion, but we focus on performing efficient low-level feature detection and content description, in terms of low-memory footprint and small processing time, for the task of violence classification.

3. Violence Detection Framework

Violence is an abstract and complex concept, whose translation to visual characteristics is not straightforward. To cope with such complexity, we follow the literature and betake a three-layered BoVW-based approach for reducing the semantic gap between the low-level visual data representation (e.g., pixels), and the high-level concept of violence. Furthermore, we explore a performance-tuned spatio-temporal framework, which is liable to be executed in real-time, even on modest hardware, since it presents low-memory footprint.

Figure 1 depicts the used framework, with the three layers properly chained in a low-to-mid and mid-to-high fashion, from the left to the right. The existence of a visual codebook, and a supervised learning classification model, implies that every system, constructed under the guidance of such framework, shall operate in one of two modes. In the off-line operation, the visual codebook is constructed (or updated) for posterior reference, and the desired behavior of the system is learned from labeled video examples (often referred to as the training phase). In the on-line operation, arbitrary unknown videos are presented to the system for content labeling (a.k.a., test phase), based on the previously learned codebook and classification model.

As one might observe, the first layer is related to the task of video description (Steps A:1 and B:1, in Figure 1), which is usually implemented with the support of local descriptors [34]. Given that we want to push temporal information early on in the low-level stage, we recommend using spatio-temporal descriptors. These descriptors deliver features that somehow encode the variation of the frame pixel values, regarding not only their spatial configuration, but also their disposition along the video time-line (i.e., pixels are analyzed as voxels). STIP [27] and dense trajectories [35] are typical representatives of such descriptors. However, if spatio-temporal data are not parsimoniously used, they lead to a higher computational cost in terms of both processing time and memory footprint. In this context, we employ TRoF, an efficient — and yet effective — spatio-temporal

[1] http://www.multimediaeval.org/

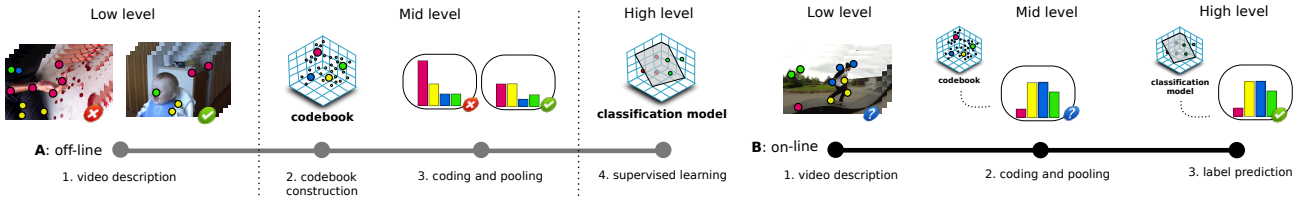


Figure 1. The explored three-layered BoVW-based framework for video violence classification. On the left, the lighter path depicts the "off-line" or "training" phase, in which the system is fed with labeled videos. On the right, the darker path depicts the on-line phase, in which the previously learned model (composed of codebook and classification model) is used to predict the label of new videos.

local video descriptor (c.f., Section 4).

In the mid level, the aim is the combination of the low-level features into global video representations, with intermediate complexity, which are closer to the concept of violence. The mid-level pipeline is broken into two steps [7]: *coding* and *pooling* (Steps A:3 and B:2). The coding step quantizes the low-level descriptors according to a codebook (Step A:2), while the pooling step summarizes the codes obtained into a single feature vector. Here, we rely upon the extraction of Fisher Vectors [31] — one of the best mid-level representations in the Computer Vision literature [9] — which encode the average first- and second-order differences between the low-level descriptions, and the components of a Gaussian mixture model (GMM)-estimated codebook.

Finally, in the high level, the goal is the application of a supervised-learning method, for inducing a proper classification model from previously labeled samples (a.k.a., training dataset), which shall be used to predict the class of any new observation. In the off-line operation, a classifier is trained on the labeled mid-level feature vectors (Step A:4, in Figure 1), and it is further used for predicting the label of unknown vectors, in the on-line operation (Step B:3). At this point, we apply a linear SVM, following the recommendations of the Fisher Vector-related literature [31].

4. Temporal Robust Features

Spatio-temporal local video descriptors usually operate at a high computational cost, in terms of memory footprint and of processing time, thus preventing their execution on limited hardware (e.g., mobile devices). To cope with such limitations, we (i) custom-tailor a video motion detector, which quickly computes an optimized amount of spatio-temporal interest points, and (ii) apply an alternative interest point descriptor, which efficiently represents local motion, namely Temporal Robust Features (TRoF). In Section 4.1, we explain the TRoF detection strategy, while in Section 4.2, we present the TRoF description approach.

4.1. TRoF Detector

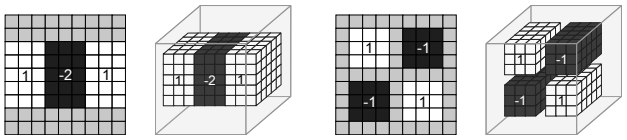
The TRoF detector is inspired by the still-image speeded-up robust features (SURF) [4] detector, which is very fast. The original solution identifies interesting local structures (a.k.a., blobs) with scale σ, on a target image I, by thresholding the determinants of three-variable Hessian matrices $H(x, y, \sigma)$, which are centered at candidate pixels $I(x, y)$. To quickly compute the Hessian determinants, the SURF method replaces the inherent two-dimensional Gaussian second-order derivatives with approximative box filters, which can be readily convolved with the integral image of the target image. This leads to a fast blob detector, yet scale-invariant.

Willems et al. [37] introduced an extension to such mechanism, by adding the time dimension to the Gaussian second-order derivatives. They suggested the use of separated standard deviations for space (σ_s) and for time (σ_t), what led to five-variable Hessian matrices $H(x, y, t, \sigma_s, \sigma_t)$.

In a similar fashion, we also extend the Hessian matrices, but with a different formulation, which is fundamental for real-time operation. In Equation 1, we express the content of a four-variable spatio-temporal Hessian matrix $H(x, y, t, \sigma_{st})$, such as we are adopting in this work. Within it, $L_{xx}(x, y, t, \sigma_{st})$ is the convolution of the Gaussian second-order derivative $\partial^2 G(x, y, t, \sigma_{st})/\partial xx$ with the voxel $\mathbf{x}(x, y, t)$ of the target video. Similarly, $L_{xy}(x, y, t, \sigma_{st})$ refers to the convolution of $\partial^2 G(x, y, t, \sigma_{dt})/\partial xy$ with the voxel $\mathbf{x}(x, y, t)$, and so forth L_{xt}, L_{yt}, L_{yy}, and L_{tt}.

$$H(x, y, t, \sigma_{st}) = \begin{bmatrix} L_{xx}(x, y, t, \sigma_{st}) & L_{xy}(x, y, t, \sigma_{st}) & L_{xt}(x, y, t, \sigma_{st}) \\ L_{xy}(x, y, t, \sigma_{st}) & L_{yy}(x, y, t, \sigma_{st}) & L_{yt}(x, y, t, \sigma_{st}) \\ L_{xt}(x, y, t, \sigma_{st}) & L_{yt}(x, y, t, \sigma_{st}) & L_{tt}(x, y, t, \sigma_{st}) \end{bmatrix}. \quad (1)$$

As one might observe, similar to the detection step proposed by Knopp et al. [24], we employ a single standard deviation σ_{st} for both space and time. At this point, differently from Willems et al. [37], and for a matter of simplification, we adopt a joint strategy that — as a relaxation — lets us



(a) (b) (c) (d)

Figure 2. Original SURF and respective TRoF box filters for approximating Gaussian second-order derivatives. (a) Original SURF $\partial^2 G(x, y, \sigma)/\partial xx$ approximative filter. (b) TRoF $\partial^2 G(x, y, t, \sigma_{st})/\partial xx$ approximative filter. (c) Original SURF $\partial^2 G(x, y, \sigma)/\partial xy$ approximative filter. (d)TRoF $\partial^2 G(x, y, t, \sigma_{st})/\partial xy$ approximative filter.

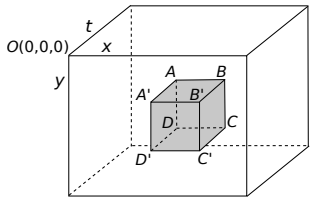


Figure 3. Integral video representation. The outer box represents the video space-time, with the x axis associated to the width, the y axis to the height, and the t axis to the video duration. The inner gray box represents the cuboid region that is calculated by Equation 2.

variate the scale of the detectable blobs faster and closer to the former statement of Bay et al. [4]. We thus apply four octaves of increasing Gaussian spatio-temporal standard deviations, and we perform a fast non-maximal suppression, for gathering the blobs that present the largest Hessian values, within four-dimensional neighborhoods, considering the immediate Hessian neighbors along the x-, y-, t-, and σ_{st}-axes directions.

At first glance, the employment of a joint scale σ_{st} may sound counterintuitive, given the distinct nature of space and time. However, preliminary experiments revealed that, besides the advantage of enabling real-time video description, thanks to the scale-space simplification, such strategy works on par with scale-separated solutions, in the case of detecting inappropriate content. That happens because of the nature of the problem that we intend to solve. While Willems et al. [37] aimed at action recognition, a duty that is fundamentally of specialization nature, we are interested in violence classification, a generalization task that does not require a precise detection of repeatable interest points.

To continue the SURF detection extension to the spatio-temporal case, we substitute the Hessian-related Gaussian second-order derivatives $\partial^2 G(x, y, t, \sigma_{st})/\partial xx$, $\partial^2 G(x, y, t, \sigma_{st})/\partial xy$, $\partial^2 G(x, y, t, \sigma_{st})/\partial xt$, etc., for proper three-dimensional box filters. Figure 2(a) depicts the original SURF box filter that approximates the Gaussian second-order derivative $\partial^2 G(x, y, \sigma)/\partial xx$, with its respective cubic version $\partial^2 G(x, y, t, \sigma_{st})/\partial xx$, in Figure 2(b). Similarly, Figure 2(c) depicts the original box filter that is related to the Gaussian second-order derivative $\partial^2 G(x, y, \sigma)/\partial xy$, with its cubic counterpart $\partial^2 G(x, y, t, \sigma_{st})/\partial xy$, in Figure 2(d). In these examples, all filters approximate Gaussians with $\sigma = \sigma_{st} = 1.2$. For obtaining the remaining four cubic filters, one just needs to apply the proper rotations. Gray filter positions have weight zero in the further convolutions, while white areas are positive, and black are negative.

Finally, to complete the SURF-inspired TRoF detection strategy, we follow the work of Kläser et al. [23], and adequate the notion of flat integral images to the volumetric nature of videos, in order to let their content be efficiently convolved with the Gaussian-approximative cubic filters. That leads to the concept of integral videos. Equation 3 defines the value of an integral video $V_\Sigma(\mathbf{x})$ at a spatio-temporal location $\mathbf{x}(x, y, t)$. It is given by the sum of all voxel values belonging to the video V, which rely in a rectangular cuboid region formed between $\mathbf{x}$ and the video origin.

$$V_\Sigma(\mathbf{x}(x, y, t)) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} \sum_{k=0}^{k \leq t} (i, j, k). \quad (2)$$

Once the integral video is computed, it only takes eight accesses and seven operations to calculate the sum of the voxel values, inside any rectangular cuboid region, independently of its size. For instance, the value V of the volume represented in gray in Figure 3 is given by Equation 3.

$$V = (A + C) - (B + D) - (A' + C') + (B' + D'). \quad (3)$$

With the integral video technique, we convolve box filters of any scale with the video space-time in constant time.

4.2. TRoF Descriptor

The former detection step delivers TRoF blobs, i.e., interest points within the video space-time, which are characterized by their three-dimensional position $P(x, y, t)$, and a spatio-temporal scale σ_{st}. These blobs shall encompass relevant motion phenomena.

To take advantage of such detected regions of interest and use them associated with machine learning solutions, we need to efficiently and effectively describe them mathematically. As suggested in [29], for efficient description, we take only a limited amount of the blob voxels, yet considering their spatio-temporal disposition. We describe only the voxels that belong to three orthogonal planes of interest: the blob-centered spatial $[x, y]$-plane, and the blob-centered temporal $[x, t]$- and $[y, t]$-planes. For effective description, contrary to [29], where SURF descriptors were applied, we

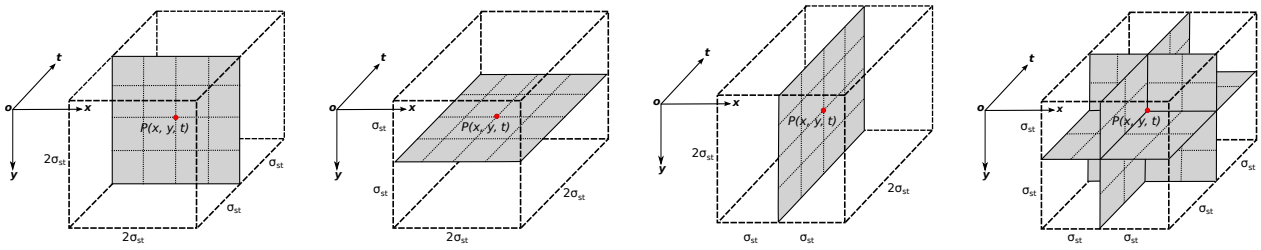


Figure 4. TRoF-described blob planes. The solid gray rectangles are HOG description blocks, which are all centered at the position $P(x, y, t)$, and present a spatio-temporal scale of $2\sigma_{st}$. Each HOG block is divided into 4×4 inner cells, which are represented by internal dashed rectangles. P and σ_{st} come from a formerly detected interest point. (a) HOG block that is projected onto the $[x, y]$ plane. (b) HOG block that is projected onto the $[x, t]$ plane. (c) HOG block that is projected onto the $[y, t]$ plane. (d) Resulting spatio-temporal structure, which is formed by the union of the three HOG blocks.

propose using histograms of oriented gradients (HOG) [15] for capturing the variation of the blob voxel values.

Each plane of interest is described by a HOG block, divided into 4×4 inner cells. Figures 4(a-c) depict each one of these HOG blocks, in the form of solid gray rectangles. As one might observe, each rectangle is properly divided by 4×4 dashed subrectangles, which represent the HOG inner cells. Figure 4(d) depicts the structural union of these three HOG blocks. The resulting structure is inscribed inside a spatio-temporal cuboid, expressed in black dashed lines. Such cuboid is supposed to be linked to a formerly detected interest point: it is centered in the position $P(x, y, t)$ of such point, and has a spatio-temporal scale of $2 \times \sigma_{st}$.

For a low-memory footprint, we limit the number of gradient histogram bins that are calculated in each HOG cell to four. Thus, each HOG block delivers four values for each one of its 4×4 inner cells, leading to a total of 64 description values. With the intent to register eventual correlations among the three HOG blocks, that could be helpful to distinguish violent from non-violent material, we obtain the final TRoF feature vector by concatenating the three 64-dimensional block descriptions, in the following order: $[x, y]$-, $[x, t]$-, and $[y, t]$-plane. As a practical result, the TRoF descriptor yields a set of 192-dimensional feature vectors.

5. Experimental Setup

To validate the designed solution and compare it with the existing methods in the literature, we adopt the MediaEval 2013 Violent Scenes Detection (VSD) dataset [16], which is — to our best knowledge — the most recent dataset that is appropriate for violent shot[2] classification[3]. It comprises 25 Hollywood titles of diverse genres, from extremely violent to musical. Shot segmentation is provided for all the movies, and the resulting segments are individually annotated as containing or lacking violent scenes, which "one would not let an eight-year old child see" [16]. The annotation process was carried out by seven human assessors, with varied ages and cultural backgrounds, and the shot segmentation was obtained through a proprietary software.

The dataset comes separated into a training set, with 18 movies distributed among $32,678$ shots, and a test set comprising seven movies divided into $11,245$ shots. Approximately, 20% of all shots are violent.

The VSD task motivation was the development of systems that could help users choose suitable titles for their children, by retrieving the most violent movie parts, for parental preview [17]. As a consequence, competitors' solutions are compared from the perspective of retrieval: the best performing systems are the ones that return the largest number of violent shots, at the first positions of the top-k retrieved shots, properly ranked by violence classification confidence. For achieving that, the competition suggests using the Mean Average Precision (MAP) at the 100 top ranked violent shots (MAP@100), as the official evaluation metric. Therefore, a solution is considered better than another if it presents a higher MAP@100 value, since it indicates that such solution returns less false positive shots in the first positions of a 100-violent-shot ranked answer. Additionally, as we provide a violence classifier, we also report the area under the receiver operating characteristic curve (AUC).

As the experimental setup, to make the comparisons fair, we use the same mid and high levels — Fisher Vectors computed on 256-word GMM codebooks, and linear SVMs implemented with LIBLINEAR [20] — for all herein evaluated techniques. In the low level, we first pre-process the videos by resizing them to 100 thousand pixels, if larger, similar to Akata et al. [2]. We extract TRoF, and compare

[2] A shot represents a spatio-temporally coherent frame sequence, which captures a continuous action from a single camera.

[3] Further editions of the VSD competition asked participants to *localize* violent scenes, instead of classifying pre-segmented shots.

it with a dense application of HOG, and of STIP descriptors (either detected, or under dense extraction, DSTIP). Regardless of the low-level descriptors, we apply principal component analysis (PCA) to reduce by half their dimensionality, as suggested in [31, 35]. For training, we apply a grid search to find the best C SVM parameter, with $C \in \{2^c : c \in [-5, -3, \ldots, 15]\}$.

HOG details: To provide a controlled baseline for the use of TRoF, we extract HOG descriptions [15], which operate over static images only, with the OpenCV C++ application programming interface (API) [8]. With the FFmpeg library [5], we extract the I-frames, which are densely described with a regular spatial grid, at five scales. Precisely, we use patch sizes of 24, 32, 48, 68 and 96 pixels, with step sizes of 4, 6, 8, 11 and 16 pixels, respectively. Each patch is described by a single HOG block, which is divided into 4×4 HOG cells. Each cell is described by eight bins, leading to $4 \times 4 \times 8$ description values per patch. Hence, the obtained HOG feature vectors are 128-dimensional.

STIP and DSTIP details: For the sake of saving experimental time, we choose STIP [27] as the representative of well-established spatio-temporal local descriptors, instead of dense trajectories [36][4], which are very time- and memory-consuming. In the experiments, we extract both sparse — i.e., Harris-detected (STIP) — and dense STIP (DSTIP) descriptors, with the code provided by Laptev [27].

TRoF details: During the calculation of the integral video, in face of streams with long duration, the sum of voxel values may lead to numerical overflow, besides presenting large-memory footprint. To avoid this, we split the video stream and compute the integral video at every 250 frames. In addition, given that the video streams are very assorted, we cannot find a single Hessian threshold that works for all the cases, when discarding irrelevant blobs. Therefore, we select the 3,000 most relevant blobs within each integral video, after sorting the candidate interest points according to their Hessian values. All the mentioned values were empirically determined.

6. Results

Table 1 shows the results for shot classification on the MediaEval 2013 test dataset. The most successful official competitors employed multimodal approaches, by combining auditory and visual content descriptors. In such works, efficiency was not a major concern, hence all of them combined more than four distinct content descriptors, including still-image approaches (such as HOG), and even two different types of spatio-temporal descriptors, at the same time (such as STIP and dense trajectories), often producing a computationally costly solution in terms of processing time and memory footprint.

[4]The similarities between TRoF and the detector of Willems et al. [37] (both are extensions of the SURF detection process to the spatio-temporal case) would make the former solution a natural choice for performing comparisons. However, since source codes and executables are no longer available, and due to a lack of details in the related paper, we could not manage to reproduce their method in a timely manner.

	Solution	Media Type	MAP@100†	AUC
Competitors	Multimodal [19]	audio & video	0.690	*
	Multimodal [33]	audio & video	0.689	*
	Multimodal [13]	audio & video	0.682	*
	Multimodal [25]	audio & video	0.596	*
	Dense HOG (DHOG)	video only	0.459	0.706
	Detected STIP (STIP)	video only	0.541	0.694
	Dense STIP (DSTIP)	video only	0.588	0.739
	TRoF (Proposed)	**video only**	**0.508**	**0.722**

*Competitors did not report AUC.
†MAP@100 values were obtained with the same evaluation tool of [16].
All multimodal competitors' solutions employed five or more description modalities, with at least one being spatio-temporal.

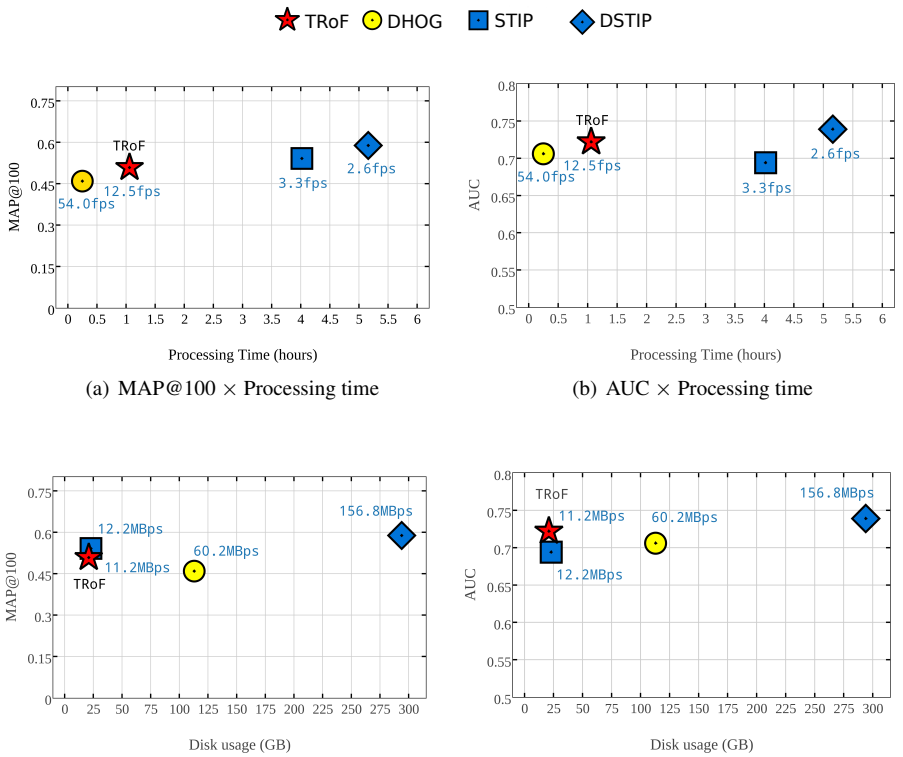
Table 1. Results on the MediaEval 2013 dataset.

On the other hand, in our case, we limited the explored classifiers to the use of a single modality (either DHOG, STIP, DSTIP, or TRoF), in order to investigate how one may deal with the effectiveness-vs.-efficiency tradeoff.

Single modal solutions presented reasonable classification effectiveness, with the DSTIP-based method presenting the highest AUC (0.739). TRoF scored second best (0.722). Regarding MAP, as expected, the still-image approach presented the worst results, confirming that the spatio-temporal information improves violence classification. DSTIP again scored best — among the single modal solutions — approaching even some multimodal performances (c.f., [25], in Table 1). TRoF, in turn, obtained a more modest MAP@100 value (0.508); notwithstanding, it presents the best performance, in terms of processing time and memory footprint, as we shall see shortly.

To further investigate the required computational time, Figure 5(a) depicts the correlation between MAP@100 and the computational time spent to classify a selected portion of 30 minutes of violent content, for each evaluated classifier. Figure 5(b), in turn, correlates the AUC with the computational time. In both charts, internal values indicate the processing frame rate, in frames per second (fps). The higher the rate, the better the solution. As one might observe, TRoF leads to a rate of 12.5 fps, in spite of being spatio-temporal.

Likewise, for evaluating the strategies in terms of memory footprint, we also correlate MAP@100 and the AUC with respect to the total disk space that is spent to store the low-level feature vectors of the entire test dataset, which consists of 15 hours of video footage. Figure 5(c) shows the correlation between MAP@100 and disk usage, while



(a) MAP@100 × Processing time

(b) AUC × Processing time

(c) MAP@100 × Disk usage

(d) AUC × Disk usage

Figure 5. Performance on the MediaEval 2013 dataset (shot classification) for effectiveness vs. efficiency. On the left, effectiveness denotes MAP@100, while on the right it denotes AUC. On the top row, efficiency denotes computational time spent to classify a selected portion of 30-minute video footage (same shots for all methods). Internal values indicate the processing frame rate, in frames per second (fps). The higher the rate, the better the solution. On the bottom row, efficiency concerns disk storage space for the entire test dataset (15 hours of video footage). Internal values indicate the amount of generated description megabytes per second of movie (MBps). The smaller the amount, the better the solution. In all charts, the best solutions are at the top-left corner.

Figure 5(d) depicts the correlation between AUC and disk usage. In both charts, internal values indicate the amount of generated description in megabytes per second of footage (MBps). The smaller the amount, the better the result.

In all charts, the best solutions occur on the top left regions: they present high performance, despite of spending less computational resources. In all the cases, TRoF is near such privileged region. All experiments were conducted on a 64-bit Linux machine, powered by a 2-GHz 12-core Intel(R) Xeon(R) processor (E5-2620), with 24 GB of RAM.

Although we do not have the proper time and disk usage measurements of the MediaEval 2013 competitors' solutions, we can still infer their performance from the implemented STIP and DSTIP solutions, because they are included among the many modalities that were employed by those works. STIP and DSTIP can be seen as lower-bounds to the time and disk usage of these solutions.

Finally, the numbers of TRoF in face of the MediaEval 2013 VSD dataset are promising. It presents the same memory footprint of STIP, despite being four times faster, and presenting reasonable values of MAP@100 and AUC.

7. Conclusion

Violent video classification is a problem that has gained attention from the scientific community, due to its relevance. Specially in the last few years, progress in the field has been quantified mainly due to the MediaEval VSD task [16]. Among the proposed solutions, the typical approach relies upon multimodal video characterization, with the compulsory use of spatio-temporal descriptors as one of the modalities. That happens because it has long been proven that spatio-temporal descriptors — such as STIP and dense trajectories — improve the effectiveness of violence detectors. We indeed verify it through the still-image HOG-based solution, which presents the worst MAP@100 experimental results.

In this vein, although the general perception in the literature dictates that spatio-temporal techniques are normally computationally expensive and present a high-memory footprint, the research on violence detection, in general, lacks performance evaluation. In opposition to that, we report the performance of the explored solutions in terms of

spent processing time and memory footprint. The results have shown that the TRoF usage yields a processing frame rate capacity of nearly 12 fps, and that it spends a memory amount of nearly 11 megabytes per second of described footage, in spite of being spatio-temporal.

The performance of TRoF is possible mainly due to two aspects. First, a four-variable spatio-temporal Hessian matrix, which uses a spatio-temporal standard deviation that is shared between space and time, for detecting the scale of interesting phenomena. Second, a fast description of the detected spatio-temporal interest points, which slices up the detected region in three perpendicular planes $[x, y]$, $[x, t]$, and $[y, t]$ of interest. Each plane is further described with histograms of gradients, yielding a compact descriptor in $\Re^{192}$. This efficient representation, allied with a sparse set of detected and representative points, allow us to properly capture the most important motion-related aspects of a target video sequence.

As the shared spatio-temporal scale parameter is key for the performance gain, one might wonder when it is interesting to apply it. In our experience, we have learned that whenever we have a generalization problem (e.g., generalizing video motion to more general concepts, such as pornography and violence) this representation is appropriate. On the other hand, when we have a specialization problem (e.g., using video motion for detecting specific actions, such as gestures, walking, running, jumping, etc.), possibly an untangled representation for the space and time scales would be more appealing.

Although the current provided method yields a classification quality of about 72% that may appear far from ideal, it only shows the difficulty of the problem, when compared to other existing solutions. The obtained results with TRoF hold promise for deployment in mobile devices for on-demand video analysis, and for filtering with faster detection rates than its counterparts. For that, future directions include further refining the detector and descriptor to increase their discriminability, and also for parallelizing some of their steps using available graphics processing units in the used devices. Taking into consideration the current popularization and impressive results of deep neural networks, it is worth considering putting them in perspective with the solution explored herein, as well as investigating appropriate forms of combining them, and exploring their complementarity, if existent. Finally, exploring other generalization tasks is also a future work worth pursuing.

Acknowledgments

Part of the results presented in this paper were obtained through the project “Sensitive Media Analysis”, sponsored by Samsung Eletrônica da Amazônia Ltda., in the framework of law No. 8,248/91. We also thank the financial support of the Brazilian Council for Scientific and Technological Development – CNPq (Grants #477662/2013-7, #304472/2015-8), the São Paulo Research Foundation – Fapesp (DéjàVu Grant #2015/19222-9), and the Coordination for the Improvement of Higher Level Education Personnel – CAPES (DeepEyes project).

References

[1] E. Acar, F. Hopfgartner, and S. Albayrak. Violence Detection in Hollywood Movies by the Fusion of Visual and Mid-level Audio Cues. In *ACM International Conference on Multimedia*, pages 717–720, 2013. 2

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Good Practice in Large-Scale Learning for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):507–520, 2014. 5

[3] S. Avila, D. Moreira, M. Perez, D. Moraes, I. Cota, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha. RECOD at MediaEval 2014: Violent Scenes Detection Task. In *MediaEval*, 2014. 2

[4] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-Up Robust Features (SURF). *ACM Computer Vision and Image Understanding*, 110(3):346–359, 2008. 3, 4

[5] F. Bellard. FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. `http://www.ffmpeg.org`, 2016. 6

[6] E. Bermejo, O. Deniz, G. Bueno, and R. Sukthankar. Violence Detection in Video Using Computer Vision Techniques. In *Springer Computer Analysis of Images and Patterns*, pages 332–339, 2011. 1, 2

[7] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010. 3

[8] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 1(1):1–6, 2000. 6

[9] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVABritish Machine Vision Conference*, pages 1–12, 2011. 3

[10] L.-H. Chen, H.-W. Hsu, L.-Y. Wang, and C.-W. Su. Violence Detection in Movies. In *IEEE International Conference on Computer Graphics, Imaging and Visualization*, pages 119–124, 2011. 2

[11] M. Chen and A. Hauptmann. MoSIFT: Recognizing Human Actions in Surveillance Videos. Technical report, Carnegie Mellon University, 2009. 2

[12] Council on Communications and Media. Policy Statement – Media Violence. *AAP Pediatrics*, 124(5):1495–1503, 2009. 1

[13] Q. Dai, J. Tu, Z. Shi, Y.-G. Jiang, and X. Xue. Fudan at MediaEval 2013: Violent Scenes Detection Using Motion Features and Part-Level Attributes. In *MediaEval*, 2013. 6

[14] Q. Dai, Z. Wu, Y.-G. Jiang, X. Xue, and J. Tang. Fudan-NJUST at MediaEval 2014: Violent Scenes Detection Using Deep Neural Networks. In *MediaEval*, 2014. 2

[15] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005. 5, 6

[16] C.-H. Demarty, B. Ionescu, Y.-G. Jiang, V. Lam, M. Schedl, and C. Penet. Benchmarking Violent Scenes Detection in Movies. In *IEEE International Workshop on Content-based Multimedia Indexing*, pages 1–6, 2014. 1, 5, 6, 7

[17] C.-H. Demarty, C. Penet, M. Schedl, B. Ionescu, V. Lam, and Y.-G. Jiang. The MediaEval 2013 Affect Task: Violent Scenes Detection. In *MediaEval*, 2012. 5

[18] N. Derbas and G. Quénot. Joint Audio-Visual Words for Violent Scenes Detection in Movies. In *ACM International Conference on Multimedia Retrieval*, pages 1–4, 2014. 2

[19] N. Derbas, B. Safadi, and G. Quénot. LIG at MediaEval 2013 Affect Task: Use of a Generic Method and Joint Audio-Visual Words. In *MediaEval*, 2013. 6

[20] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 5

[21] T. Giannakopoulos, A. Makris, D. Kosmopoulos, S. Perantonis, and S. Theodoridis. Audio-visual fusion for detecting violent scenes in videos. In *Springer Artificial Intelligence: Theories, Models and Applications*, pages 91–100, 2010. 2

[22] Y. Gong, W. Wang, S. Jiang, Q. Huang, and W. Gao. Detecting Violent Scenes in Movies by Auditory and Visual cues. In *Springer Advances in Multimedia Information Processing*, pages 317–326, 2008. 2

[23] A. Kläser, M. Marszałek, and C. Schmid. A Spatio-Temporal Descriptor Based on 3D-Gradients. In *BMVA British Machine Vision Conference*, pages 1–10, 2008. 4

[24] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. V. Gool. Hough Transforms and 3D SURF for robust three dimensional classification. In *ACM European Conference on Computer Vision*, 2010. 3

[25] V. Lam, D.-D. Le, S. Phan, S. Satoh, and D. Duong. NII-UIT at MediaEval 2013 Violent Scenes Detection Affect Task. In *MediaEval*, 2013. 6

[26] V. Lam, S. Phan, D.-D. Le, D. Duong, and S. Satoh. Evaluation of multiple features for violent scenes detection. *Springer Multimedia Tools and Applications*, 1(1):1–25, 2016. 2

[27] I. Laptev. On Space-Time Interest Points. *ACM International Journal of Computer Vision*, 64(2):107–123, 2005. 1, 2, 6

[28] I. Mironică, I. Duţă, B. Ionescu, and N. Sebe. A modified vector of locally aggregated descriptors approach for fast video classification. *Springer Multimedia Tools and Applications*, 1(1):1–28, 2015. 2

[29] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha. Pornography classification: The hidden clues in video space–time. *Elsevier Forensic Science International*, 268(1):46–61, 2016. 1, 4

[30] J. Nam, M. Alghoniemy, and A. Tewfik. Audio-visual content-based violent scene characterization. In *IEEE International Conference on Image Processing*, pages 353–357, 1998. 2

[31] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher Kernel for Large-scale Image Classification. In *ACM European Conference on Computer Vision*, pages 143–156, 2010. 3, 6

[32] F. Souza, G. Cámara-Chávez, E. Valle, and A. Araújo. Violence Detection in Video Using Spatio-Temporal Features. In *SBC Conference on Graphics, Patterns and Images*, pages 224–230, 2010. 1, 2

[33] C. Tan and C.-W. Ngo. The Vireo Team at MediaEval 2013: Violent Scenes Detection by Mid-level Concepts Learnt from Youtube. In *MediaEval*, 2013. 6

[34] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Now Publishers Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008. 2

[35] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *ACM International Journal of Computer Vision*, 103(1):60–79, 2013. 1, 2, 6

[36] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, pages 3551–3558, 2013. 6

[37] G. Willems, T. Tuytelaars, and L. V. Gool. An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In *ACM European Conference on Computer Vision*, pages 650–663, 2008. 3, 4, 6

[38] World Health Organization. WHA49.25 – Prevention of violence: a public health priority. http://www.who.int/violence_injury_prevention/resources/publications/en/WHA4925_eng.pdf. 1

[39] B. Zhang, Y. Yi, H. Wang, and J. Yu. MIC-TJU at MediaEval Violent Scenes Detection (VSD) 2014. In *MediaEval*, 2014. 2