Forensic Science International 268 (2016) 46–61

Contents lists available at ScienceDirect

Forensic Science International

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

Pornography classification: The hidden clues in video space–time

Daniel Moreira [a,*], Sandra Avila [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

Article history:
Received 26 February 2016
Received in revised form 8 September 2016
Accepted 12 September 2016
Available online 21 September 2016

Keywords:
Sensitive media
Video pornography classification
Temporal information
Fisher Vector
TRoF

ABSTRACT

As web technologies and social networks become part of the general public's life, the problem of automatically detecting pornography is into every parent's mind — nobody feels completely safe when their children go online. In this paper, we focus on video-pornography classification, a hard problem in which traditional methods often employ still-image techniques — labeling frames individually prior to a global decision. Frame-based approaches, however, ignore significant cogent information brought by motion. Here, we introduce a space-temporal interest point detector and descriptor called *Temporal Robust Features* (TRoF). TRoF was custom-tailored for efficient (low processing time and memory footprint) and effective (high classification accuracy and low false negative rate) motion description, particularly suited to the task at hand. We aggregate local information extracted by TRoF into a mid-level representation using Fisher Vectors, the state-of-the-art model of Bags of Visual Words (BoVW). We evaluate our original strategy, contrasting it both to commercial pornography detection solutions, and to BoVW solutions based upon other space-temporal features from the scientific literature. The performance is assessed using the Pornography-2k dataset, a new challenging pornographic benchmark, comprising 2000 web videos and 140 h of video footage. The dataset is also a contribution of this work and is very assorted, including both professional and amateur content, and it depicts several genres of pornography, from cartoon to live action, with diverse behavior and ethnicity. The best approach, based on a dense application of TRoF, yields a classification error reduction of almost 79% when compared to the best commercial classifier. A sparse description relying on TRoF detector is also noteworthy, for yielding a classification error reduction of over 69%, with 19× less memory footprint than the dense solution, and yet can also be implemented to meet real-time requirements.

1. Introduction

Pornography diffusion over the Internet has systematically increased in recent years [1]. This poses a challenge as web technologies reach broader uses and audiences, since pornographic content is unwelcome in many contexts, especially where underage viewers are concerned.

The need for regulating the diffusion of Internet pornography clashes with the international, distributed, and large-scale nature of the web. Trying to regulate the diffusion from the side of creators and distributors is a sisyphean task. Regulation from the consumer side, in the form of content filtering is more promising, and thus is employed by governments, companies, tutors, and parents against inappropriate access to pornography. If we are to meet the daunting growth rates of content creation, this content-filtering has to be automated.

From the point of view of health and social sciences, the understanding of the impacts of pornography production and consumption on society is still incipient and understudied [2], with inconclusive results [3]. Regardless of that, some modalities of porn are illegal, with child pornography being the obvious case in most countries [4]. Because of that, pornography detection receives growing attention in Law enforcement and Forensic activities. Besides the selection of relevant material for attaching to legal dossiers, detecting pornographic files (i.e., the fast filtering of pornographic content among millions of files) at crime scenes brings great benefits, including the immediate arrest of criminals. Furthermore, once all porn-related files are singled out, we can employ additional techniques such as the ones involving face detection and recognition, child-pornography detection, age estimation, etc., for further selecting videos of higher importance

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

http://dx.doi.org/10.1016/j.forsciint.2016.09.010
0379-0738/© 2016 Elsevier Ireland Ltd. All rights reserved.

for an investigation. The method could be used directly on servers for monitoring, during search-and-seizure for proper confiscation of suspected materials and equipments, or even in police premises to quickly glean over apprehended hard disks, thus decreasing the amount of human police resources currently put into place for this type of analysis.

Most conventional, commercially available, content-filtering solutions regulate the access to pornographic content by blacklisting URLs and looking at metadata (keywords in file names and descriptions, parental advisory metadata, etc.). In contrast, analyzing the visual information itself is mandatory to robust pornography filtering, since the visual information, contrarily to meta-information, is much more difficult to conceal. Therefore, a few off-the-shelf solutions include visual-content analysis in their features [5–8]. However, according to the experimental results we report in this paper, those tools are yet far from being effective.

In the literature, the first efforts for pornography detection conservatively associated pornography to nudity. Since then, plenty of solutions have been proposed, aiming at identifying nude people by the means of skin detection [9–15]. Notwithstanding, those strategies suffer from high rates of false positives in situations of non-pornographic body exposure (e.g., swimming, sunbathing, baby breastfeeding, etc.).

In contrast to nudity detection, in the scope of this work, we want to classify pornography as “any explicit sexual matter with the purpose of eliciting arousal” [1]. In such vein, the current state of the art of pornography classification relies on Bags-of-Visual-Words (BoVW)-based strategies, to reduce the semantic gap between the low-level visual data representation (e.g., pixels), and the high-level target concept of pornography [16–29]. However, it is still very common to extend the still-image solutions to video, by labeling the frames independently, and then thresholding the quantity of sensitive samples [20–24]. That strategy misses opportunities because motion pictures offer extra space–time information, where one can look for additional features. Motion information for example, can be very revealing about the presence of pornographic content. Thus, in this work, we aim at taking a step further by incorporating temporal information to the task of video pornography classification, in a pursuit of more effective and efficient solutions.

This paper proposes an end-to-end BoVW-based framework of video-pornography classification, allowing to incorporate temporal information in different ways, according to different choices of low-level time-aware local descriptors — e.g., Space Temporal Interest Points (STIP) [30], or Dense Trajectories [31] — to BoVW-based mid-level representations for the entire video footage. To perform experiments and validation, we introduce the Pornography-2k dataset, a new challenging pornographic benchmark that comprises 2000 web videos, available upon request and the sign of a proper responsibility agreement.

Additionally, we introduce Temporal Robust Features (TRoF), a novel space-temporal interest point detector and descriptor, which provides a speed compatible with real-time video processing and presents low-memory footprint. TRoF yields essentially the same classification accuracy of Dense Trajectories [31] — the current state-of-the-art space-temporal video descriptor — with 50× less memory footprint.

We organize the remainder of this paper into six sections. In Section 2 we explore related work, while in Section 3 we present the proposed framework to classify video pornography. In Section 4 we introduce TRoF, while in Section 5 we explain the experimental setup. In turn, in Section 6 we discuss the obtained results and, finally, in Section 7 we conclude the paper and elaborate on future work.

2. Related work

In this section, we survey some of the literature on the pornography detection approaches, focusing on BoVW-based approaches and relevant nudity classifiers. Table 1 summarizes these solutions. In addition, we explore commercial tools that block web sites or scan computers for pornographic content.

Table 1
BoVW-based pornography classification. Most results employ different protocols/datasets and are not directly comparable, except for the last seven lines of the table, that employ the Pornography-800 dataset [21].[a]

	Reference	Media	Dataset (#pos/#neg)	Low level		Mid level		High level (SVM kernel)	ACC (%)
				Feature detector	Feature descriptor	Codebook	BoVW		
Image	Lopes et al. [25]	Nude	90/90	SIFT blobs	HueSIFT	*k*-means	Traditional	Linear	84.6
	Steel [27]	Nude	1500/1500	Skin ROIs	Mask-SIFT	*k*-means	Traditional	RBF	[b]
	Deselaers et al. [29]	Porn	1700/6800	SIFT-based blobs	Difference of Gaussians	GMM	Traditional	Hist. intersection	[c]
	Ulges and Stahl [28]	Porn	4248/20,000	Regular grid	DCT	*k*-means	Traditional	χ^2	[c]
	Zhang et al. [26]	Porn	4000/8000	Skin ROIs	Color, texture, intensity	*k*-means	Traditional	Not reported	90.9
	Yan et al. [16]	Porn	20,000/70,000	Skin ROIs	SURF	*k*-means	Traditional	RBF	[d]
	Zhuo et al. [32]	Porn	8000/11,000	Skin ROIs	ORB	*k*-means	Traditional	RBF	93.0
Video	Lopes et al. [24]	Nude	89/90	SIFT blobs	HueSIFT	*k*-means	Traditional	Linear	93.2
	Jansohn et al. [23]	Porn	932/2663	Regular grid	DCT, Motion histogram	*k*-means	Traditional	Not reported	[c]
	Avila et al. [22]	Porn	400/400	Regular grid	HueSIFT	*k*-means	BOSSA	χ	87.1
	Valle et al. [18]	Porn	400/400	STIP blobs	STIP	Random	Traditional	Linear	91.9
	Souza et al. [17]	Porn	400/400	Color-STIP blobs	STIP	Random	Traditional	Linear	91.0
	Avila et al. [21]	Porn	400/400	Regular grid	HueSIFT	*k*-means	BossaNova	χ^2	89.5
	Caetano et al. [20,19]	Porn	400/400	Regular grid	Binary descriptors	*k*-medians	BossaNova	χ^2	90.9
	Caetano et al. [33]	Porn	400/400	Regular grid	Binary descriptors	*k*-medians	BossaNovaVD	χ^2	92.0
	TRoF (this work)	Porn	400/400	3D Hessian blobs	TRoF	GMM	Fisher Vector	Linear	**95.0**

Traditional BoVW mid-level representation is obtained with hard-assignment coding and average pooling. ACC: accuracy; SVM: Support Vector Machine; RBF: Radial Basis Function. In bold, we highlight the result of the present work.

[a] Very recently, Moustafa [34] applied a deep learning technique to classify pornographic content on the Pornography-800 dataset. He achieved an accuracy rate of 94.1%, by using static visual features only and a majority voting scheme.

[b] Uses False Positive Rate (FPR) as evaluation measure.

[c] Uses Equal Error Rate (EER) as evaluation measure.

[d] Uses Receiver Operating Characteristic (ROC) curve as evaluation measure.

The first efforts to detect pornography conservatively associated pornography with nudity, where the solutions tried to identify nude or scantily-clad people [9,11,10,12–15]. In such works, the detection of human skin played a major role, followed by the identification of body parts.

The presence of nudity is not a good conceptual model of pornography. There are non-pornographic situations with plenty of body exposure. Conversely, there are pornographic scenes that involve very little exposed skin. Nevertheless, nudity detection is related to pornography detection, with a vast literature of its own. A comprehensive survey on skin-detection techniques can be found in [35].

More recently, Lopes et al. developed a BoVW approach, which employed the HueSIFT color descriptor (Hue Scale-Invariant Feature Transform), to classify images [25] and videos [24] of nudity. For video classification, they proposed a majority voting scheme over the video frames. Similarly, Steel [27] proposed a BoVW-based nudity detection by using a Gaussian skin masking for feature isolation and the mask-SIFT in a cascading image classification system.

The clear drawback of using skin detectors to identify pornography is the high false-positive rate, specially in situations of non-pornographic body exposure (e.g., swimming, sunbathing, boxing). Therefore, Deselaers et al. [29] proposed, for the first time, to pose pornography detection as a Computer Vision classification problem (akin to object classification), rather than a skin-detection or segmentation problem. They extracted patches around difference-of-Gaussian interest points, and created a visual codebook using a Gaussian Mixture Model (GMM), to classify images into different pornographic categories. Their Bag-of-Visual-Words (BoVW) model greatly improved the effectiveness of the pornography classification.

Moving from nudity detection toward pornography classification, we face the challenge in defining the notion of pornography. The subjectivity of the task is (in-)famously highlighted in the discourse of US Supreme Court Justice Potter Stewart, in *Jacobellis vs. Ohio*.

> "I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description ['hard-core pornography'], and perhaps I could never succeed in intelligibly doing so. But *I know it when I see it*, and the motion picture involved in this case is not that."

This notion of "*I know it when I see it*" became famous to the point of cliche, but it is much too subjective for the purposes of Computer Vision. Therefore, many works [22,18,17,21,20,19] have adopted the definition of pornography proposed by Short et al. [1]: "any explicit sexual matter with the purpose of eliciting arousal", which while still subjective, establishes a set of criteria that allow deciding the nature of the material (sexual content, explicitness, goal to elicit arousal, purposefulness). We, too, endorse this definition and employ it in this work.

Nevertheless, some researchers opted for tackling the problem as a matter of finding porn and non-porn material. Comprehensively, most of them did not delve into defining or adopting a clear concept for pornography, due to the difficulty of such task. For instance, Ulges and Stahl [28] adopted a forensic setup, aimed at the classification of child pornography in images. They densely described the target images in patches, properly submitting them to a DCT (Discrete Cosine Transformation) in the YUV color space, before constructing their visual codebooks.

Influenced by the idea of combining skin detection with BoVW approaches, Zhang et al. [26] employed a skin-color-aware visual attention model to identify image ROIs (Region of Interest), prior to the low-level description process. As such model relied on the detection of faceless skin-toned patches in the compressed domain of the target images, the authors were able to select the yet-to-decompress ROIs that should be effectively described, thus reducing the total time spent with pornographic content filtering. To describe such ROIs, they applied a combination of color-, intensity-, texture-, and skin-based descriptors.

Yan et al. [16] also used a color-aware visual attention model, that relied on the identification of salient and skin-colored faceless image ROIs. For a fast description, the researchers proposed the use of the SURF descriptor (Speeded Up Robust Features) [36].

Similarly, Zhuo et al. [32] proposed a BoVW approach that also focused on the fast description of formerly detected skin-colored regions, by employing the ORB descriptor (Oriented FAST and Rotated BRIEF) [37].

On the occasion of using binary-classification strategies to tackle the problem of pornography detection, each one of the mentioned works adopted a particular interpretation of the pornography concept, besides reporting results on unrelated datasets, preventing direct and fair comparisons amongst different works. Moreover, with the exception of Zhang et al. [26], all these works still inherited the drawbacks of skin-detection-based filters. For instance, they are not useful for recognizing pornographic cartoons (which are very common in pornographic websites, and do not contain live-action[1] human skin).

Avila et al. [22,21] managed to solve the problem of classifying video pornography, in the Pornography-800 dataset, by the occasion of proposing BOSSA [22] and BossaNova [21] (both extensions to the BoVW formalism). They focused on enhancing the BoVW mid-level data representation, by enriching the expression of the HueSIFT descriptors extracted from the target images, with respect to the ones selected from the visual codebook. In both works, they applied a voting scheme based on the classification of the individual video frames.

More recently, Caetano et al. [20,19,33] also tackled the pornography classification problem related to the Pornography-800 dataset. In [20,19], as they maintained the BossaNova technique within their solution, their innovation relied on the use of fast-to-compare binary low-level image descriptors. Moreover, in [33], the authors improved the classification results by also establishing a single bag for the entire target video (an extension to the BossaNova approach), instead of a bag for each extracted video frame.

An advantage of the aforementioned works [22,21,20,19,33] is that they used the Pornography-800 dataset [22], a representative dataset of 800 web videos, available upon request. Thus, the numbers reported are directly comparable with one another. Additionally, different from most approaches, the aforementioned works used an enhanced BoVW model for detecting pornographic video content. As a drawback, however, all of those works only used bags of static features, which ignore significant and cogent information brought by video motion.

Few works have applied space-temporal features or other motion information for the classification of pornography. Valle et al. [18] proposed the use of space-temporal local descriptors (such as STIP descriptor [38]), in a BoVW-based approach for pornography classification. In the same direction, Souza et al. [17] improved Valle et al.'s [18] results by applying ColorSTIP and HueSTIP, color-aware versions of the STIP detector and descriptor, respectively. Both works established a single bag for the entire target video, instead of keeping a bag for each video frame, prior to voting schemes.

Employing or not the BoVW model, other works relied on the motion vectors intrinsically encoded by MPEG compression [23,39–41]. Particularly, Jansohn et al. [23] proposed a BoVW

[1] In videographic jargon, live action refers to the motion pictures that do not depict animated cartoons, but "real" actors.

approach to describe only the static visual features, but they did not apply it to the motion-aware data.

In addition to those scientific results, there is commercial software that blocks web sites with pornographic content (e.g., K9 Web Protection, CyberPatrol, NetNanny). Additionally, there are products that scan a computer for pornographic content (e.g., MediaDetective [5], Snitch Plus [6], PornSeer Pro [7], NuDetective [8]). MediaDetective [5] and Snitch Plus [6] are off-the-shelf products, that rely on the detection of human skin to find potential pictures or movies containing nude people. Similarly, PornSeer Pro [7] is a free pornography-classification system that relies on the identification of specific features (e.g., nipple, breast, lips, eyes) on individual video frames. The work of Polastro and Eleuterio [8] (a.k.a., NuDetective) also adopts skin detection, and it is intended for the Federal Police of Brazil, in forensic activities.

Therefore, to the best of our knowledge, few scientific works and no commercial solutions took advantage from the space–time nature of video for tackling pornography detection.

3. Proposed framework

Web filters and scan-based software play an important role in preventing pornography from reaching unintended or inappropriate audiences, in particular, underage viewers. However, as stated in Section 2, the vast majority of those solutions and a great number of the published methods are based on human skin detection in images. Besides suffering from high rates of false positives, there are plenty of pornographic materials where very little skin is exposed, (from pornographic cartoons, e.g., hentai, to fetishist activities where the participants are almost fully clothed).

As discussed in the previous section, pornography is a much more complex notion than nudity. Pornography is a high-level semantic category, whose translation to visual characteristics is not straightforward.

To cope with that complexity, we design a BoVW-based framework to perform video-pornography classification. We expect to avoid the inherent drawbacks of skin detectors, as well as to reduce the semantic gap between the low-level visual data representation (e.g., pixels), and the high-level concept of pornography. Moreover, while most systems represent videos with keyframes and then apply well-known techniques for static images, we investigate temporal information as a discriminative clue for pornography classification.

Fig. 1 depicts the typical BoVW framework, which can have its operation properly framed in a three-layered representation. Within it, the (i) Low-Level Feature Extraction layer refers to the video description, a process that commonly employs local descriptors to extract perceptual features directly from the pixel values (Steps A:1 and B:1). One level up, the (ii) Mid-Level Feature Extraction layer aims at combining the low-level features into global video representations, with intermediate complexity. This is done through two steps: coding and pooling (Steps A:3 and B:2), with the support of a visual codebook, which is a summarization of the low-level feature space. The coding step quantifies each low-level description with respect to its similarity to the words that compose the visual codebook. The pooling step, in turn, aggregates the quantization obtained in the coding stage, by registering, usually in a single feature vector per video frame, how often the visual words are being manifested. And on top of that, the (iii) High-Level Classification layer deals with the challenge of learning (Step A:4) and predicting (Step B:3) the classes of the mid-level features.

As one might observe in Fig. 1, the existence of a visual codebook, and a supervised learning classification model, implies that every system constructed under the guidance of such framework can operate in two modes: offline (darker horizontal box) and online (lighter horizontal box). Firstly, in the offline operation, after the extraction of the low-level local descriptors (Step A:1), the visual codebook is obtained by unsupervised learning over a sample of local descriptors from the training data (Step A:2). The following coding and pooling stages (Step A:3) return labeled mid-level video representations that are treated as input vectors for a machine learning algorithm, which builds the classification model (Step A:4). Secondly, in the online operation, arbitrary videos are presented to the system; in this case, the system must determine the video labels (a.k.a., test phase), based on the codebook and classification model that were formerly learned.

In the following sections, we delve into the alternatives and decisions we have made, for each one of the mentioned framework levels, regarding the incorporation of temporal information. Fig. 2 depicts the proposed pipeline in depth, with details of the three levels and two operation modes that are inherited from Fig. 1.

3.1. Low-Level Feature Extraction: time-aware local video extraction

First of all, for the sake of efficiency — and similar to Akata el al. [42] — we resize the video frame resolution to *fr* pixels, if larger, keeping the original aspect ratio. That is related to Steps A:1 and B:1, in Fig. 2, and considerably reduces the amount of data to be analyzed.

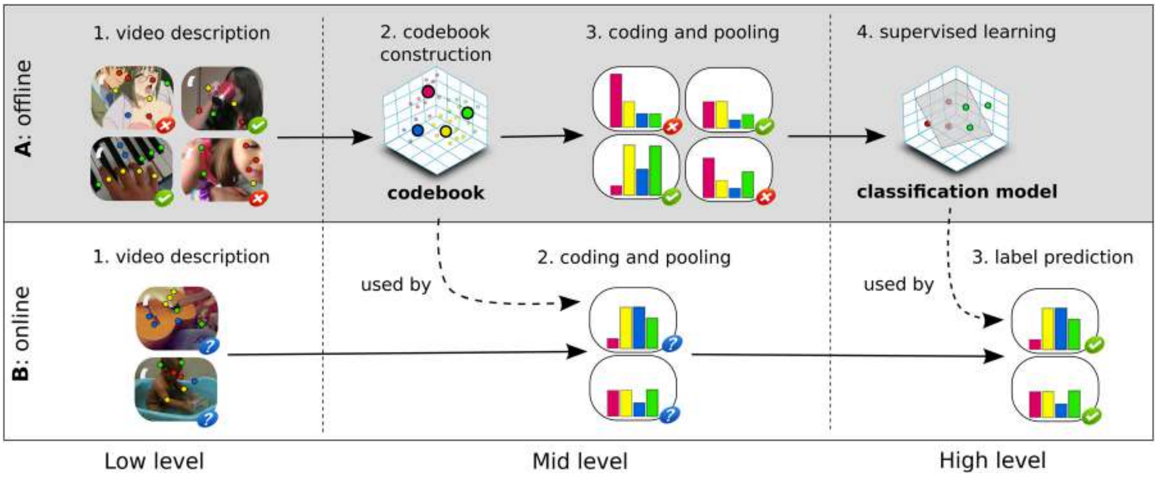


Fig. 1. The typical three-layered BoVW-based framework. On top, the darker box depicts the "offline" or "training" phase, where we have access to labeled videos. On bottom, the lighter box depicts the online phase, where the previously learned model (codebook and classification model) is used to predict the label of new videos. Notice that, in this example, the video labels are employed/predicted only in the later, rightmost, stage.

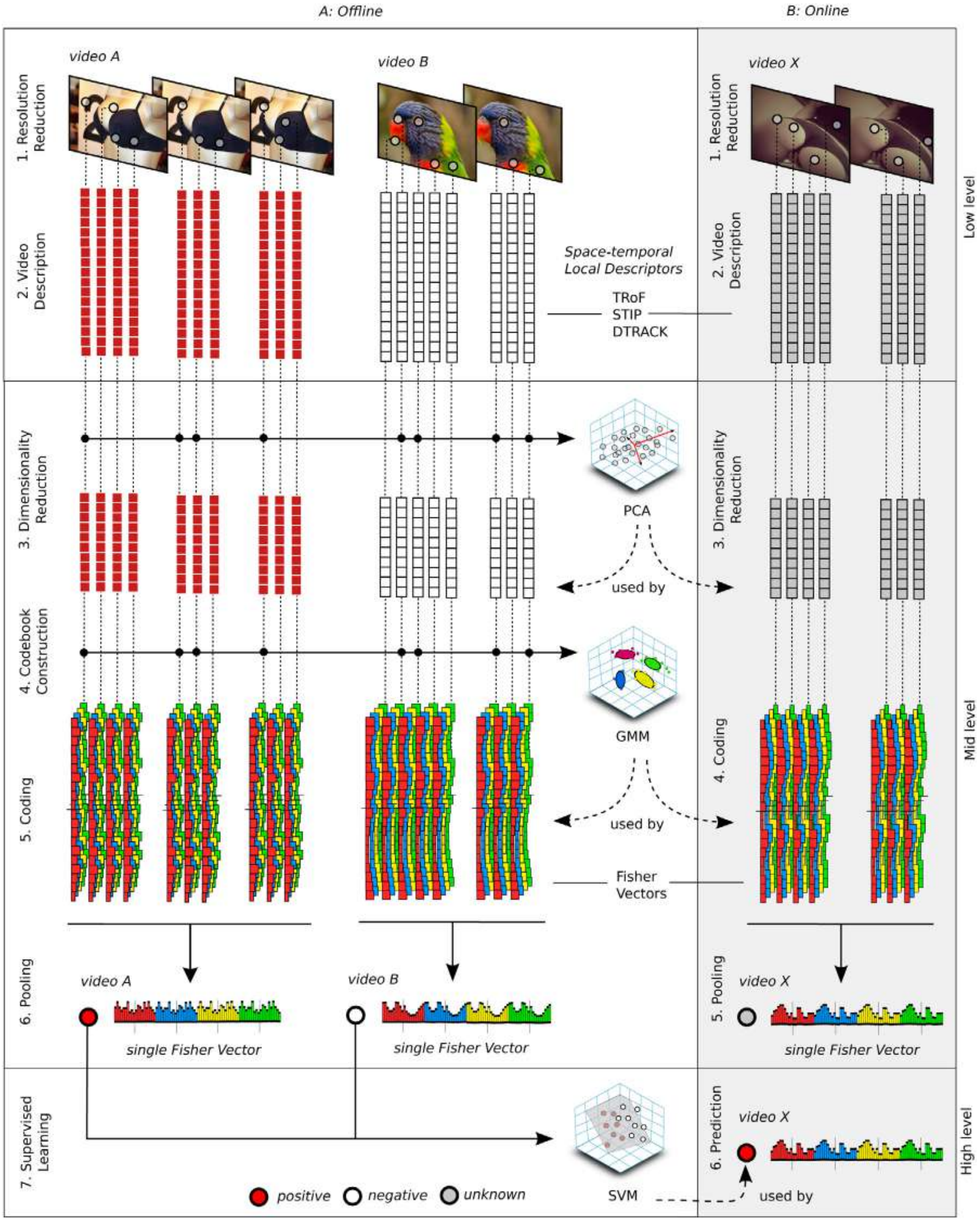


Fig. 2. Three-level pipeline for efficient sensitive-video classification. On the left, the larger column depicts the offline pipeline execution, in which video labels are known in advance, and are used for calculating the principal component analysis (PCA) transformation matrix (in Step A:3), generating the GMM codebook (in Step A:4), and training the SVM classification model (in Step A:7). On the right, the darker column depicts the online execution, in which the formerly learned models are used by the system, for predicting the class of arbitrary videos. This pipeline incorporates temporal information in the low and mid levels, by means of (i) local space-temporal descriptors (Steps A:2 and B:2), and (ii) entire-footage mid-level feature pooling (Steps A:6, and B:5), respectively.

The next step is related to the task of video description (Steps A:2 and B:2, in Fig. 2). At this point, each video frame corresponds to a collection of pixels, that have no semantic information by themselves. Thus, the inherent challenge is to extract useful information from such numbers.

Concerning that challenge, Tuytelaars and Mikolajczyk [43] early attested the success of the employment of local descriptors to the development of computer vision systems. One can find in the literature many types of local descriptors, with Scale-Invariant Feature Transform (SIFT) [44] and Speeded-Up Robust Features (SURF) [36] being probably the most referenced ones. These descriptors differ mostly in the type of visual phenomenon they rely on to extract features, and in the engineered methods to combine these features.

Keeping in mind the video nature as a sequence of frames in time, the conventional descriptors rely on the space domain of the frames, thus analyzing pixel values strictly in the frame they occur. Such descriptors can be considered *static*, in the sense that they consider neither the video time dimension nor the order in which the frames occur within the video. SIFT and SURF are examples of these descriptors: they describe the content of the frames, but they do not say a thing about how that content changes along with the video.

In contrast with the static features, there are descriptors that analyze the frame pixels as voxels. In this scenario, pixel values are analyzed considering their spatial information in the frame, and also their position in the video time. Such descriptors can be considered *time-aware*, in the sense that the features they deliver somehow encode the spatial-temporal information inherent to a video stream. For instance, Space–Time Interest Points (STIP) [38] and Dense Trajectories [31] are representatives of such descriptors.

Nevertheless, as a result of the addition of the third dimension in the description process, more data becomes available to be analyzed. Thus, if the space-temporal data is not used carefully, it leads to a higher computational cost, both in terms of processing time and memory footprint. Therefore, we suggest the use of varied space-temporal strategies of video description, ranging from the emblematic solution of STIP (the first space-temporal descriptor proposed in the literature), to the state-of-the-art strategy of Dense Trajectories, and to TRoF approach, a novel time-aware local descriptor that saves computational resources, yet maintaining reasonable video-description capability. With that, we aim at pushing temporal information early on the low-level stages of the framework pipeline. In Section 4, we present TRoF in details.

3.2. Mid-Level Feature Extraction: a single and richer video representation

In the mid-level phase, the main goal is to transform the previously extracted local descriptions into a global and richer video representation.

Firstly, for the sake of reducing memory footprint, and eventually providing better system classification accuracy [45], we reduce the d_f-dimensional low-level feature vectors to $p_f \leq d_f$ dimensions, with principal component analysis (PCA). That is related to Steps A:3 and B:3, in the pipeline. More specifically, regarding Step A:3 — in the particular case of offline operation (where performance is not a major concern) — we obtain the eigenvectors and the eigenvalues of the covariance matrix that is calculated over a random sampling of the low-level training feature space, for further online use. Notwithstanding, in order to provide a more content-aware strategy, we randomly select k_p low-level descriptions, with half of them coming from the positive training samples, and the other half coming from the negative ones.

As it follows, the pipeline can be broken into two steps [46]: *coding* (Steps A:5 and B:4) and *pooling* (Steps A:6 and B:5). The coding step quantifies the low-level local descriptors according to a visual codebook of k visual words, which is usually built by clustering a set of local descriptors (e.g., k-means clustering algorithm [47]). The pooling step, in turn, aggregates the codes obtained into a single feature vector.

In the Bag of Visual Words (BoVW) [48], the most popular mid-level image representation, the coding step associates the local descriptors to the closest element in the codebook (called hard-assignment coding), and the pooling takes the average of those codes (called average pooling). Both steps of coding and pooling have been subject of important improvements over the years, since the BoVW representation has important limitations. Comparisons of coding and pooling strategies can be found in [46,49].

Perronnin et al. [50] introduced the best mid-level representation currently available, the Fisher Vector. It extends upon the BoVW method by encoding the average first- and second-order differences between the local descriptors and the elements of the codebook. Therefore, as we wanted to benefit from breakthrough mid-level representations, we realized that, to the best of our knowledge, Fisher Vector has never been applied to the pornography classification problem. Hence, to the mid-level stage of the proposed framework, we extract Fisher Vectors, using a Gaussian Mixture Model (GMM) to obtain the visual codebook (see Step A:4, in Fig. 2), as suggested in [50].

Notwithstanding, we notice that the pooling step offers an interesting chance to incorporate temporal information, in the mid-level stage, for static local descriptors. Thus, instead of pooling the codes per video frame, we can pool them all together along the time dimension. As a result, it becomes possible to gather a single feature vector for a video sequence (i.e., a single Fisher Vector). By working with this reduced representation, we believe that the classification performance can often yield more accurate results, while computational costs may also be significantly reduced. We suggest to follow this strategy in the proposed framework.

3.3. High-Level Video Classification

In the high-level phase, a supervised classification algorithm induces a prediction function using the obtained mid-level vectors. The learned function tries to predict the correct label associated with any new observation. Thus, in offline operation (or training stage), a classifier is trained on the mid-level vectors (see Fig. 2, Step A:7). Once the classification model is learned, it can be used to predict the label of a new (test) instance. Step B:6 illustrates the online operation.

Many machine learning algorithms can be used in this last step. In the BoVW literature, nonlinear Support Vector Machines (SVM) are the most widely used technique. We use a linear SVM, since it is well known that nonlinear kernels do not improve classification performances for Fisher Vector representation (see [50]).

4. Temporal Robust Features (TRoF)

Local space-temporal features are a successful representation for action recognition [38,31]. Nevertheless, one important factor deterring the consideration of these features for real-time applications is the high computational cost, regarding both memory footprint and computational time.

To solve this problem, we propose a fast space-temporal video approach, with low-memory footprint, which can be performed on limited hardware, such as mobile devices. To deal with the memory usage issue, we introduce a sparse strategy, which detects an optimized amount of space-temporal interest points, while maintaining high accuracy to the pornography classification task. For that, we investigated what kind of clues we could observe in a video, and we singled out the motion information.

Therefore, (i) we custom-tailor a detector for finding motion in videos, and (ii) we design a novel space-temporal interest point descriptor to represent such motion, leading to what we call Temporal Robust Features (TRoF). In the following, we give more details about TRoF. Section 4.1 introduces the TRoF detection method, while Section 4.2 explains its description approach. For a reader mostly interested in the forensics takeaways, Sections 4.1 and 4.2 can be skipped without losing the essential idea of the paper.

4.1. TRoF detector

The TRoF detector is directly inspired by the still-image Speeded-Up Robust Features (SURF) detector [36], which is very fast. It relies on three major extensions of the original method, to use the video space–time: the employment of five-variable Hessian matrices, three-dimensional box filters, and the concept of integral video. In the following, we explain each one of these expansions.

4.1.1. Five-variable Hessian matrix

The original SURF detector [36] identifies interesting visual local structures (a.k.a. blobs) in an image, by relying on the determinants of Hessian matrices, that are calculated at different locations onto the image surface, with varied scales.

Every Hessian matrix $H(x, y, \sigma)$ is a function of the location $\mathbf{x}(x, y)$ and the scale σ. As pointed out by Bay et al. [36], the Hessian matrices with the highest determinants are the ones that share a location $\mathbf{x}(x, y)$ and present a scale σ that fits well to the size of an occurring blob. Hence, the selection of the location and the scale of interesting blobs are done by taking the candidate points and scales whose Hessian determinants are above a given threshold.

To find the candidate locations, the best effort must look at every pixel of the image. To tackle different scales, Bay et al. [36] suggest dividing the scale space into a list of octaves. Each octave encompasses a scaling factor that is half the scaling factor of the next octave, and they are subdivided into a constant number of four inner scale layers. Given that various Hessian matrices with different scales are calculated at a same candidate location, a non-maximum suppression is applied both spatially and over the neighboring scales, to select those with the highest determinants. Each selected Hessian thus leads to a detected blob.

Willems et al. [51] propose a straightforward extension of such mechanism to the case of video, by adding the time dimension to the Hessian matrices, and using separated scales for space (σ_s) and for time (σ_t), i.e., the original $H(x, y, \sigma)$ becomes $H(x, y, t, \sigma_s, \sigma_t)$. With that, they expect the Hessian matrices with the highest determinants to coincide with interesting space-temporal phenomena, within the video space–time.

Eq. (1) depicts the temporal extension to the original Hessian matrices. Within it, $L_{xx}(x, y, t, \sigma_s, \sigma_t)$ is the convolution of the Gaussian second order derivative $\partial^2 G(x, y, t, \sigma_s, \sigma_t)/\partial xx$ with the voxel $\mathbf{x}(x, y, t)$ of the target video. Similarly, $L_{xy}(x, y, t, \sigma_s, \sigma_t)$ refers to the convolution of $\partial^2 G(x, y, t, \sigma_s, \sigma_t)/\partial xy$ with the voxel $\mathbf{x}(x, y, t)$, and so on for L_{xt}, L_{yt}, L_{yy}, and L_{tt}.

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

As one might observe, the extension relies on five-variable Hessian matrices $H(x, y, t, \sigma_s, \sigma_t)$, where x is related to the video-frame width, y is related to the video-frame height, t is related to the video duration, and σ_s and σ_t are respectively the standard deviations (a.k.a. scaling factors) for space and for time. Due to the presence of five variables, the amount of calculable Hessian values may be large, depending on the video resolution, quantity of frames, and number of considered scales while inspecting the scale search space.

Willems et al. [51] suggest the use of o_s five-layered spatial scale octaves, and o_t five-layered temporal scale octaves, for the task of inspecting the scale search space. Even though they give neither clues on the actual values used for the candidate standard deviations, nor how these values may be combined,[2] we can stipulate that they must compute at most $o_s \times 5 \times o_t \times 5$ Hessian values, for every voxel.

At this point, to perform a fast extraction of interesting space-temporal blobs, yet maintaining a reasonable scale-invariance detection, we reduce the number of calculable Hessian values, by co-variating the spatial and temporal Gaussian standard deviations, while exploring the scale search space. For that, we use o scale octaves (our first detection parameter) with dual nature (spatial and temporal), whose layers are defined by a particular spatial standard deviation, and a particular temporal standard deviation. As a result, it becomes necessary to compute only $o \times 4$ Hessian values, for every candidate voxel.

To provide such scale search space reduction, we extend the four-layered octaves that were settled by Bay et al. [36] — by complementing their layers with temporal standard deviations — and we keep the scale-increasing policies, this time changing spatial and temporal scales simultaneously. For instance, a first space-temporal octave would start with a scale of 9 × 9 × 9 voxels, and it would present an inter-layer increase of six voxels, for both space and for time. The resulting space-temporal octave would thus comprise four scales, with 9 × 9 × 9, 15 × 15 × 15, 21 × 21 × 21, and 27 × 27 × 27 voxels, respectively.

Similar to the still-image case, once all the necessary Hessian values are calculated, a non-maximum suppression strategy must be performed for obtaining only the extreme values within a five-dimensional neighborhood, considering the immediate Hessian neighbors along the x-, y-, t-, σ_s-, and σ_t-axis directions. After the selection of an extremum, we use the variation of the Hessian values that are within the suppression neighborhood, to interpolate the x, y, t, σ_s, and σ_t values of the detected blob, with sub-voxel accuracy.

Finally, as it is impractical to consider every voxel of the video space–time as a candidate — for every scale combination — we propose to use two detection parameters s_s and s_t, which define the initial sampling steps in spatial and temporal directions, respectively, for selecting the points where to calculate the Hessian values. We also recommend to double these steps at every new octave, due to the property of an octave encompassing a scaling factor of two, when compared to the previous one. On the occasion of selecting values for such parameters, one must consider that larger values of s_s and s_t result in a faster detection process, at the cost of reducing the accuracy in the detection of the position and the scale of the interest points.

4.1.2. Three-dimensional box filters

To quickly compute the various Hessian determinants, the original SURF method approximates the inherent two-dimensional Gaussian second-order derivatives by proper box filters, which can be readily convolved to the integral image of the target image.

Fig. 3(a) shows the discretized version of the Gaussian second order derivative $\partial^2 G(x, y, \sigma)/\partial xx$, with $\sigma = 1.2$, projected onto a 9 × 9 image segment, and its correspondent original two-dimensional SURF cropped filter, that constitutes the actual box filter used to support the calculation of the Hessian determinant. Similarly, Fig. 4(a) shows the discretized version of $\partial^2 G(x, y, \sigma)/\partial xy$, and its cropped counterpart.

In the case of TRoF, we have five-variable Hessian matrices (please refer to Eq. (1)), hence the related Gaussian second order derivatives are three-dimensional (spreading across the x, y, and t directions of the video space–time), with spatial scale σ_s (in x and y directions), and temporal scale σ_t (in t direction). We approximate these derivatives by cuboid filters.

Figs. 3(b) and 4(b) show two of the six Gaussian filters, in both discretized and cuboid cropped versions. The other remaining four cuboid filters can be easily deduced by simply applying the proper rotations.

In all elements of Figs. 3 and 4, Gaussian filters are shown as pixel-discretized heat maps, whereby red zones refer to the higher values, in opposition to the blue parts which represent the smaller ones. Yellow and green zones are in the middle, with yellow closer

[2] Executables are no longer available and, due to a lack of details in Willems et al.'s paper, we could not reproduce their method, making direct comparison impossible.

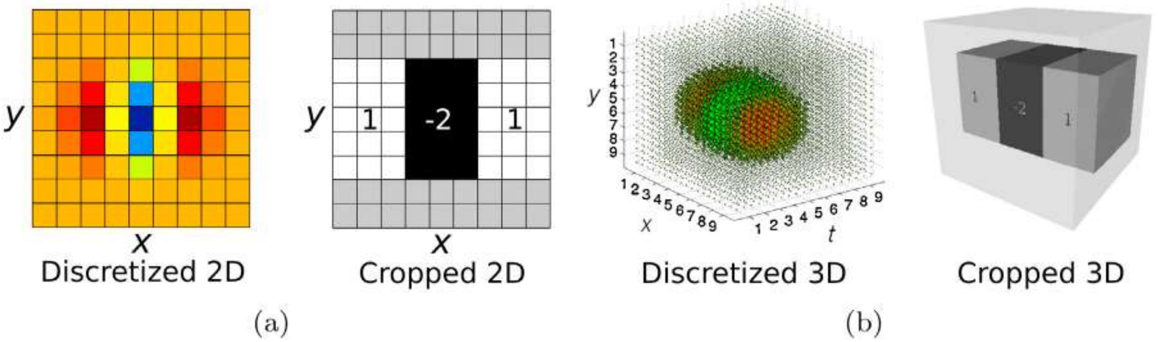


Fig. 3. A visualization of the derivative filters $\partial^2 G(\mathbf{x}, \sigma)/\partial xx$, and their approximations. (a) The original two-dimensional filter, with its discretized and cropped versions. (b) The respective three-dimensional versions. The rightmost cuboid filter is one of the six filters used by TRoF detector to support the calculation of Hessian matrices.

to red, and green closer to blue. Cropped box filters, in turn, are approximations, with values explicitly shown on the images. As adopted in [36], gray positions have zero value, while white areas are positive, and black are negative.

4.1.3. Integral video

The original SURF detector relies on integral images to quickly perform image convolutions. In the case of TRoF, that operates within the video space–time, we must extend the concept of integral image to the idea of integral video, by considering three dimensions rather than two.

Eq. (2) states the value of an integral video $V_{\Sigma}(\mathbf{x})$ at a space-temporal location $\mathbf{x}(x, y, t)$. It is thus given by the sum of all pixel values belonging to the video V, that rely on a rectangular cuboid region formed by $\mathbf{x}$ and the video origin.

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

Once the integral video is computed, it only takes eight accesses and seven operations to calculate the sum of the pixel values inside any rectangular cuboid region, independently of its size. For instance, the value V of the volume that is represented in gray in Fig. 5 is given by Eq. (3).

$$V = (A + C) - (B + D) - (A' + C') + (B' + D'). \tag{3}$$

With the integral video technique, we can convolve box filters of any scale with the video space–time, in constant time. Nevertheless, one implementation issue remains, regarding the calculation of the integral video. For streams with long duration and high resolution, the sum of pixel 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 c frames (our fourth detection parameter). A smaller c results in a smaller amount of memory needed to store an integral video. However, if it considers only a few video frames, it may segment the motion information and, therefore, destroy it with a higher probability.

Finally, given that video streams may be very assorted — especially in terms of camera quality, camera position, and illumination conditions — we cannot find a single Hessian threshold to discard irrelevant blobs, that works for all the cases. Thus, to proceed in a less ad-hoc direction, we select the b most relevant blobs within each integral video, after sorting the candidate interest points according to their Hessian values. Hence, we do not need a threshold to identify relevant space–time phenomena, we just take the b strongest ones (our fifth and last parameter), which are the ones with the b highest Hessian determinants.

4.2. TRoF descriptor

The former detection step delivers interest points within the video space–time, that are individually characterized by a three-dimensional position $P(x, y, t)$, plus a spatial scale σ_s, and a temporal scale σ_t. The next step regards a description process to represent these elements.

At this point, we aim at performing an efficient and effective time-aware description of the previously detected space-temporal TRoF blobs, with low-memory footprint. Regarding efficiency, we take for description only a small amount of the blob voxels, yet considering their space-temporal disposition. For that, we describe only the voxels that are projected onto three orthogonal planes of interest: the blob-centralized spatial [x, y]-plane, and the

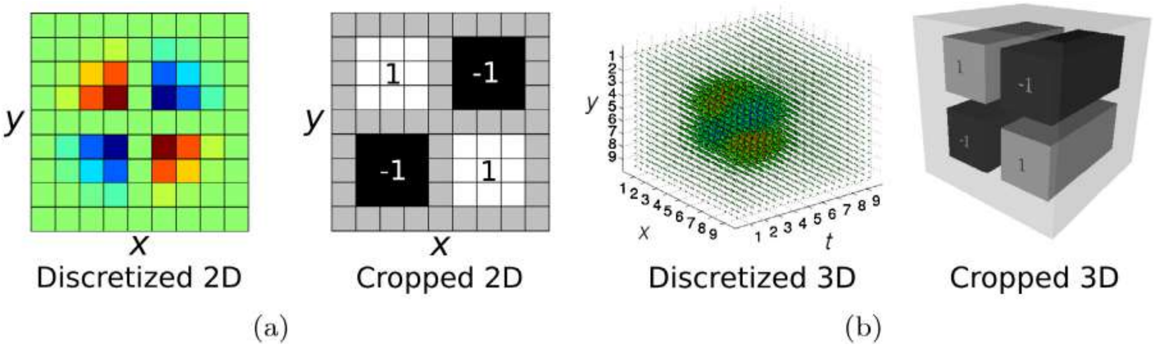


Fig. 4. A visualization of the derivative filters $\partial^2 G(\mathbf{x}, \sigma)/\partial xy$, and their approximations. (a) The original two-dimensional filter, with its discretized and cropped versions. (b) The respective three-dimensional versions. The rightmost cuboid filter is one of the six filters used by TRoF detector to support the calculation of Hessian matrices.

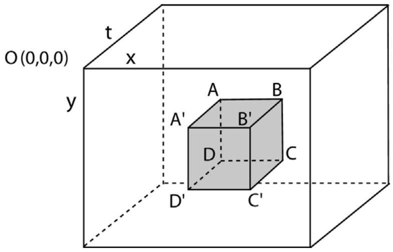


Fig. 5. Integral video representation. The outer box represents the video space–time, with the x-axis associated to the width, the y-axis to the height, and the t-axis to the video duration. The inner gray box represents the cuboid region, which is calculated by Eq. (3).

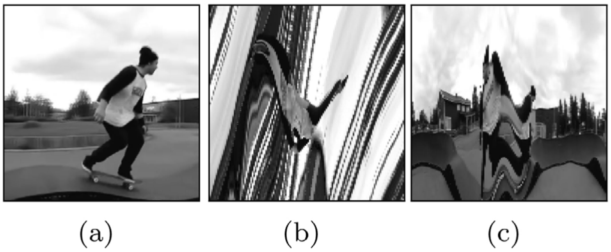


Fig. 7. Visual representation of the voxels described in a sample TRoF blob. (a) Voxels described in the purely spatial [xy]-plane. (b) Voxels described in the space-temporal [xt]-plane. (c) Voxels described in the space-temporal [yt]-plane. All three images are individually described with a SURF descriptor [36].

blob-centralized temporal [x, t]- and [y, t]-planes. Regarding effectiveness, we suggest the use of SURF [36] descriptor to properly capture the variation of the values of the blob voxels, but other effective image descriptors (e.g., Histograms of Oriented Gradients — HOG [52]) can alternatively be used.

Fig. 6(a)–(c) depicts each one of the three flat SURF blobs, in the form of solid gray circles, that we propose to describe within a target TRoF blob. Fig. 6(d) depicts the structural union of these SURF blobs. The resulting structure is inscribed inside a space-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 space-temporal scale s, which is equal to the smallest value between σ_s and σ_t, to perfectly fit the inherently symmetric SURF blobs [36], without leftovers.

For the sake of illustration, Fig. 7(a)–(c) depicts the visual content of the voxels that are described in each one of the three orthogonal planes of an eventually detected TRoF blob. Fig. 7(a) contains the voxels belonging to the [xy]-plane, which — by being purely spatial — is the only one that is visually intelligible to humans. Fig. 7(b), in turn, contains the voxels belonging to the [xt]-plane, while Fig. 7(c) contains the voxels described in the [yt]-plane. We consider only these three images for applying a SURF descriptor [36], or other ones (e.g., HOG [52]).

With the intent to register eventual correlations among the three 64 dimensional SURF blobs, that could be helpful to distinguish porn and non-porn material, we generate the final TRoF feature vector by concatenating the three 64-dimensional blob descriptions, in the following order: [x, y]-, [x, t]-, and [y, t]-plane. Thereby, as a practical result, the SURF-based TRoF

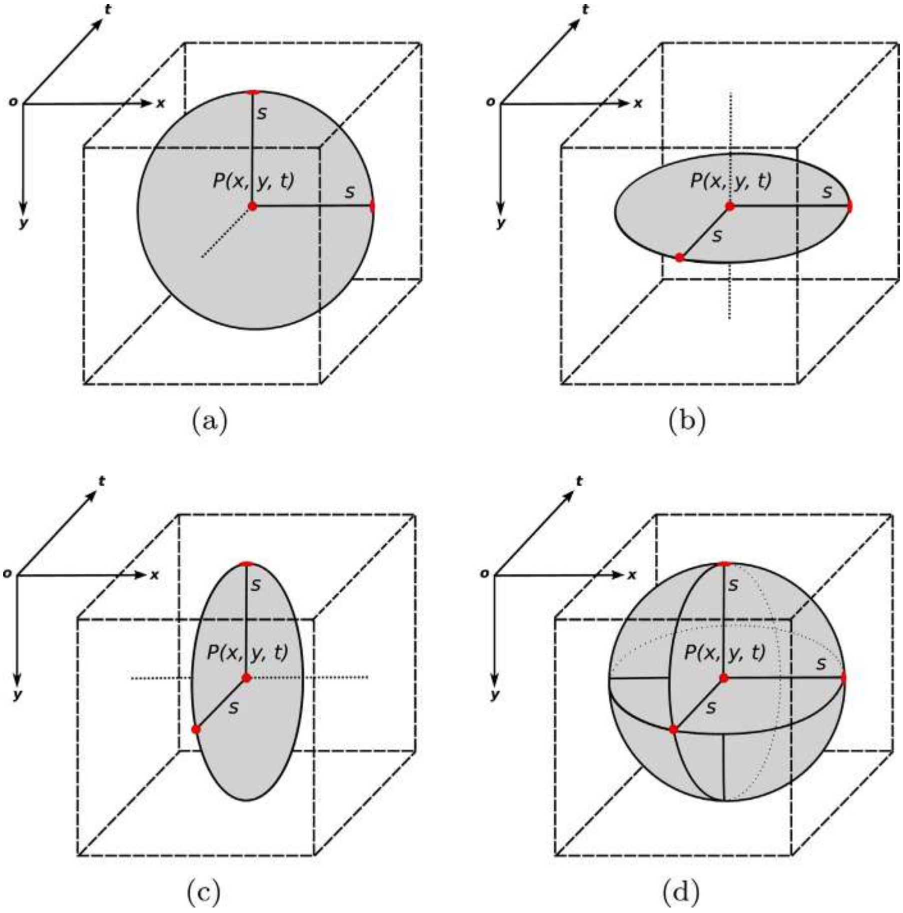


Fig. 6. TRoF described union structure. The gray circles are conventional SURF blobs, which are all centered at position $P(x, y, t)$, and present scale $s = min(\sigma_s, \sigma_t)$. P, σ_s, and σ_t come from a formerly detected interest point. (a) SURF blob that is projected onto the [x, y] plane. (b) SURF blob that is projected onto the [x, t] plane. (c) SURF blob that is projected onto the [y, t] plane. (d) Resulting space-temporal structure, which is formed by the union of the three SURF blobs.

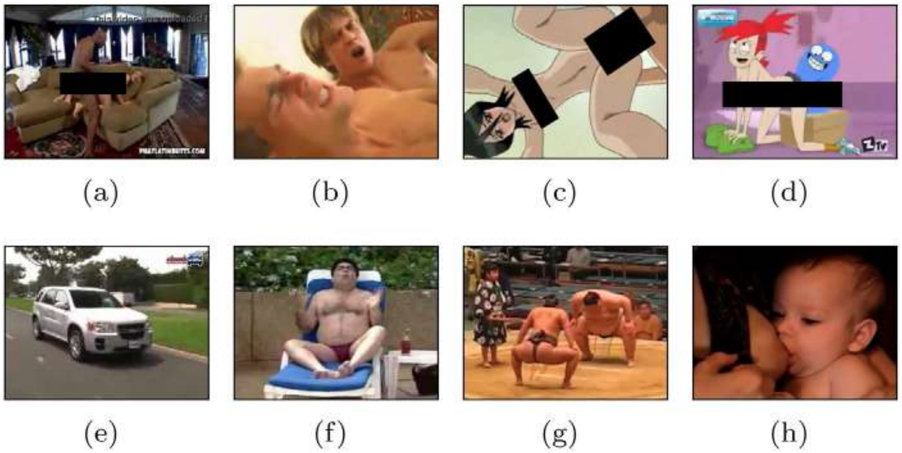


Fig. 8. Frames sampled from the Pornography-2k dataset. On the top row, we show representative sensitive content, including pornographic cartoons. The black censor bars were added by us in the understanding that this paper can reach a broad audience, including underage readers — they are not present in the original material. On the bottom row, we show non-pornographic content, emphasizing examples with non-sexual skin exposure. We expect skin-detector-based solutions to succeed in labeling samples (a–b), (e), but to fail in labeling samples (c–d), (f–h). In (c–d), we do not have real actors' skin, despite of having pornographic material. In (f–h), we have non-pornographic cases with a lot of body exposure.

descriptor outputs a set of 192-dimensional feature vectors, for every target video stream.

5. Experimental setup

This section describes the experimental setup, including the parametric values used for each approach. First, it is worth mentioning that previous work in the pornography classification literature presented a limited validation, with no standard datasets or metrics, except for the published methods in [20,19,21,17,18], which used the Pornography-800 dataset [21] with 800 videos. Hence, aiming at providing a standard validation benchmark, we augmented that dataset to 2000 videos, leading to 99.5 h of pornographic content, and 40.5 h of non-pornographic material. Section 5.1 introduces the new pornography dataset. In Section 5.2, we present the metrics we use to evaluate the experimental results. Next, Section 5.3 presents third-party solutions and discusses how they differ among themselves. Finally, Section 5.4 details the experimental setup of the BoVW-based solutions.

5.1. Pornography-2k dataset

The Pornography-2k dataset is an extended version of the Pornography-800 dataset, originally proposed in [21]. The new dataset, introduced here, comprises nearly 140 h of 1000 pornographic and 1000 non-pornographic videos, which varies from 6 s to 33 min.

Concerning the pornographic material, unlike Pornography-800 [22], we did not restrict to pornography-specialized websites. Instead, we also explored general-public purpose video networks,[3] in which it was surprisingly easy to find pornographic content. As a result, the new Pornography-2k dataset is very assorted, including both professional and amateur content. Moreover, it depicts several genres of pornography, from cartoon to live action, with diverse behavior and ethnicity.

With respect to non-pornographic content, we proceeded similarly to Avila et al. [22]. We collected *easy* samples, by randomly selecting files from the same general-purpose video networks. Also, we collected *difficult* samples, by selecting the result of textual queries containing words such as "wrestling", "sumo", "swimming", "beach", etc. (i.e., words associated to skin exposure). Fig. 8 depicts some example frames from the Pornography-2k dataset.

The Pornography-2k dataset is available free of charge to the scientific community but, due to the potential legal liabilities of distributing large quantities of pornographic/copyrighted material, the request must be formal and a responsibility term must be signed.

To evaluate the results of our experiments, we apply a 5 × 2-fold cross-validation protocol [53]. It consists of randomly splitting the Pornography-2k dataset five times into two folds, balanced by class. In each time, training and testing sets are switched and consequently 10 analyses for every model employed are conducted.

5.2. Pornography classification metrics

To assess the performance of the pornography classifiers, we report the *normalized accuracy* (ACC), and the F_2 *measure* (F_2), both averaged in all experimental folds.

ACC tells us the hit rate of the method, regardless of the class labels. Mathematically, this can be expressed as:

$$ACC = \frac{TPR + TNR}{2} \tag{4}$$

where TPR is the true positive rate and TNR is the true negative rate.

F_2, in turn, is the weighted harmonic mean of precision and recall, which gives twice the weight to recall ($\beta = 2$) than to precision. *Recall* measures the fraction of actual positive examples that are predicted as positive (i.e., the number of true positives divided by the sum of true positives and false negatives), while *precision* measures the fraction of actual positives among those examples that are predicted as positive (i.e., the number of true positives divided by the sum of true positives and false positives). High recall means a low number of false negatives, and high precision means a low number of false positives. For example, suppose we have 20 videos to classify (10 pornographic and 10 non-pornographic). In addition, suppose that a classifier correctly classifies eight of the pornographic videos and seven of the non-pornographic ones. Then, in this case, precision is 73%, recall is 80% and F_2 measure (which gives twice the weight to recall than to precision) is 78%.

In the case of pornography filtering, the F_2 measure is crucial because false negative results are harmful, allowing us to be exposed to pornographic content. It is thus less prejudicial to

[3] YouTube (http://www.youtube.com), Vimeo (vimeo.com) and Vine (vine.co)

wrongly deny the access to non-pornographic material, than to wrongly disclose pornographic content. F_β measure is defined as:

$$F_\beta = (1+\beta^2)\times\frac{precision\times recall}{\beta^2\times precision + recall} \tag{5}$$

where β is a parameter denoting the importance of recall compared to precision. We apply $\beta = 2$, which means that recall is twice as important as the precision.

5.3. Third-party software solutions

Despite finding a myriad of pornographic-content filters available on the Internet, only a few solutions rely on visual data to classify pornographic content, and very few of them are able to inspect video content. We thus selected the most recent ones to evaluate their classification performance in detecting unsuitable material: MediaDetective [5], Snitch Plus [6], PornSeer Pro [7], and NuDetective [8].

MediaDetective and Snitch Plus are both off-the-shelf commercially available programs.[4] NuDetective, on its turn, is not available to the general public, but can be acquired by law enforcement agencies or for research purposes, with no costs. Finally, Porn Seer Pro is freely available.

All of these systems rely on content-based analysis of images/videos. Nevertheless, while MediaDetective, Snitch Plus and NuDetective apply skin-based detectors to identify pornographic content, PornSeer Pro is based on the detection of specific features (e.g., breasts, genitalia, anuses).

Furthermore, for MediaDetective and Snitch Plus, the video files are rated according to their potential (i.e., probability) for pornography. In those cases, we tag a video as pornographic if its probability is equal or greater than 50%. NuDetective and PornSeer Pro, on the other hand, assigns binary labels to the video: positive (i.e., the video is pornographic) or negative (i.e., the video is non-pornographic).

Finally, MediaDetective and Snitch Plus have four predefined execution modes, which differ mostly on the rigorousness of the skin detector. In our experiments, we opted for the most rigorous execution mode. Regarding NuDetective and PornSeer Pro, we employed their default settings.

5.4. BoVW-based approaches

The proposed framework, introduced in Section 3, is evaluated through different techniques. Specifically, we explore various methods of low-level local video description. In this section, we first describe the experimental setup and we next provide a brief description of the local descriptors we apply to our experiments.

Similar to Akata et al. [42], we first pre-process the dataset by resizing the video frame resolution to an area of up 100 thousand pixels. Since we are keeping the original aspect ratio, the area may not be exactly 100 thousand pixels. For example, a 520 × 390-pixel video frame (i.e. 202,800 pixels) can be resize to 364 × 274 pixels (i.e., 99,736 pixels), as we maintain the aspect ratio. By working with this resized video, computational costs are significantly reduced, while the classification accuracy is not affected. Moreover, regardless of the low-level descriptors, we apply principal component analysis (PCA) to reduce by half their dimensionality. These descriptors are then aggregated into an image/video-level signature.

In order to make the comparisons fair, we use the same mid-level representation for all techniques evaluated. Therefore, we extract Fisher Vectors [50], one of the best mid-level representations [45,54]. Also, the descriptor distribution is modeled using a GMM, whose parameters are trained over 10 million randomly sampled descriptors (half of the descriptors sampled from positive videos, and half from negative ones in the training set), using an expectation maximization algorithm. By default, we use 256 Gaussians, as suggested in [50].

Classification is performed by Support Vector Machines (SVM) classifiers using LIBLINEAR library [55]. We apply grid search to find the best C SVM parameter during training.

5.4.1. Speeded-Up Robust Features (SURF)

To provide a controlled baseline for the space-temporal techniques, we extract SURF descriptors [36], which operate over static images only.

Thereby, for the sake of processing time, we use the I-frames[5] from the video footage. Next, we discard 10% of the image borders to remove possible watermarks. Our preliminary experimental results showed that this operation has no significant effect on the error rates. SURF descriptors are then extracted on a dense 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.

In the classification phase, the classifier opinion is asked for each individual frame, and the final decision is reached by majority voting (SURF-MJV, baseline). It means that temporal information is incorporated at the high-level step only.

Alternatively, we also propose adding temporal information at the mid-level step (SURF-MLP), by pooling the mid-level features over the entire video.

5.4.2. Space–Time Interest Points (STIP)

It was the first local descriptor designed for analyzing the video space–time.

Roughly speaking, the STIP detector [30] is an extension of the Harris corner detector, which adds a third dimension — the time — to the equations. The STIP descriptor relies on Histograms of Oriented Gradients and Histograms of Optical Flow (a.k.a., HOG-HOF descriptions), that are computed from three-dimensional video patches, distributed along the neighborhood of the detected interest points.

For the experiments, we extract both sparse — i.e., 3D-Harris-detected (STIP) — and dense STIP (DSTIP) descriptors, with the code provided by Laptev [30], using default values.

5.4.3. Dense Trajectories (DTRACK)

It represents the current state of the art in the field of time-aware local descriptors.

In general terms, the Dense Trajectories [31] describe movement by the means of coding the trajectories of interest points. It samples the interest points on a regular grid in each video frame, and tracks them using an improved optical flow algorithm. Therefore, it describes such trajectories by the application of HOG-HOF descriptors, combined with Motion Boundary Histograms.

To extract the Dense Trajectories from the video files, we use the code provided by Wang et al. [31], with default values. It is noteworthy to mention that, to the best of our knowledge, Dense Trajectories have never been applied to the task of pornography classification.

5.4.4. Temporal Robust Features (TRoF)

Similar to STIP, we extract both sparse — i.e., Hessian-detected (TRoF) — and dense TRoF (DTRoF) descriptors, with the support of

[4] We have purchased MediaDetective v3.1, and Snitch Plus v3.1.

[5] An *I-frame*, or intra frame, is a self-contained frame that can be independently decoded without any reference to other images (frames). For more details, please refer to Ozer [56].

Table 2
Space-temporal octaves used in the experimental setup. We use the same values for spatial and temporal scales. Values are measured in pixels.

Octave	Scales			
1	$9 \times 9 \times 9$	$15 \times 15 \times 15$	$21 \times 21 \times 21$	$27 \times 27 \times 27$
2	$15 \times 15 \times 15$	$27 \times 27 \times 27$	$39 \times 39 \times 39$	$51 \times 51 \times 51$
3	$27 \times 27 \times 27$	$51 \times 51 \times 51$	$75 \times 75 \times 75$	$99 \times 99 \times 99$
4	$51 \times 51 \times 51$	$99 \times 99 \times 99$	$147 \times 147 \times 147$	$195 \times 195 \times 195$

SURF descriptors to represent the TRoF blob content (please refer to Section 4.2).

In the sparse case, to apply the TRoF detector and obtain the three-dimensional blobs of interest, we calculate the integral video at every 250 frames of the target video (i.e., $c = 250$). Thereafter, for each obtained integral video, to describe video very fast, we sample one in every four video voxels, in all directions ($s_s = s_t = 4$), and we use the feature of scale search space reduction (please refer to Section 4.1.1), with four space-temporal scale octaves ($o = 4$), to perform the Hessian calculations. Table 2 details the scales of each one of the four space-temporal octaves used in the experiments.

Finally, we extract 3000 blobs at every 250 video integral frames ($b = 3000$). Experimental results showed that 250 frames, 4 voxels, 4 octaves, and 3000 blobs represent a good compromise between effectiveness and efficiency.

In the dense case, we sample the video space–time at a regular grid with three scales. We use cubic patches with sizes of 24, 48, 96 pixels, and step sizes of 8, 16 and 32 pixels, respectively, in all directions.

It should be mentioned that, although we have proposed the TRoF detector, we also consider a dense sampling strategy (DTRoF), in the interest of a more complete comparison.

6. Experiments and validation

This section evaluates the performance of different methods on the Pornography-2k dataset. The results are compared in Table 3. We report the accuracy rate (ACC) and the F_2 measure (F_2), on a 5×2-fold cross-validation protocol. Additionally, we report the true positive (TPR) and true negative (TNR) rates, to give the reader a broader view of the classification results.

As one might observe, the BoVW-based approaches remarkably outperform the third-party solutions. Not surprisingly, the skin-detector-based systems cannot handle the challenging videos (both pornographic and non-pornographic) of the Pornography-2k dataset. The strength of BoVW-based techniques is further accentuated when we compare the baseline BoVW-based approach (SURF-MJV) to the best third-party solution (PornSeer Pro). It provides an error reduction of over 44% and 68% with respect to ACC and F_2, respectively.

Among the BoVW-based solutions, the use of time-aware local video descriptors leads to more effective classifiers. It corroborates the assumption that motion information carries relevant clues regarding the presence of pornography within a video stream, and that being able to incorporate temporal information to the task of video description might help to capture such motion details.

Furthermore, the use of dense video description also leads to more effective classifiers. For instance, the three best solutions (DSTIP, DTRACK, and DTRoF) rely on a dense description of the video space–time.

Despite of the higher effectiveness, space-temporal and dense strategies often lead to inefficient classifiers, specially with respect to the spent processing time and memory footprint. STIP, DSTIP, DTRACK, and the dense application of TRoF will certainly not run on mobile devices and other hardware-limited platforms.

Fig. 9(a) shows the correlation between the accuracy and the computational time that is spent to perform end-to-end classification, for each BoVW-based solution. Given that we needed to conduct these experiments under the same controlled hardware conditions, we have randomly selected 3 h of video footage from the Pornography-2k dataset, to assess the computational time spent for classification. All experiments were conducted on a 64-bit Linux machine, powered by Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00 GHz with 12 cores and 24 GB RAM. Fig. 9(b) correlates the F_2 measure with computational time.

Likewise, Fig. 9(c) shows the correlation between the accuracy and the quantity of descriptors extracted from the entire Pornography-2k dataset, for each BoVW-based solution. Fig. 9(d), in turn, correlates the quantity of descriptors with the F_2 measure.

At this point, one might argue that using less descriptions does not imply the use of a more efficient description process. It happens because the descriptions do not code the same visual phenomena and, as a consequence, they do not have the same size. For instance, the baseline solutions rely on static 64-D SURF points, STIP on 162-D descriptions, DTRACK on 426-D Dense Trajectories, and TRoF on 192-D low-level feature vectors. Thus, using a large amount of a small description may be equivalent to using a small amount of a large one.

Therefore, in order to evaluate the strategies in terms of memory footprint, we also correlate the classification accuracy and the F_2 measure with respect to the total disk space that is spent to store the low-level feature vectors of the entire Pornography-2k dataset. Fig. 9(e) depicts the correlation between accuracy and disk usage, in a lin-log chart, for a better representation. Fig. 9(f) depicts the correlation between F_2 measure and disk usage.

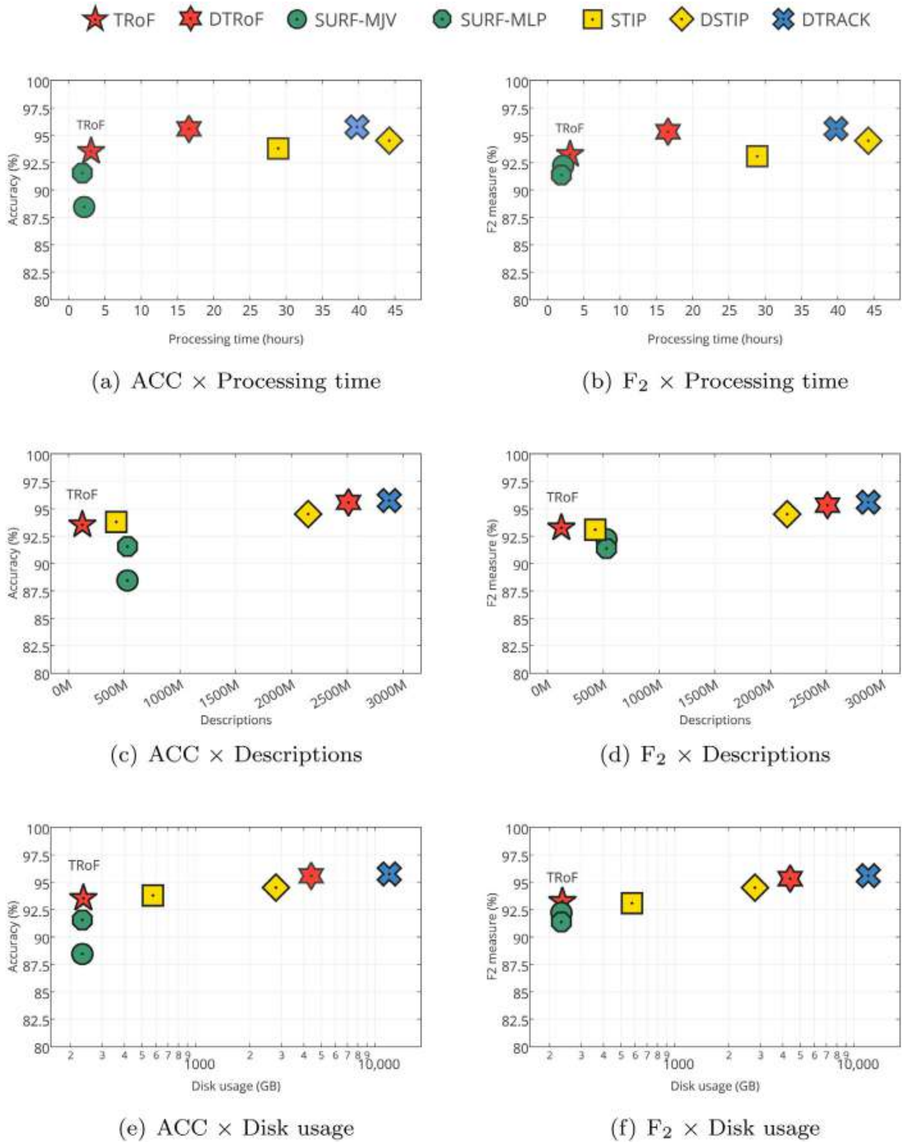
Table 3
Results on the Pornography-2k dataset. We report the average performance on 5×2 folds.

	Solution		TPR (%)	TNR (%)	ACC (%)	F_2 (%)
Third-party	Snitch Plus [6]	Skin	41.86	91.30	66.58	46.35
	MediaDetective [5]		63.30	80.40	71.85	66.54
	NuDetective [8]		59.70	85.50	72.60	62.94
	PornSeer Pro [7]		74.10	84.10	79.10	75.61
BoVW-based	SURF-MJV [36]	Static	94.42	82.48	88.45	92.22
	SURF-MLP [36]		91.26	91.86	91.56	91.37
	STIP [30][a]	Temporal	92.68	94.92	93.80	93.09
	DSTIP [30]		94.50	94.54	94.52	94.51
	DTRACK [31][b]		95.50	96.02	95.76	95.60
	TRoF[a]		93.10	93.98	93.54	93.26
	DTRoF[b]		95.18	95.98	95.58	95.33

TPR: true positive; TNR: true negative rate; ACC: accuracy; F_2: F_2 measure.

[a] TRoF and STIP are not statistically different.

[b] DTRoF and DTRACK are not statistically different.



(a) ACC × Processing time

(b) F_2 × Processing time

(c) ACC × Descriptions

(d) F_2 × Descriptions

(e) ACC × Disk usage

(f) F_2 × Disk usage

Fig. 9. Performance of BoVW-based solutions on the Pornography-2k dataset, putting effectiveness (vertical axes) in perspective with efficiency (horizontal axes). On left, effectiveness regards accuracy, while on right, it regards the F_2 measure. On top row, efficiency regards computational time spent to classify over 3 h of video footage (same system for all methods). On middle row, efficiency concerns the number of descriptors extracted for the entire dataset. On bottom row, efficiency concerns (log scale) disk storage space for the entire dataset. In all charts, the best solutions are at the top-left corner.

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, the sparse application of TRoF — in its Hessian-blobs-detected version — occupies such privileged position.

Fig. 10 details the processing time spent by each BoVW-based solution, by the occasion of performing an end-to-end classification (i.e., online operation only) of the 3 h of randomly chosen video footage. As one might observe, TRoF is the fastest space-temporal descriptor, even in the case of being densely applied (DTRoF).

In the particular case of the sparse TRoF, besides the advantage of counting on a faster descriptor, the proposed detection process allows us to extract a minimum amount of interest points, optimally centered at moving objects. As a consequence, the small quantity of good descriptions directly reduces the processing time that is needed to project data (in PCA), and to perform the calculation of Fisher Vectors. Hence, it presents a video processing rate of almost 30 fps, indicating that it might be suitable for real-time video analysis.

In order to visualize the quality of the TRoF detection process, we have created four synthetic videos[6] that depict moving particles at different scales, performing basic trajectories (e.g., vertical, horizontal, diagonal, etc.), along with static elements. Fig. 11 shows TRoF interest points that were detected along six frames of one of these videos. The white circles, line and stars correspond to the original video content, prior to the detection process. The only moving objects are the white circles that present distinct scales. The colored circumferences refer to the detected space-temporal blobs. As expected, this experiment illustrates what TRoF detector does: it pays more attention to the moving objects, and describes their space-temporal neighborhood with scale invariance.

In addition, for the sake of giving examples of motion detection that are associated with pornographic and non-pornographic contents, we make available in the Supplementary Materials two illustrative live-action video segments,[7] which put the TRoF

[6] Original and detected videos are available at the Supplementary Materials.

[7] These segments were obtained from the Pornography-2k dataset. The pornographic one is not explicit, though, for protecting readers.

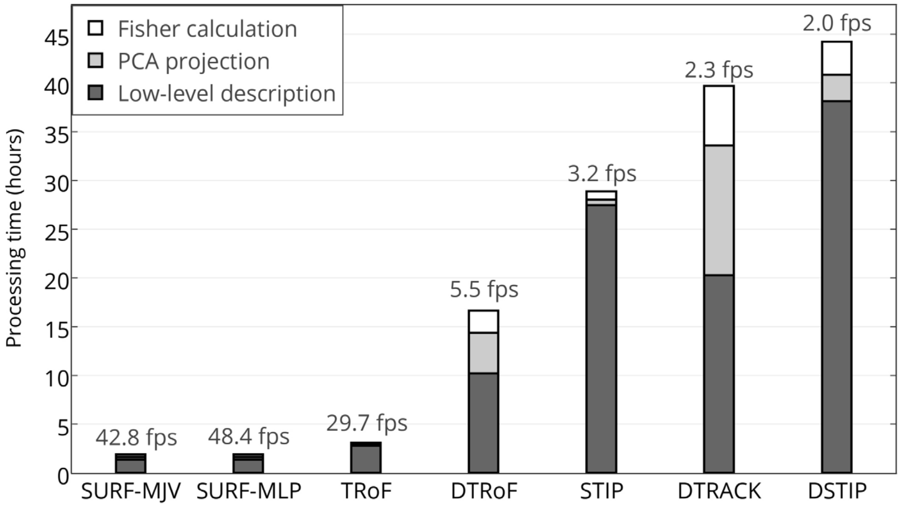


Fig. 10. Breakdown of the processing time spent by each BoVW-based solution. The computational times refer to the amount of time used to classify over 3 h of randomly chosen video footage, under the same system. That time is divided among three subtasks: low-level video description, PCA projection, and Fisher calculation. At the top of each bar, the respective video processing rate, in frames per second (fps). Notice that the sparse variant of TRoF is the only space-temporal solution able to provide speeds compatible with real-time video processing.

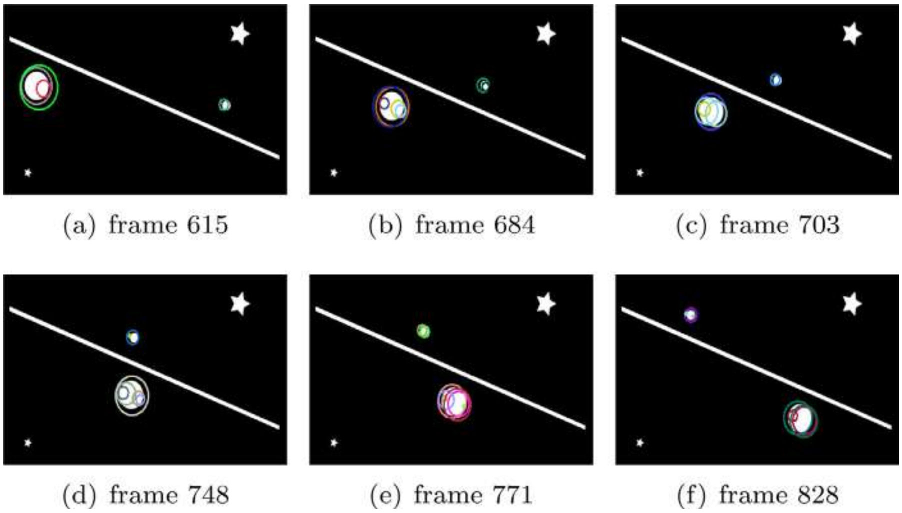


Fig. 11. TRoF blob detection on six frames sampled from a synthetic video. The original content, prior to the detection process, is expressed in white. The video depicts two moving circles that have distinct scales. All other white elements are static. Colored circles refer to the detected space-temporal blobs. As expected, TRoF detects only the moving objects, with scale invariance. An animated version of these frames can be found at the Supplementary Materials. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

detection capabilities in perspective with the Dense Trajectories [31] and STIP [30] strategies, for both positive and negative cases, respectively.

Finally, the numbers of TRoF in face of the Pornography-2k dataset are promising. Despite of presenting the same memory footprint than the baseline static solutions (SURF) — 236 GB to store the entire dataset description — sparse TRoF provides an error reduction of over 23%, and of over 13%, for ACC and F_2 measure, respectively, thanks to its space-temporal capabilities. Moreover, despite of being over twice as faster, the dense application of TRoF provides results that are on par with the performance of DTRACK.

7. Conclusion and future work

In this work, we proposed a BoVW-based framework for video-pornography classification, novel both in the low and mid-level stages.

In the low-level stage we have introduced TRoF, a new space-temporal interest point detector and local video descriptor, that quickly detects an optimized amount of interest points, allowing us to sparsely describe the video space–time in a very fast way.

In addition, to the best of our knowledge, it was the first time that the dense application of STIP [30] and Dense Trajectories [31] were evaluated to solve the problem of pornography classification. Similarly, this paper uses, for the first time, the Fisher Vector representation as a mid-level stage in pornographic content classification.

Our experiments confirmed that the incorporation of space-temporal information leads to more effective video-pornography classifiers and observed that the dense low-level video descriptions increase the system effectiveness (accuracy), but at prohibitive reductions in efficiency (computational time and memory footprint). Such drawback makes it impractical to apply dense strategies or conventional space-temporal approaches on hardware-limited hand-held device players, such as mobile phones and tablets. The sparse strategy of TRoF in contrast, allows a very good compromise of effectiveness and efficiency.

Our evaluation of the proposed framework took steps that also will help to advance the state of the art in pornography classification. The first, was the acquisition of the Pornography-2k dataset, a new challenging benchmark, with 1000 pornographic, and 1000 non-pornographic video clips, properly collected from the Internet.

Another useful contribution of this work is the evaluation of third-party classifiers. Among those solutions, we included two commercial programs based upon skin detectors, confirming that they are far from being reliable.

As future work, we envision the extension of our solution to work on live video streams, where the decision has to be taken in real-time, as the frames arrive. In this scenario, the challenge will reside in providing dynamic feature-pooling methods, and almost real time classification, in order to cut a scene at the moment it becomes pornographic. Additionally, we intend to extend the proposed pipeline to classify other sensitive video content, such as violence.

As the TRoF is a generic local feature locator/descriptor for videos, we also want to study its use and behavior to other computer vision tasks.

Acknowledgements

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).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.forsciint.2016.09.010.

References

[1] 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.

[2] G. Tyson, Y. Elkhatib, N. Sastry, S. Uhlig, Are people really social on porn 2.0? in: AAAI Conference on Web and Social Media (ICWSM), 2015, 1–9.

[3] W. Fisher, T. Kohut, L.D. Gioacchino, P. Fedoroff, Pornography, sex crime, and paraphilia, Curr. Psychiatry Rep. 15 (6) (2013) 1–8.

[4] Child Pornography: Model Legislation & Global Review. www.icmec.org/en-X1/pdf/Child-Pornography-Model-Law-English-7th-Edition-2012.pdf (accessed 11.12.14).

[5] Media Detective. www.mediadetective.com.

[6] Snitch Plus. www.hyperdynesoftware.com.

[7] PornSeer Pro. www.yangsky.com/products/dshowseer/porndetection/PornSeePro.

[8] M. Polastro, P. Eleuterio, Nudetective: a forensic tool to help combat child pornography through automatic nudity detection, in: IEEE Database and Expert Systems Applications (DEXA), 2010, 349–353.

[9] C. Platzer, M. Stuetz, M. Lindorfer, Skin Sheriff: a machine learning solution for detecting explicit images, in: International Workshop on Security and Forensics in Communication Systems, 2014, 45–56.

[10] J.-S. Lee, Y.-M. Kuo, P.-C. Chung, E.-L. Chen, Naked image detection based on adaptive and extensible skin color model, Pattern Recognit. 40 (8) (2007) 2261–2270.

[11] J.-L. Shih, C.-H. Lee, C.-S. Yang, An adult image identification system employing image retrieval technique, Pattern Recognit. Lett. 28 (16) (2007) 2367–2374.

[12] M. Jones, J. Rehg, Statistical color models with application to skin detection, Int. J. Comput. Vis.) 46 (1) (2002) 81–96.

[13] D. Forsyth, M. Fleck, Automatic detection of human nudes, Int. J. Comput. Vis. 32 (1) (1999) 63–77.

[14] D. Forsyth, M. Fleck, Body plans, in: Conference on Computer Vision and Pattern Recognition (CVPR), 1997, 678–683.

[15] M. Fleck, D. Forsyth, C. Bregler, Finding naked people, in: European Conference on Computer Vision (ECCV), 1996, 593–602.

[16] C.C. Yan, Y. Liu, H. Xie, Z. Liao, J. Yin, Extracting salient region for pornographic image detection, J. Vis. Commun. Image Represent. 25 (5) (2014) 1130–1135.

[17] F. Souza, E. Valle, G. Cámara-Chávez, A. Araújo, An evaluation on color invariant based local spatiotemporal features for action recognition, in: Conference on Graphics, Patterns and Images (SIBGRAPI), 2012, 31–36.

[18] E. Valle, S. Avila, F. Souza, M. Coelho, A. Araújo, Content-based filtering for video sharing social networks, in: Brazilian Symposium on Information and Computer System Security (SBSeg), 2012, 625–638.

[19] C. Caetano, S. Avila, S. Guimarães, A. Araújo, Pornography detection using BossaNova video descriptor, in: European Signal Processing Conference (EUSIPCO), 2014, 1681–1685.

[20] C. Caetano, S. Avila, S. Guimarães, A. Araújo, Representing local binary descriptors with BossaNova for visual recognition, in: ACM Symposium On Applied Computing (SAC), 2014, 49–54.

[21] 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 (2013) 453–465.

[22] S. Avila, N. Thome, M. Cord, E. Valle, A. Araújo, BOSSA: extended bow formalism for image classification, in: IEEE International Conference on Image Processing (ICIP), 2011, 2909–2912.

[23] C. Jansohn, A. Ulges, T. Breuel, Detecting pornographic video content by combining image features with motion information, in: ACM International Conference on Multimedia (MM), 2009, 601–604.

[24] A. Lopes, S. Avila, A. Peixoto, R. Oliveira, M. Coelho, A. Araújo, Nude detection in video using bag-of-visual-features, in: Conference on Graphics, Patterns and Images (SIBGRAPI), 2009, 224–231.

[25] A. Lopes, S. Avila, A. Peixoto, R. Oliveira, A. Araújo, A bag-of-features approach based on hue-SIFT descriptor for nude detection, in: European Signal Processing Conference (EUSIPCO), 2009, 1152–1156.

[26] J. Zhang, L. Sui, L. Zhuo, Z. Li, Y. Yang, An approach of bag-of-words based on visual attention model for pornographic images recognition in compressed domain, Neurocomputing 110 (2013) 145–152.

[27] C. Steel, The mask-SIFT cascading classifier for pornography detection, in: IEEE World Congress on Internet Security (WorldCIS), 2012, 139–142.

[28] A. Ulges, A. Stahl, Automatic detection of child pornography using color visual words, in: IEEE International Conference on Multimedia and Expo (ICME), 2011, 1–6.

[29] T. Deselaers, L. Pimenidis, H. Ney, Bag-of-visual-words models for adult image classification and filtering, in: IEEE International Conference on Pattern Recognition (ICPR), 2008, 1–4.

[30] I. Laptev, On space–time interest points, Int. J. Comput. Vis. 64 (2–3) (2005) 107–123.

[31] 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.

[32] L. Zhuo, Z. Geng, J. Zhang, X.G. Li, ORB feature based web pornographic image recognition, Neurocomputing 173 (3) (2016) 511–517.

[33] C. Caetano, S. Avila, W. Schwartz, S. Guimarães, A. Araújo, A mid-level video representation based on binary descriptors: a case study for pornography detection, Neurocomputing (2016) (in press).

[34] M. Moustafa, Applying deep learning to classify pornographic images and videos, in: Pacific Rim Symposium on Image and Video Technology (PSIVT), 2015.

[35] W. Kelly, A. Donnellan, D. Molloy, Screening for objectionable images: a review of skin detection techniques, in: IEEE International Machine Vision and Image Processing Conference (IMVIP), 2008, 151–158.

[36] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, SURF: speeded up robust features, Comput. Vis. Image Underst. 110 (3) (2008) 346–359.

[37] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, in: IEEE International Conference on Computer Vision (ICCV), 2011, 2564–2571.

[38] I. Laptev, M. Marszałek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2008, 1–8.

[39] Q. Zhiyi, L. Yanmin, L. Ying, J. Kang, A method for reciprocating motion detection in porn video based on motion features, in: IEEE International Conference on Broadband Network & Multimedia Technology (ICBNMT), 2009, 183–187.

[40] T. Endeshaw, J. Garcia, A. Jakobsson, Classification of indecent videos by low complexity repetitive motion detection, in: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2008, 1–7.

[41] N. Rea, G. Lacey, C. Lambe, R. Dahyot, Multimodal periodicity analysis for illicit content detection in videos, in: European Conference on Visual Media Production (CVMP), 2006, 106–114.

[42] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Good practice in large-scale learning for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 507–520.

[43] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Found. Trends Comput. Graph. Vis. 3 (3) (2008) 177–280.

[44] D. Lowe, Object recognition from local scale-invariant features, in: IEEE International Conference on Computer Vision (ICCV), vol. 2, 1999, 1150–1157.

[45] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: British Machine Vision Conference (BMVC), 2011, 1–12.

[46] Y.-L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2010, 2559–2566.

[47] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, 281–297.

[48] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: International Conference on Computer Vision (ICCV), vol. 2, 2003, 1470–1477.

[49] P. Koniusz, F. Yan, K. Mikolajczyk, Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection, Comput. Vis. Image Underst. 117 (5) (2013) 479–492.

[50] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European Conference on Computer Vision (ECCV), 2010, 143–156.

[51] G. Willems, T. Tuytelaars, L.V. Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: European Conference on Computer Vision (ECCV), 2008, 650–663.

[52] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2005, 886–893.

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

[54] 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.

[55] 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 (2008) 1871–1874.

[56] J. Ozer, Understanding H.264 Encoding Parameters – I, P and B-Frames, 2009, http://www.streaminglearningcenter.com/articles/producing-h264-video-for-flash-an-overview.html?page=4.