Video pornography detection through deep learning techniques and motion information

Mauricio Perez[a], Sandra Avila[b], Daniel Moreira[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*

Abstract

Recent literature has explored automated pornographic detection — a bold move to replace humans in the tedious task of moderating online content. Unfortunately, on scenes with high skin exposure, such as people sunbathing and wrestling, the state of the art can have many false alarms. This paper is based on the premise that incorporating motion information in the models can alleviate the problem of mapping skin exposure to pornographic content, and advances the bar on automated pornography detection with the use of motion information and deep learning architectures. Deep Learning, especially in the form of Convolutional Neural Networks, have striking results on computer vision, but their potential for pornography detection is yet to be fully explored through the use of motion information. We propose novel ways for combining static (picture) and dynamic (motion) information using optical flow and MPEG motion vectors. We show that both methods provide equivalent accuracies, but that MPEG motion vectors allow a more efficient implementation. The best proposed method yields a classification accuracy of 97.9% — an error reduction of 64.4% when compared to the state of the art — on a dataset of 800 challenging test cases. Finally, we present and discuss results on a larger, and more challenging, dataset.

Keywords: Pornography classification, Deep learning and motion information, Optical flow, MPEG motion vectors, Sensitive video classification

1. Introduction

Filtering sensitive media (pornographic, violent, gory, etc.) has growing importance, due to the booming consumption of online media by people of all ages; and among sensitive media types, pornography is often the most unwelcome. A

*Corresponding author: anderson.rocha@ic.unicamp.br

range of applications has increased societal interest on the problem, e.g., detecting inappropriate behavior via surveillance cameras; or curtailing the exchange of sexually-charged instant messages, also known as "sexting", by minors. In addition, law enforcers may use pornography filters as a first sieve when looking for child pornography in the forensic examination of computers, or internet content. The main application, however, remains preventing uploading or accessing undesired content for certain demographics (e.g., minors), or environments (e.g., schools, workplace).

The precise definition of pornography is, of course, subjective, but here we will consider "any sexually explicit material with the aim of sexual arousal or fantasy" [1].

A natural approach to pornography detection consists in first trying to detect nudity [2–5] and then defining appropriate thresholds to further filter the content. Such solutions commonly use human skin features, such as color and texture, and human geometry [6–9]. Those methods normally use that information for modeling which pixel values and spatial distribution characterize a nude person. Although the motivation for such methods is intuitive, it reveals ultimately naïve. People may show a lot of skin in activities that have nothing to do with sex (e.g., sunbathing, swimming, running, wrestling), leading to a lot of false positives. Skin exposure in itself, is not a reliable proxy for pornography detection. Conversely, some sexual practices involve very little exposed skin, leading to unacceptable false negatives. In addition, the reliance on many adhoc thresholds hinders the generalization of those techniques when facing diversity of ethnicities and skin colors.

Departing from the low-level skin-based methods, in more recent years, several authors have explored other types of solutions for adult content filtering, specially the ones inspired by the bag of words model from text classification, [10–14]. Those methods insert an intermediary description stage between the low-level features extracted from the images, and the classification component. Such methods normally involve choosing some low-level feature representation (e.g., gradient-like information), and creating a representative codebook. The involved steps are referred to as generating the codebook, coding the features and pooling the codewords count. In the end, a classifier learns, through examples, which representations belong to the pornography class. Clearly, such methods are more robust than the skin-based ones, but still suffer from some ambiguous cases. Choosing the codewords, the size of the codebook and which of the many coding and pooling strategies to use are also crucial steps for the good performance of the solutions.

Although thus far relatively underestimated for this problem, the motion information available in videos would likely help to disambiguate the most difficult cases in pornography classification. Unfortunately, only a few works have exploited spatio-temporal features or motion information in this problem until now [11, 15–17]. In these cases, the spatio-temporal feature evaluated was Space-Time Interest Points (STIP) [17], Dense Trajectories [18], Temporal Robust Features (TRoF) [16], and the motion information coming from a statistical analysis of the MPEG Motion Vectors. Particularly, in [16], the experiments

confirmed that the incorporation of spatio-temporal information leads to more effective video-pornography classifiers. In that work, the authors showed that a custom-tailored method to capture motion outperforms the mentioned dense trajectories. In this work, in turn, we show that data-driven features are even more powerful, especially when extracted both from spatial and temporal data.

Given the difficulty of developing appropriate thresholds for skin-based detectors and also the several available choices when coding low- and mid-level features, in addition to the lack of proper motion-based features, and the recent success of Deep Learning solutions on similar tasks, we set forth the task of designing and developing deep learning techniques, to automatically grasp static and motion-based deep representations, straight from the data, that could leverage pornography classification.

Amongst the many machine learning techniques available, Deep Neural Networks, more specifically Convolutional Neural Networks (CNN), are showing groundbreaking results for image and video classification tasks [19–22]. Of particular attention, some authors have been studying how to adapt CNNs for human action recognition in videos, whereby the spatio-temporal information can be explored to improve the extracted features [20, 23–25]. Different architectures are possible, each one combining the spatial and temporal information in different ways, leading to better or worse features for the classification task. Some authors sought to extract the motion information implicitly by feeding a sequence of frames to the CNNs [23–25], while others opted for explicitly feeding this information to the network through a previously computed Optical Flow Displacement Fields image representation [20].

In spite of the success of deep learning techniques in the computer vision arena, their literature on pornography detection is very scarce. Pioneering the trend for pornography detection, Moustafa [22] has explored majority voting classification on a sample of frames classified with off-the-shelf CNNs. However, the authors did not explore the most appropriate network configurations, parameters nor any spatio-temporal or motion information in their solution.

When targeting at sensitive media filtering, some interesting challenges appear for deep learning-based solutions: how to define an appropriate architecture; the possibility of reusing already trained architectures for related image categorization problems, thus avoiding the need for huge amounts of training data; and how to incorporate time/motion information, which complements the spatial/static information.

In this work, we design and develop deep learning-based approaches to automatically extracting discriminative spatio-temporal characteristics for filtering pornographic content in videos. As far as we know, this is the first time convolutional neural networks — along with motion information — is applied for pornography detection in videos. Although in this work we focus on the pornography modality, the methodology we discuss herein is versatile and its extension to other types of sensitive content is straightforward. The contributions of this paper are three-fold:

i) A novel method for classifying pornographic videos, using convolutional

neural networks along with static and motion information;

ii) A new technique for exploring the motion information contained in the MPEG motion vectors [26];

iii) A study of different forms of combining the static and motion information extracted from questioned videos.

We organized the remaining of this paper into five sections. In Sec. 2, we discuss existing approaches for dealing with the pornography detection problem, while in Sec. 3, we present a short summary of the necessary concepts to understanding this work. We then move on to Sec. 4, in which we introduce the methods we propose for classifying pornography in videos, incorporating static and motion information. In Sec. 5, we present the experimental setup, along with the experiments and validation of the proposed methods and existing counterparts in the literature. Finally, in Sec. 6, we conclude the work and point out to some possible future research directions.

2. Literature Review on Pornography Detection

Short et al. [1] wrote a review of 46 articles that approached, to some extent, internet pornography. In their work, the authors highlighted the importance of explicitly defining the term pornography in each work, since it has direct influence on the results and issues that can be encountered further on. The definition is also relevant for comparisons among different works. As an example, some works consider the presence of genitals as being enough for classifying the content as pornography, whereas other authors argue that explicit sexual acts are necessary. It is proposed that a well-formalized definition should contain the type of pornography and the reason that it is apparently expecting to motivate the viewers. The definition we adopted in this work, proposed by Short et al. [1], denotes pornography as "any sexually explicit material with the aim of sexual arousal or fantasy".

2.1. Skin-based Techniques

Nudity detection using skin information has been extensively explored in the literature [2–5, 27]. Fleck et al. [2] proposed a two-step content-based retrieval strategy for returning images with naked people. First, the method filters the images that have large areas of skin regions. To identify skin pixels, it is used thresholds on the intensity, hue and saturation value of each pixel. Then, these areas are grouped and analyzed geometrically, validating if they could represent human limbs. On one hand, the authors point out that the first phase is vulnerable to scale and saturation, and returns false positives from scenes with many people, or from materials with colors that are similar to human skin. On the other, the geometrical analysis suffers from missing limbs because of occlusion, close-ups or even by failure of the skin detector, among other reasons. These aspects lead to low precision and recall measures, when in comparison to newer methods we shall see later.

Following a similar path, Jones and Rehg [6] focused exclusively on the color information from the pixels, building skin-based statistical models. A histogram of 256 bins for each channel is computed from the skin images, and another for the non-skin. These histograms model the probability of the color belonging to a skin region. With a standard likelihood ratio approach, an RGB value can be labeled skin if above a certain threshold. A feature vector is then created comprising features that include the number of pixels detected as skin and the average confidence of the detected skin. A C4.5 decision tree classifier is used for the decision-making process.

2.2. Bag-of-Visual-Words techniques

The next milestone for the pornography detection problem was reached with the Bag-of-Visual-Words (BoVW) models. Deselaers et al. [10], aware that this type of solution had showed good results in many image classification problems, built a classifier for adult images using visual codebooks. Patches around interest points, with scaling and dimensionality reduction via Principal Component Analysis (PCA), were used as features. Codewords were selected through Gaussian mixture models, generating the codebook. The authors employed a hard-assignment coding policy followed by sum pooling. Other types of coding and pooling were proposed later on. The decision making considered Support Vector Machines (SVM) and Log-linear classifiers. The reported results showed that their method clearly outperforms the previous methods, mainly based on color features. In addition, combining this method with skin-based ones does not bring anything new to the table.

2.3. Classifying Videos

When it comes to video, the basic approach considers extracting the frames and applying an image-based description and classification approach. However, these methods disregard valuable information that videos provide, the concept of motion. Although not directly in the field of pornography detection, the importance of temporal information for action recognition has been assessed for many years now. Dollar et al. [28] proposed a corner detector algorithm similar to the Harris detector [29], that seeks for "corners" in time. The detected "motion" corners are then described with cuboids around them. With the help of a codebook from the cuboids, the histograms of features from the short scenes demonstrated greater discrimination than the spatial-based descriptors. Some recent works keep the trend of exploring motion information for action recognition as Laptev et al. [30], Wang et al. [18], and Simonyan and Zisserman [20].

Turning our attention to pornography detection, Jansohn et al. [11] were one of the first authors to explore the time information while detecting pornography. They used a statistical analysis of MPEG-4 motion vectors, with a bag of visual words similar to the one proposed by Deselaers et al. [10]. Different ways of combining the motion vector information in overlapping windows of time were experimented. A description of the video was generated by pooling these windows, generating a motion histogram. The decision making in the end considers

an SVM classifier. The classifier using the time information alone gave effective results and was improved upon when combined with a BoVW-based (spatial characterization) approach.

Avila et al. [14] proposed an extension of the bag-of-visual-words approach for the pornographic video detection task. The improved design involved new pooling and coding formalisms for the local descriptors. Instead of simply summing up the activations in the pooling step, as in Deselaers et al. [10], an estimation on the distribution of the descriptors distance to the codewords is used. For the new coding, a semi-soft scheme was used, on which different softness parameters, based on the variation of each cluster, are applied. The decision-making process considers an SVM classifier. Different datasets were used to validate the extension including a pornographic benchmark available online.

Another supplementary information provided by video, that can be used for pornography detection, comes from audio. Rea et al. [31] proposed an audio feature extraction approach for this problem, that consists of analyzing the periodicity from the sound. The inspiration comes from the fact that this type of content usually has repetitive sounds. To capture and measure the periodicity, an autocorrelation of the energy filter is applied to the audio signal, and the area between the local maxima's and minima's curves is computed. If the area is above a certain threshold, that configures a repetitive sound, suggesting it comes from pornography. In an evaluation with diverse audio samples, not from pornographic content, a false alarm rate of 2% was reported. However, as the authors point out, this approach alone is not robust to other periodic sounds, such as in a tennis match, hence visual features should also be present to remove ambiguities.

Although the audio information might also be useful in the intricate task of pornography detection, in this work, we do not take audio into consideration. As a matter of fact, we opted to solely focus on visual information.

2.4. Convolutional Neural Networks

Although Deep Learning has been responsible for most of the current breakthroughs in image classification tasks, few explorations have been made for these techniques within the context of pornography detection in video. Moustafa [22] performed a superficial adaptation of well-known CNN architectures for image classification, to the pornographic video classification task. He used AlexNet [19] and GoogLeNet [21] architectures directly on selected frames, for classifying them in porn or non-porn, integrating the final result for a video through a majority voting process. The author used the weights learned from ImageNet dataset, fine-tuning only the last layer, which corresponds to the classifier. Within this approach, no motion information was explored.

2.5. Third-party Solutions

The nudity and pornography detection problem has not been tackled only in the academia. There exists some software solutions, mostly commercial, aiming at solving this problem. Some focus on blocking websites that contain this type

of content (e.g., CyberPatrol, CYBERsitter, NetNanny, K9 Web Protection, Profil Parental Filter) while others scan the hard drive in search of pornography (e.g., SurfRecon, Porn Detection Stick, PornSeer Pro). There is even a Brazilian software, called NuDetective, developed by the Brazilian Federal Police, that focus on detecting child pornography. These solutions are mainly based on skin detection approaches, and none explored the space-time nature of videos for aiding the detection of pornography. Therefore, such solutions normally fall short when compared to the current state of the art for this problem.

2.6. Summary Table

Table 1 presents an overview of the related works on the pornography detection problem and its sub-problems, skin and nudity detection.

3. Related Concepts

In this section, we cover the main concepts necessary for understanding this manuscript. In Sec. 3.1, we explain some Convolutional Neural Networks (CNNs) extensions to deal with motion/temporal information while, in Sec. 3.2, we explain the motion information sources of interest in this work, Optical Flow and MPEG Motion Vectors.

3.1. Motion/Temporal Networks

The first CNNs architectures were designed targeting at image classification. Although they can be used for video classification in a frame-wise approach, the temporal information will be almost completely discarded. As previously discussed, this type of information should not only increase the performance, but it could also be indispensable for removing the ambiguity on the pornography classification problem. However, only a few authors have addressed video classification with Convolutional Networks [20, 25] thus far, offering us a whole new venue for possible original contributions.

Karpathy et al. [25] explored a variety of architectures to implicitly capture temporal information amongst a sequence of frames. These architectures received sequential frames, or frames temporally close to each other, as input. The reported results exhibited small performance variance between the fusion approaches and also in comparison with the single-frame network. These initial results indicate that Convolutional Networks have some troubles to implicitly capture the motion information from sequential frames.

Following a different strategy, Simonyan and Zisserman [20] proposed a Two-Stream convolutional neural network, that uses optical flow to supply complementary information to the classification. Inspired upon the biological aspect of human vision, they designed an architecture related to the two-stream hypothesis, in which the visual cortex separately recognizes objects and motion [41]. This is accomplished by having an architecture with two pathways, one for the frames and another for the motion information. The pathways are later combined by score averaging. For the motion information, the authors used

Table 1: Summary of approaches on skin, nudity or pornography detection. The "Type" column comprises four categories: Nude Image (NI), Nude Video (NV), Porn Image (PI) and Porn Video (PV).

Authors	Type	Method	Classifier
Fleck et al. [2]	NI	Skin detection; geometrical analysis	Threshold
Lopes et al. [32]	NI	BoVW model; PCA on SIFT and Hue-SIFT descriptors	Linear SVM
Lopes et al. [33]	NV	BoVW model; PCA on SIFT and Hue-SIFT descriptors; voting scheme	Linear SVM
Jones and Rehg [6]	PI	Skin color histogram; color probabilities	C4.5 decision tree
Rowley et al. [7]	PI	Skin color histogram; skin texture histogram; face detection	RBF SVM
Zheng et al. [5]	PI	Skin color detection; skin region detection; shape descriptors	AdaBoost with C4.5 decision tree
Zuo et al. [34]	PI	Patch-based skin color detection; human body part detection	Random forest
Deselaers et al. [10]	PI	BoVW model; PCA on SIFT descriptors; GMM model	SVM; histogram intersection kernel
Ulges and Stahl [12]	PI	BoVW model; DCT in YUV color space	SVM; χ^2 kernel
Steel [13]	PI	Mask-SIFT; skin percentage	Cascade classifier of three stages
Zaidan et al. [35]	PI	Bayesian method with a grouping histogram; segmentation with back-propagation neural network	Artificial neural network
Zhuo et al. [36]	PI	ORB descriptors; BoVW model	SVM
Nian et al. [37]	PI	CNN architecture CaffeNet	CNN softmax
Jansohn et al. [11]	PV	BoVW model; DCT in YUV color space; motion histograms	SVM; late fusion
Avila et al. [14]*	PV	BoVW-based model: BossaNova; Hue-SIFT descriptors	RBF SVM
Caetano et al. [38]*	PV	BoVW-based model: BossaNova; binary descriptors	RBF SVM
Caetano et al. [39]*	PV	BoVW-based model: BossaNova; binary descriptors; multiple aggregation functions	RBF SVM
Valle et al. [15]*	PV	BoVW model; STIP descriptors	Linear SVM
Moreira et al. [16]*	PV	BoVW-based model: TRoF descriptors	Linear SVM
Rea et al. [31]	PV	Skin color estimation; MPEG motion information; periodic patterns detection	Threshold over periodicity measure
Ulges et al. [40]	PV	BoVW model; DCT in YUV color space; MFCC audio features; motion histograms; skin detection	SVM; RBF and χ^2 kernels; late fusion
Moustafa [22]*	PV	CNNs on raw frames	Majority voting

*The reported results from these works are used for comparison with our proposed approaches

stacked optical flow displacement fields. This stacking comprises the image representations of the vertical and horizontal components, from the displacement

vector field, of an arbitrary number of consecutive frames. This new representation strategy led to important improvement upon previous state-of-the-art deep neural nets on action recognition datasets such as the UCF-101 [42] and HMDB-51 [43].

The aforementioned networks were designed for action recognitions tasks (e.g., golfing, sitting, running, etc.), which at first may seem very different than pornographic detection on videos. But, although pornography is more complex and subjective than such actions, it can be reduced to a collection of smaller actions that characterize it. Therefore, through learning, convolutional networks can also identify the static and temporal visual patterns, that will lead to discriminative features for pornography detection in videos.

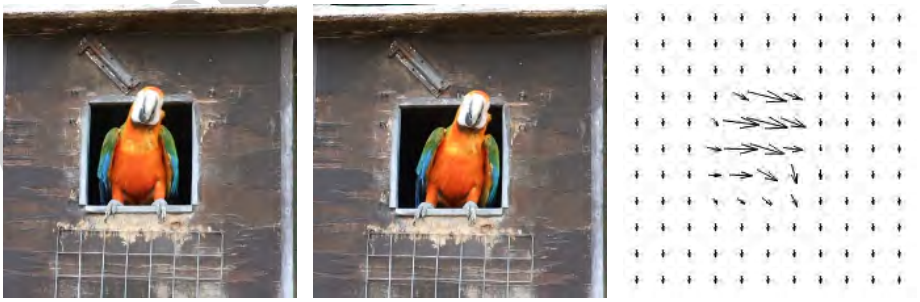
3.2. Motion Information

As stated earlier, the motion information contained in a sequence of images is invaluable for tasks of video classification. We now review two motion information sources of particular interest: Optical Flow and MPEG Motion Vectors.

3.2.1. Optical Flow

Optical flow comes from the computer vision problem of estimating the visual motion between two images [44]. Although this is an old problem in Computer Vision, there are somewhat recent works improving optical flow computing [45].

The final output for the different optical flow computation methods is a displacement field, which contains, in each pixel, its relative movement from the source image to the reference one. Each position in this field has a displacement vector indicating the estimation of which direction the respective pixel has moved to and the intensity (gradient) of this movement. Fig. 1 depicts an example of the output. Altogether, these vectors provide us with a relevant proxy for the motion of the objects in the scene.



(a) Previous Frame (b) Next Frame (c) Displacement Field

Figure 1: Sequential raw frames (left and middle) and the respective Optical-Flow Displacement Field (right) computed from them. The regions with more movement in the raw frames (e.g., macaw's body and head) are also the ones with the greatest displacement vectors in the field. The original images are under Creative Commons Licence.

3.2.2. MPEG Motion Vectors

Another source of motion information frequently used for video classification [11, 31] is the motion vectors contained in the MPEG coding of the analyzed media. Differently from the optical flow, this motion estimation was not originally created to allow an understanding of the movement in the scene; but rather, it was designed for aiding the compression of the video [26].

During video compressing, there is a method named Motion estimation and compensation [26], which in one of its steps, maps the pixel movements between the current frame being compressed onto a reference frame. This information is what is called *Motion Vectors*. Motion estimation in video compression is performed in a block-based fashion, where pixels are grouped in macroblocks in order to reduce computational complexity.

Motion vectors are then computed per macroblock and contain the following information: the position (x,y) of the macroblock of pixels in the current frame; its position (x',y') in the reference frame; and the size of the macroblock $(M \times N)$. Fig. 2 shows an example. This mapping only occurs during compression, hence, when the video is decompressed for analysis, this information is readily available. The gathering of the motion vectors, from a frame being reconstructed, gives us useful information about the motion that has occurred at that time.

Although we reference here to the motion vectors from the MPEG codec, and this codec is one of the most used today, this source of information is commonly present in other codecs as well, such as Google's VP9 [46].

Figure 2: Example of a macroblock and its respective Motion Vector between the current frame (left) and the reference frame (right). The original images are under Creative Commons Licence

4. Methodology

In this section, we present the details of our proposed method for exploring and developing deep learning techniques, jointly with motion information, targeting at video pornography detection. The approaches we designed, were mainly inspired upon the seminal work of Simonyan and Zisserman [20], in

which the motion information is explicitly provided to the convolutional neural network, and each type of information (static and motion) is independently processed by the network. Notwithstanding, we explore the motion information differently and incorporate novel sources of motion information in our work. Moreover, we also propose new ways for combining static and motion for a more effective decision making.

In Sec. 4.1, we present the details for the static stream of information while in Sec. 4.2, we explain the motion stream and also the two motion sources explored in this work: optical flows and MPEG motion vectors. Thereafter, in Sec. 4.3, we describe the distinct ways we have explored for fusing the static and motion information in this work. Finally, in Sec. 4.4, we detail the CNN architecture and training process we adopted.

4.1. Static Information

In the static pipeline we propose, which is represented in Fig. 3, we start with a chosen sampling of the video frames and extract their features with a convolutional network. These features are average pooled to form a single description of the whole video (there are some alternatives to the pooling, e.g., voting, and other types of pooling, such as max and sum, but throughout some prior experiments and our own experience, we opted for a standard average pooling procedure). Finally, we feed a classifier with the video description for the final classification. One can see this is the simplest possible approach for tackling a video: divide it into frames, pool the different features and train a classifier.

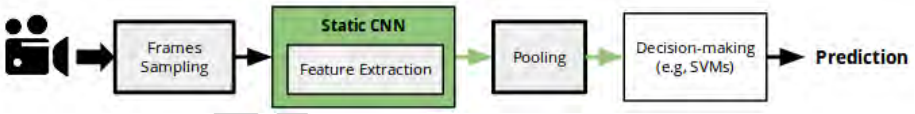


Figure 3: Pipeline for the static information flow. It comprises the feature extraction from a sampling of the frames, which are average pooled for feeding a decision-making classifier in the end.

In addition, each frame is preprocessed, being resized, maintaining the aspect ratio and having its smaller dimension as the network input dimension (224×224 pixels). Then a center cropping is performed, resulting in an image with the necessary shape for the convolutional network architecture we adopt.

For the static CNN, we explored two paths for solving the problem. The first one considers a network model trained with natural images obtained with the ImageNet dataset [47] whilst the second model is custom-tailored (i.e., fine-tuned and properly adapted) to our problem, starting with the weights obtained with the ImageNet samples during a pre-training step rather than using random weights for initializing the network. Our experience shows that initializing weights with a related (although not directly) problem is more effective than random weights for this particular problem.

4.2. Motion Information

The most important challenge we want to tackle is how to add time information (motion) to the pipeline, since it was demonstrated that it enhances classification power [11, 20, 25] in other problems. Initially, we analyze the motion information independently from the static information. The pipeline (Fig. 4) for this type of information is somewhat similar to the static pipeline, with differences in the input and output of the convolutional network.

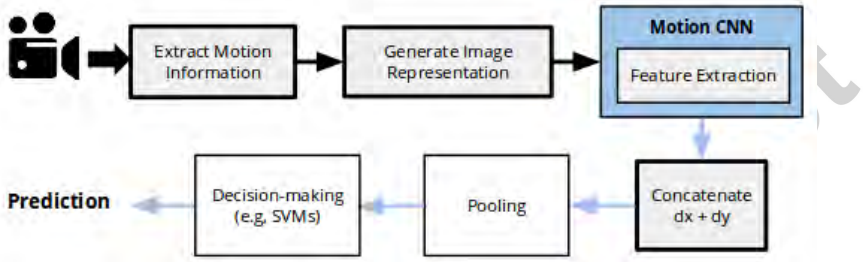


Figure 4: Pipeline for the motion information flow. It comprises the extraction of the motion information from the video; generation of an image representation to this extracted information; the feature extraction with the selected motion CNN (each motion source has its own CNN model); concatenation of the horizontal (dx) and vertical (dy) descriptions; average pooling of the descriptions; and a classifier (e.g., Support Vector Machines) for the final classification.

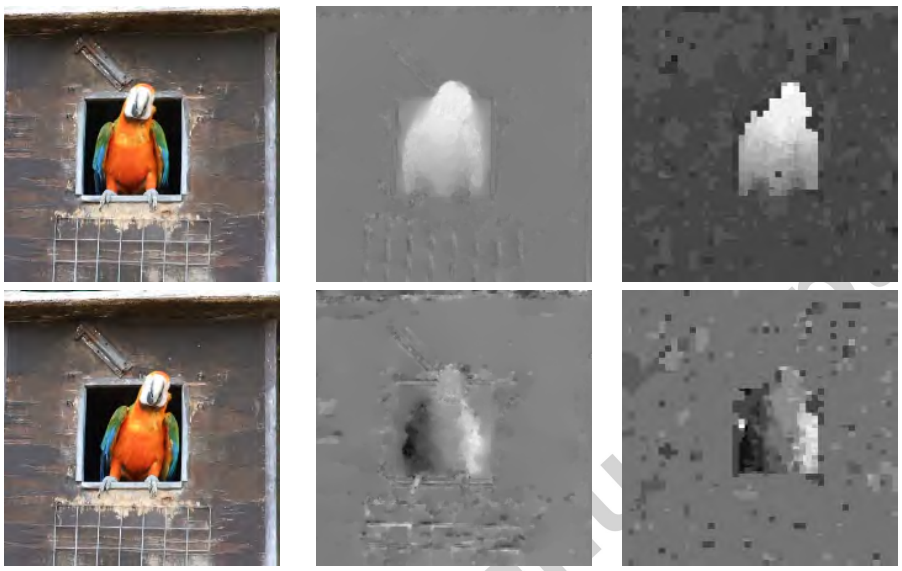
In our methodology, we evaluate two sources for the motion information: optical flow displacement fields [44] and MPEG motion vectors [26]. The motion sources follow the pipeline independently, therefore there is a specific motion CNN model and classifier for each. Each source requires an unique form for extracting the motion information, whose details we shall present later on.

It is important to highlight that the motion information does not come ready-to-use in a CNN and require, upfront, a proper representation. We represent the motion information, extracted with optical flows or MPEG motion vectors, by two motion maps, one for the horizontal (dx) component of the motion and another for the vertical (dy), containing in each (x,y) position, a measure of motion in that respective direction. When transforming these motion maps to images, we linearly rescale them to the $[0, 255]$ interval and store them as gray-scale images, one image for each component of the motion. Fig. 5 depicts examples of the generated image representations.

After the feature extraction, the descriptions of the components (dx and dy) from the same motion are concatenated to form a single feature vector. The rest of the pipeline is then similar to the static one: the combined descriptions are pooled and fed to a classifier for final decision making.

At first sight, this pipeline is similar to Simonyan and Zisserman [20] temporal stream. However, here we have opted for each motion information and each component to be separately processed by the convolutional neural network, in contrast to Simonyan and Zisserman [20], who stacked both components of the motion information from a temporal neighborhood (e.g., displacement fields

from an arbitrary number of consecutive frames) before feeding it to the network.



(a) Sequential Raw frames (b) Optical Flow (c) Motion Vectors

Figure 5: Sequential raw frames (a) and motion image representations from optical flows (b) and MPEG motion vectors (c). The horizontal (dx) component is on top, and the vertical (dy) one is on the bottom. The regions with more movement in raw frames (e.g., macaw's head and body) appear highlighted (dark or light) in the motion representations, while regions without movement correspond to the neutral middle gray. The original images are under Creative Commons Licence.

4.2.1. Optical Flow

Our first explored source of motion information is the optical flow displacement fields technique. Each position of interest provides us with the gradient's magnitude and the direction of the motion. For a more direct representation, we decompose this information in its horizontal (dx) and vertical (dy) components, generating two motion maps with the magnitude values for each component separately. Fig. 5(b) depicts an example of the optical flow representation, calculated from the generated motion maps (see Sec. 4.2).

We compute the optical flow displacement fields using Brox et al.'s method [45], whose GPU implementation is readily available at OpenCV 2.4.10 toolbox. The frames, and their pairs, were preprocessed before extraction of the optical flows, just as the raw frames: resizing preserving the aspect ratio, then center cropping with the input dimensions of the chosen CNN.

4.2.2. MPEG Motion Vectors

Another explored source of the motion information is the motion vector data encoded within the MPEG codec. In each vector for a particular frame, it is

encoded the position from a given macroblock of pixels in the current frame and its position at the reference frame. In this work, we propose a novel representation for the motion information contained in these vectors. We measure how much the block from each motion vector has moved by computing the distance, in pixel coordinates, from the reference position to the current position in each direction, horizontal and vertical, separately. Furthermore, these distances are analogous to the magnitude of the movement at the region contained in that macroblock, and generate two motion maps, one for each direction, similarly to the optical flow motion extraction.

Motion vectors are extracted using FFMPEG 2.7 API. They are extracted from the original videos and no resizing is performed. Therefore, differently from optical flow, for the motion vectors, we apply the resizing operation later on, directly on the generated image representations. Fig. 5(c) illustrates an example of the generated image representation.

4.3. Fusion

The static and motion information can lead to more effective results if their collected evidence (video telltales) are complementary in some sense. Therefore in this section, we explore different forms of combining them.

4.3.1. Early Fusion

In the early fusion method, the static and the motion information are combined at the very beginning of the pipeline, being processed together by a special convolutional network. This way, the features benefit from both the static and the motion information. Fig. 6 depicts a representation of the pipeline.

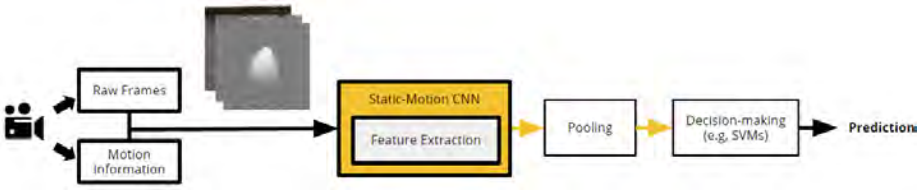


Figure 6: Pipeline for the early fusion strategy. The static and motion information are combined before feature extraction, through a custom-tailored CNN trained for extracting features with both the static and motion information.

The three color channels of the frame, along with its respective motion representations, dx and dy, are stacked together for input in the convolutional network, giving rise to a 5-channel input. It is also straightforward to generate an image containing the raw frame information in gray scale on one of its channels and the motion information on the other two channels, one for the horizontal component and another for the vertical. The advantage of having a 3-channel input is the ability to custom-tailor the network weights starting from pre-trained 3-channel network weights instead of starting the weights randomly from scratch. In this work, we have explored both options.

4.3.2. Mid-level Fusion

Differently from the early fusion strategy, in the mid-level fusion, we concatenate the features extracted from each type of information (static or motion-based), and from each independent CNN, into a single feature vector before feeding a classifier. Fig. 7 shows a representation of the mid-level fusion pipeline.

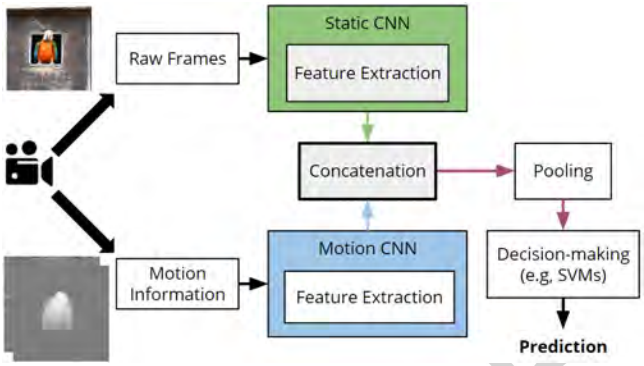


Figure 7: Pipeline for the mid-level fusion. The fusion of static and motion information takes place after feature extraction, and before the decision making, by concatenating the feature vectors into a single representation vector.

4.3.3. Late Fusion

In this fusion scheme, each information is processed by a separate decision-making approach (e.g., SVM classifier), generating independent classification scores that can then be combined later on on a single score for the final classification. Fig. 8 depicts a representation of this pipeline.

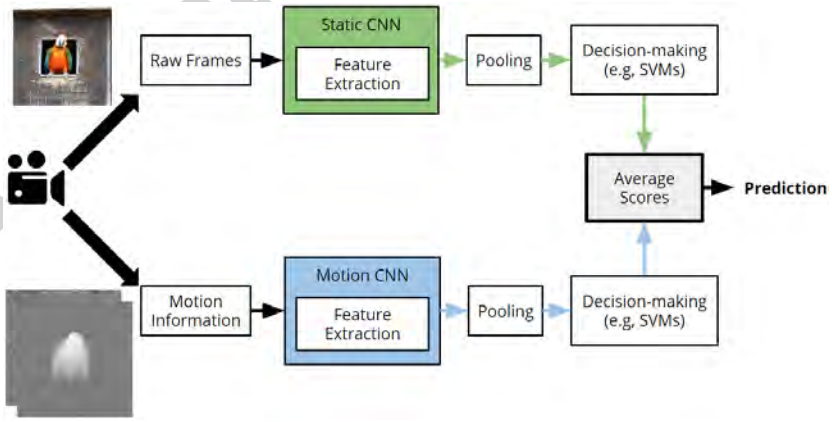


Figure 8: Pipeline for the late-fusion scheme. The information is combined at the end, after each classifier (e.g., SVM) produces a prediction score, by averaging the scores for the final classification.

4.4. Architecture Specifications

The convolutional neural network architecture we adopt for the experiments was proposed in [21], and is referred to as GoogLeNet. This architecture was employed for all types of data: Static (raw frames), Motion (optical flow and motion vectors) and Static-Motion (early fusion).

For feature extraction, we pick the output from the last layer — fully-connected (FC) — before the final classification. Indeed, for other CNN architectures, it is common to utilize as features not only the output from the last FC layer, but also from other layers before it (earlier layers). However this only makes sense if those lower layers are also FC, which are able to capture the patterns from the low-level features of the convolutional, or pooling, layers bellow it, and output mid-level features. With GoogLeNet, between the convolutional layers and the final classification layer, there is only one FC layer.

Although GoogLeNet architecture contains other FC layers, they are associated with auxiliary classifiers in branches at the middle of the network, being located too early in the network. Thus, these layers are still too close to the raw data, containing too low-level information that may mislead a high-level classifier later on such as SVM. Moreover, as described by Szegedy et al. [21], these auxiliary classifiers are only used during the learning strategy, by adding a low weighted amount to the gradient loss, but they are not used at inference time as they often do not contribute to the final decision-making process. Given these observations and previous studies in the literature on this matter, for our experiments, we have opted to use as features only the output from the last FC layer. The output from this layer has a dimensionality of 1,024-d.

The motivation for exploring this CNN model as a feature extractor, comes from the fact that GoogLeNet was the winner of ImageNet 2014 Challenge [47], achieving a striking 6.67 top-5 error rate in the object classification competition.

The ImageNet training dataset comprises about 1.2 million images, containing 1,000 classes with a wide range of subjects, from plants and animals to persons, sports, and objects. Thus it is expected that GoogLeNet architecture has the capability of learning to extract highly discriminative visual features from input images, although not initially fine-tuned to the problem of interest in this work. It is also expected that a model pre-trained with ImageNet 2014 dataset should hold an advanced state of optimization for image feature extraction, which may be useful for application on pornography detection, by itself, or by generating a custom-tailored model with weights fine-tuned to our problem.

To adapt the weights to our particular 2-class detection problem, taking as input the static and motion data of interest, we extend upon the initial architecture to map the last layer with 1,000 filter outcomes (which is the number of classes in the ImageNet classification problem) to two (porn vs. non-porn). In addition, all the network weights, except within Early Fusion, are fine-tuned to the problem of interest herein via backpropagation, initializing the weights with the values learned on the ImageNet 1.2 million images.

5. Experiments and Results

We now discuss the experimental setup, including: the dataset; evaluation metrics; training specifications; and details on the existing methods in the literature as well as third-party solutions (Sec. 5.1). Next, we present the experiments and obtained results, comparing the proposed method with the existing ones in the literature, and third-party solutions (Sec. 5.2).

5.1. Experimental Setup

In the next subsections, we present the experimental setup designed for the evaluation of the proposed methods.

5.1.1. Datasets

We adopted two datasets in our experiments: Pornography-800 [14] and Pornography-2k [16]. As a matter of fact, Pornography-2k is an extension of Pornography-800. Therefore, we opted to report all the experiments with the proposed methods on the Pornography-2k (more complete and challenging), along with the methods we choose as baselines. Finally, we evaluate our best proposed approaches on Pornography-800, for direct comparisons with existing work in the literature.

Pornography-800 dataset

This dataset[1] was originally proposed in [48] and is distributed upon acceptance of a user agreement. It comprises approximately 80 hours, spanning 800 videos, 400 pornographic and 400 non-pornographic.

The videos with pornography content were acquired from websites specialized on that type of content, searching for samples within a wide range of genres and with actors from distinct ethnicities (e.g, Asians, Blacks, Caucasian).

With respect to non-pornographic content, general-public purpose video networks were considered[2] for acquiring the videos. The dataset contains two levels of difficulty, *easy* and *difficult*. The former comprises videos randomly selected from various websites, while the latter considers videos gathered through textual queries containing words such as "wrestling", "sumo", "swimming", "beach", etc. (i.e., words associated to skin exposure).

The official evaluation protocol for this dataset is the 5-fold cross-validation (640 videos for training and 160 for testing on each fold).

Pornography-2k dataset

The Pornography-2k dataset [16] is an extended version of the Pornography-800 dataset [48]. The new dataset comprises nearly 140 hours of 1,000 pornographic and 1,000 non-pornographic videos, varying from six seconds to 33 minutes long.

[1] https://sites.google.com/site/pornographydatabase/

[2] YouTube (www.youtube.com), Vimeo (vimeo.com) and Vine (vine.co)

The non-pornographic videos were acquired similarly to Pornography-800, targeting at both *easy* and *difficult* samples. Concerning the pornographic material, differently than Pornography-800, it is not restricted to pornography-specialized websites. Instead, it was also explored general-public purpose video networks, 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. Fig. 9 depicts some example frames from the Pornography-2k dataset. This dataset is available publicly [16].

Figure 9: Representative sample frames from Pornography-2k dataset videos. Image adapted from Avila et al. [14], with added samples. Note that the dataset is very challenging, with a variety of pornography styles (e.g., hentai vs. live-action) and difficult non-pornographic cases with a lot of skin exposure.

For the validation protocol, as suggested by Moreira et al. [16], we apply a 5×2-fold cross-validation protocol [49], which consists of randomly splitting the dataset five times into two folds, balanced by class. In each round of experiments, training and testing sets are switched and, consequently, 10 analyses are conducted for each considered method.

5.1.2. Evaluation Metrics

As evaluation metrics, we adopt the default evaluation metrics of the Pornography-2k dataset: the video classification *accuracy* (ACC) and the F_2 *measure* (F_2), both averaged in all experimental folds.

ACC is simply the percentage of correctly classified videos. F_2, in turn, is the weighted harmonic mean of precision and recall, which gives twice the weight to recall ($\beta = 2$) than precision. In the case of pornography filtering, the F_2 measure is crucial because false negative results are harmful, allowing one to be exposed to pornographic content. It is thus less prejudicial to wrongly deny the access to non-pornographic material, than to wrongly disclose pornographic content. F_β measure is defined as:

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

where β is a parameter denoting the importance of recall compared to precision.

The mean over the 5×2 folds from the evaluation metrics, ACC and F_2, may be insufficient to certify that a specific method is better than another, due to some large variations in the population of measures that may be hidden during averaging. To overcome this, we employ a non-parametric Wilcoxon signed-rank test [50], which is a paired difference test that allows us to quantify how different two populations are; in this case, the populations are sampled from each fold measure from each method, without assuming a normal distribution of the population. Therefore, we can confirm more confidently whether or not that two methods are statistically different from one another. Two methods are considered statistically different if their Wilcoxon's returned p-value is lower than 0.05 (95% confidence test).

For the Pornography-800 dataset, we only report the mean video classification *accuracy*, following the default evaluation metric of the dataset. Since we do not have the result by fold from the related works, we could not employ Wilcoxon signed-rank test [50].

5.1.3. Proposed Method's Setup

The focus of the proposed methodology is to classify whole videos as porn or non-porn. Videos are a collection of frames, but using all of them for video classification would demand a great computational effort. However, this experiment can be turned more manageable by using a sampling of the frames, and still maintain consistent effectiveness for comparison between the distinct methods proposed. With this established requirements, we opted for using a frame sampling rate of one frame per second (1fps). For the motion information, this frame sampling dictates from which frames this type of information will be extracted.

The same sampling was used for both the training and test phases. During training, the frames (or motion image representations) were utilized separately, for learning the CNN models, and pooled by video after feature extraction, for learning the classification model. During testing, they pass directly to feature extraction by the trained CNN model.

All CNNs were fine-tuned starting with the weights from ImageNet, except Static-Motion CNN for the early fusion color variation. Static-Motion CNN for color variation is trained from scratch, because its input contains 5 channels, and therefore we could not start with the the same filter configuration and weights from ImageNet, which contains 3-channel kernels.

The training of the CNN model was performed with the `Caffe` framework [51]. We picked the polynomial learning rate decay policy, because the GoogLeNet ImageNet model we considered, from `Caffe`, was trained much faster using this policy. For each type or source of information (static and/or motion-based), we picked a suitable value for the base learning rate, weight decay, polynomial power and the number of epochs to run. In the following, we overview these hyperparameters.

In a nutshell, *learning rate* is the value that determines how much the network will learn at each iteration. As a matter of fact, the learning rate can be seen as a weight that controls the rate at which the values of the filter banks will be updated during backpropagation. The learning rate is initialized to the value of this hyperparameter, also known as base learning rate, then it will decay according to a chosen policy, in our case, by the polynomial learning rate decay policy.

The *weight decay* hyperparameter controls what will be the load of the regularization technique, sometimes referred to as ℓ_1/ℓ_2 regularization, while computing the cost function during backpropagation. Regularization techniques are employed for preventing overfitting, and weight decay achieves this by penalizing filters with large weights.

Polynomial power is the value used for determining what will be the decay of the learning rate, given the polynomial learning rate decay policy. At each iteration, a new learning rate is computed using the following formula:

$$learning\ rate = base\ learning\ rate \times (1 - \frac{current\ iteration}{max\ iteration})^{polynomial\ power}.$$

An *epoch* is the number of times of all train samples are used once to update the weights. In fact the number of epochs is converted to number of iterations, according to the number of iterations necessary for an epoch.

Table 2 shows the exact values for these hyperparameters. It is important to mention that such hyperparameters have no associated units. Also, a dropout layer, with 40% ratio of dropped outputs, was maintained from the original GoogLeNet architecture.

Table 2: Learning hyperparameters used for training the architecture used in this work.

	Learning Rate	Weight Decay	Power	Max Epochs
Raw Frames	0.000009	0.005	0.5	200
Optical Flow	0.00006	0.001	0.9	200
MPEG Motion Vectors	0.0002	0.001	0.9	100
Early Fusion (Gray)	0.0002	0.001	0.9	75
Early Fusion (Color)	0.001	0.005	0.5	25

For training a CNN, another sub-split of the dataset is necessary. For that, each training fold from the 5×2-fold cross-validation was re-partitioned into actual training and validation, with a proportion of 85%/15% videos in each part.

For this problem, we did not consider data augmentation techniques while training the network model, as we could gather enough training samples due to the high quantity of frames contained in the training videos, and therefore properly optimize the fine-tuning procedure of the method.

We perform the final classification with a linear Support Vector Machine (SVM) classifier using LIBSVM [52] (version 3.18). We apply grid search to

find the best regularization C SVM parameter during training, $C \in \{2^c : c \in [-5, -3, \ldots, 15]\}$.

5.1.4. Comparison with Spatio-temporal Video Descriptors

For a better interpretation of the performance of the proposed methods, it is necessary to compare them with the current state-of-the-art spatio-temporal video descriptor: Dense Trajectories [18]. This method relies on a dense sampling of descriptors, not only spatially, at feature points in the starting frame, but also temporally, by tracking the feature points in the subsequent frames. We extract the dense trajectories from the video files using the source-code provided by Wang et al [18], with default values.

More recently, Moreira et al. [16] proposed the Temporal Robust Features (TroF), a fast spatio-temporal interest point detector and descriptor, which is directly inspired by the still-image Speeded-Up Robust Features (SURF) [53]. TRoF relies on three major extensions of the original method to use the video space-time. We extract the TRoF descriptors, with default values.

As mid-level representation, for Dense Trajectories and TRoF descriptors, we extract Fisher Vectors [54], the state-of-the-art model of bags of visual words [55, 56]. To obtain the visual codebook, the Gaussian mixture model parameters are trained over 10 million descriptions, randomly sampled (half of the descriptions sampled from positive videos, and half from negative ones in the training set), using an expectation maximization algorithm. We followed the default configuration of 256 Gaussians.

As with the feature vector from the CNNs, the mid-level descriptions generated with the Fisher Vectors can also be temporally pooled to form a single feature vector for the whole video. Finally, this information is fed to an SVM for label prediction. We perform the classification with SVM classifiers using the LIBLINEAR library [57] (version 1.94). We apply grid search to find the best regularization C SVM parameter during training, $C \in \{2^c : c \in [-5, -3, \ldots, 15]\}$.

5.1.5. Comparison with Third-party Solutions

We also compare the proposed methods with some third-party solutions readily accessible. We selected the most recent ones, that rely exclusively on visual data: MediaDetective [58], Snitch Plus [59], PornSeer Pro [60], and NuDetective [61].

For MediaDetective and Snitch Plus, the video are rated according to their potential (i.e., probability) for pornography. In those cases, we tag a video as pornographic if such probability is equal to 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).

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

As these solutions do not demand a training phase, they are executed directly at the dataset, without the need for training for each fold. Even so, the reported

metrics are the average over all 5×2 folds, for fair comparison with the other methods.

5.1.6. Comparison with Existing Methods on the Pornography-800 Dataset

After evaluating the proposed methods with the Pornography-2k dataset, we turn our attention to evaluating the best methods with the Pornography-800 dataset. We do this for comparison with previous work that adopted this dataset: Avila et al. [14, 48], Valle et al. [15], Moreira et al. [16], Moustafa [22], Caetano et al. [39, 62], and Souza et al. [63].

Avila et al. [48], which is the work that introduces the Pornography-800 dataset, employed a HueSIFT descriptor at a regular grid of interest points for obtaining the low-level features; k-means, for construction of the codebook, with BOSSA — their proposed extension to BoVW — for the mid-level; and a non-linear SVM for the final classification. In turn, Valle et al. [15] evaluated the spatio-temporal descriptor STIP [17] with a standard BoVW, with random sampling for construction of the codebook, and a linear SVM for the decision making. Following a similar path, Souza et al. [63] used a traditional BoVW, with random sampling, and a linear SVM, but with ColorSTIP for low-level description.

Continuing their previous work [48], Avila et al. [14] proposed an extension upon BOSSA, named BossaNova, maintaining the use of HueSIFT, k-means, and a non-linear SVM for decision making. Aiming at more efficient descriptors, Caetano et al. [62] experimented with binary descriptors, of which BinBoost had the best performance, replacing the HueSIFT in the pipeline from Avila et al. [14]. In [39], Caetano et al. improved the classification performance by proposing an extension to the BossaNova approach, named BossaNova Video Descriptor, with binary descriptors. Moreira et al. [16] introduced the spatio-temporal detector and descriptor TRoF and aggregated local information into Fisher Vectors [54].

Differently from previous approaches, Moustafa [22] did not use a BoVW-based method, instead, the author relied upon a CNN for the low- and mid-level representations and also for classification. The classification was given by a majority voting among the video frames. Their best results were obtained with a max fusion of scores from different CNN models, pre-trained with the ImageNet dataset and with fine-tuning of the last layer of the network using the Pornography-800 dataset.

5.2. Experimental Results

In this section, we present and discuss the obtained results from the outlined experiments. First, we assess the approaches we have proposed. Afterwards, we compare our best proposed approach to methods from the literature and third-party solutions.

5.2.1. Proposed Approaches

In Table 3, we show the obtained video classification accuracy and F_2 measure for each approach we have proposed, considering the static and motion

information as well as the fusion of different methods. In these experiments, we adopted the Pornography-2k dataset (c.f., Sec. 5.1.1 for details).

Table 3: Video classification *accuracy* and the F_2 measure, averaged over the 5×2 experimental folds, for the proposed approaches on the Pornography-2k dataset. The methods are subdivided in static, motion and fusion modalities. Fusion is performed with the fine-tuned model for static information, and with both motion sources, optical flow (OF) and MPEG motion vectors (MV), except for the early fusion, which, due to its inferior performance with OF, is not employed with MV.

	Proposed Approach		ACC (%)	F_2 (%)
Static	ImageNet		94.6	95.1
	Fine-tuned*		**96.0**	**96.1**
Motion	Optical Flow		**94.4**	**95.3**
	MPEG Motion Vectors		91.0	92.0
Fusion	Early Fusion – Gray	OF	95.5	96.0
	Early Fusion – Color	OF	90.5	90.7
	Mid-level Fusion	OF	96.3	96.8
	Late Fusion*	OF	**96.4**	**96.7**
	Mid-level Fusion	MV	96.4	96.5
	Late Fusion	MV	96.4	96.6

ACC: accuracy — F_2: F_2 measure — *Fine-tuned and Late Fusion are statistically different (p-values: ACC ≈ 0.03; $F_2 \approx 0.01$) — All standard deviations are smaller than 0.02.

In the static stream, we show that the model relying on the GoogLeNet architecture trained with ImageNet data yields an impressive performance of 94.6% ACC and 95.1% F_2. These results are further improved upon by fine-tuning the network weights with the pornographic data, thus specializing the network to the problem of interest, reaching 96.0% ACC and 96.1% F_2, a 1.5 percentage point improvement in ACC (26% error reduction) and 1 percentage point in F_2.

When considering the motion information, optical flow (OF) by itself yielded a performance close to the static model. Meanwhile, the MPEG motion vectors (MV) led to a lower performance, of 91.0% ACC and 92.0% F_2. This difference in performance between these two sources of motion information may be explained by the fact that the MV represents the motion of a macroblock of pixels, which is a much lesser fine-grained description form than OF, which takes into account the motion information for each pixel.

Despite the lower performance of the motion information alone, when we combine it with the static information from the fine-tuned network (pornography-specialized network), by mid-level fusion and late fusion, we improve the ACC and F_2 results. Both early fusion variations, Gray and Color, yielded a lower performance than using the fine-tuned static information by itself. Perhaps it is better to specialize the network to a single type of information, leaving the fusion to a higher level. Another reason might be related to

the architecture considered in this work, GoogLeNet. It may not be appropriate for processing five channels or combining static and motion right at the lowest level (e.g., raw data) of the network, demanding some customization such as increasing the number of filters or processing each information independently at the first layers.

We believe that the better performance from the gray variation over color, comes from the fact that we could fine-tune its model using the ImageNet model and that the 3-channel input data is more appropriate for the GoogLeNet architecture. However, we expect that if these issues were overcome (e.g., by training an appropriate architecture with a large collection of samples), the full potential from using all color channels could be reached, outperforming the gray-only variation of this fusion, and perhaps the other fusion approaches, mid-level and late.

Given the low performance of early fusion, and its costly requirements for training, we have opted for not fusing MPEG motion vectors this way.

Mid-level fusion and late fusion, on the other hand, apparently could better combine static and motion information, surpassing the performance of the fine-tuned network alone. Surprisingly, this happened even while combining with MV, showing that, although it had a worse performance when used alone, its complementarity to the static information is still advantageous. In addition, another advantage of using the MVs is that they are readily available during decoding of the video. Still, even that by a small margin, late fusion with OF obtained the best combination of results for ACC and F_2 measures.

In fact, our architecture was able to properly learn effective features from the motion data, as our results with middle- and late-fusion approaches showed, which take into account the information provided by the *Static Raw Frames* and *Optical Flows* simultaneously. However, it is possible that using an innate motion-based network could equally produce good results; however such network could be more complex (with more weights) than the one we have extended upon.

Moreover, unfortunately, there is no motion CNN model readily available, as far as we know, that has an input in accordance with the pipeline we proposed here, with a single motion image representation per time. Current available motion CNNs often require stacked motions and thus, are also not amenable to fast implementation and deployment in mobile devices, which is our ultimate goal in this research.

In spite of all of this, we have performed some experiments with other traditional CNNs, AlexNet [19] and VGG [64] networks, also taking, as features, the output from their last FC layers before classification. Table 4 shows the results from the experiments, which consisted in evaluating how each architecture would perform for the static information, with the ImageNet model and with the fine-tuned model from the ImageNet weights, and also for the motion information (optical flows) after fine-tuning, again from the ImageNet weights.

For all considered architectures, if we use the ImageNet weights for feature extraction (Static – ImageNet), the performance for the static information is almost equivalent, with not more than 0.3 perceptual points of difference between

Table 4: Video classification *accuracy* and the F_2 measure, over the **first fold** from the 5×2 experimental folds, for the chosen architectures on the Pornography-2k dataset. The chosen architectures are GoogLeNet, AlexNet and VGG, they are evaluated within three different setups: 1) Static data with ImageNet weights; 2) Static data with model fine-tuned from the ImageNet weights; and 3) Motion data (Optical Flows), also with model fine-tuned from the ImageNet weights.

	Static – ImageNet		Static – Fine-tuned		Motion – Fine-tuned	
CNN Architecture	ACC (%)	F_2 (%)	ACC (%)	F_2 (%)	ACC (%)	F_2 (%)
GoogLeNet [21]	94.7	95.4	95.9	95.5	94.5	93.1
AlexNet [19]	94.9	94.6	95.0	94.4	93.4	93.7
VGG [64]	94.6	95.2	95.9	95.3	95.7	96.1

the best accuracy and the worst.

After fine-tuning the models, initializing the networks with the ImageNet weights, there is a considerable improvement in performance for VGG and GoogLeNet networks, while maintaining equivalency between one another, over the results with no fine-tuning and over AlexNet.

When dealing with the motion information (Motion – Fine-tuned), VGG showed slightly better results when in comparison to GoogLeNet and AlexNet. The fact that VGG had equivalent performance to GoogLeNet for static information, but superior for motion, supports the suspicion that motion information does indeed have a particular structure, even after being represented as images, that some CNN architectures are better to capture. Therefore, an architecture specialized only in motion information could improve even more the results.

Finally, it is important to highlight that the choice for one architecture should not be based only on the classification numbers. In our case, as our ultimate goal is to implement our solution in a mobile device with more limited resources than a traditional server, the overall configuration and size of a network also matters. In this regard, for instance, GoogLeNet is superior than VGG as the former has a learned model with only 40MBs against a learned model of 533MBs of VGG, an order of magnitude of difference and a game-changer when we aim at a mobile implementation. In addition to the obvious impact when testing a new video, this difference also plays an important role during training as it has a huge influence on the batch size we can use for training, and consequently, the speed of training. For GoogLeNet we used a batch size of 96, and could have even used a bigger one. For VGG, it was 64.

5.2.2. Comparison with Existing Methods using the Pornography-2k Dataset

For a better evaluation of the proposed approaches that obtained the best results in each modality (c.f., Sec. 5.2.1), we compare them with the current state-of-the-art spatio-temporal video description and third-party solutions. Table 5 shows the respective video classification accuracy and F_2 measure of the considered methods. Note that the best proposed methods outperform most of the existing solutions.

The third-party solutions, which heavily depend on skin detection and do not

Table 5: Results on the Pornography-2k dataset for the third-party solutions, the current state-of-the-art spatio-temporal video description, and the best approaches we have proposed in each modality (Static – Fine-tuned; Motion – Optical Flow; Late Fusion with Optical Flow). We report the average performance on 5 × 2 folds.

	Solution	ACC (%)	F_2 (%)
Third-party	Snitch Plus [59]	66.6	46.4
	MediaDetective [58]	71.9	66.5
	NuDetective [61]	72.6	62.9
	PornSeer Pro [60]	79.1	75.6
BoVW-based	Dense Trajectories [18]*†	95.8	95.6
	TRoF [16]$^{\flat\ddagger}$	95.6	95.3
Proposed Approaches	Static – Fine-tuned	96.0	96.1
	Motion – Optical Flow	94.4	95.3
	Late Fusion (OF)	**96.4**	**96.7**

ACC: accuracy — F_2: F_2 measure — **Dense Trajectories* and *Late Fusion (OF)* are statistically different (p-values: ACC ≈ 0.028; F_2 ≈ 0.001) — †*Dense Trajectories* and *Static – Fine-tuned* are not statistically different in ACC, but are in F_2 (p-values: ACC ≈ 0.239; F_2 ≈ 0.037) — $^{\flat}$*TRoF* and *Late Fusion (OF)* are statistically different (p-values: ACC ≈ 0.014; F_2 ≈ 0.009) — ‡*TRoF* and *Static – Fine-tuned* are not statistically different in ACC, but are in F_2 (p-values: ACC ≈ 0.202; F_2 ≈ 0.037) — All standard deviations are smaller than 0.1.

take advantage of the space-time information, have shown a poor performance. PornSeer Pro [60] obtained the best ACC and F_2 measures among them, with 79.1% and 75.6% respectively, far below the performance using the solutions in the literature and our proposed approaches.

The proposed methods also outperform the Dense Trajectories method [18]. For instance, the spatio-temporal approach, Late Fusion (OF), outperforms Dense Trajectories by a margin of 0.5 percentage point in ACC (14.3% error reduction) and over 1.0 in F_2 measure.

Also, we can assert that motion feature plays an important role in pornography video detection when comparing the motion-based approaches (Dense Trajectories and proposed approaches) with the third-party solutions. The motion-based approaches remarkably outperform the third-party solutions.

5.2.3. Comparison with Existing Methods using the Pornography-800 Dataset

In Table 6, we compare our best proposed approaches with the reported results from other methods in the literature using the Pornography-800 dataset.

The proposed approaches significantly outperform the existing BoVW-based methods [14–16, 48, 62, 63], by 3–11 percentage points. The proposed methods also outperform, by almost four percentage points, the results reported in Moustafa [22], which also use Deep Learning. In this case, the error reduction was over 64%. Even though we could not apply Wilcoxon's test, given the large perceptual difference in accuracy between the related works and our best approaches, with smaller standard deviation in some cases, we believe that the

Table 6: Results on the Pornography-800 dataset considering the best approaches we have proposed in each modality (Static – Fine-tuned; Motion – Optical Flow; Mid-level and Late Fusion with Optical Flow) and existing methods in the literature. We report the average performance and standard deviations using the dataset's 5-fold evaluation protocol. NA stands for a non-reported information in the original work.

	Solution	ACC (%)
BoVW-based	Avila et al. [48]	87.1 ± 2.0
	Valle et al. [15]	$91.9 \pm$ NA
	Souza et al. [63]	$91.0 \pm$ NA
	Avila et al. [14]	89.5 ± 1.0
	Caetano et al. [62]	90.9 ± 1.0
	Caetano et al. [39]	92.4 ± 2.0
	Moreira et al. [16]	95.0 ± 1.3
CNN	Moustafa [22]	94.1 ± 2.0
Proposed Approaches	Static – Fine-tuned	97.0 ± 2.0
	Motion – Optical Flow	95.8 ± 2.0
	Mid-level Fusion (OF)	$\mathbf{97.9 \pm 0.7}$
	Late Fusion (OF)	$\mathbf{97.9 \pm 1.5}$

results would probably be statistically significant.

Although Moustafa [22] employs the same architecture we use in this work, GoogLeNet, there are critical differences, thus leading to the important difference in performance, we report herein: first of all, he only fine-tuned the network last layer, while in our work we fine-tuned all layers, creating a network model specialized to the problem of interest; second, the network output in that work, for each frame, was used in a majority voting scheme for classifying the video, while, in turn, we have opted for using the network as a feature extractor, pooling the frame descriptions, then feeding them to an classifier for the video classification; finally, that work only considered static information, meanwhile our methods rely upon static and motion information, as well as on effective methods for combining them.

6. Conclusions and Future Work

The evaluation of our techniques, shows that the association of Deep Learning with the combined use of static and motion information, considerably improves pornography detection. Not only over current scientific state of the art [14, 15, 22, 62], but also over off-the-shelf software solutions [58–61]. Our solution also proves to be superior to general-purpose action recognition features [18], when applied to pornography detection.

The Deep Learning solution using only static information is already competitive with state-of-the-art action recognition features, Dense Trajectories [18],

reaching an error rate of 4%, which is low for such a subjective problem as pornography. For further reducing the error rate, we believe the focus should be on the motion information: by adjusting the CNN, adapting the architecture, boosting the model with more training samples, or improving static-dynamic information fusion.

Besides improving whole-video classification, we are interested in applying our techniques to the harder task of locating in time the pornographic content within the video. To reach that goal, we are currently annotating the Pornography-2k video dataset at frame level. The main motivation for that harder task is filtering pornography in real time, an important goal for video streaming, camera-surveillance systems, or surveillance of video chats for certain publics.

Finally, in addition to adapting our current methods for the localization problem (e.g., [65, 66]), another aspect worth exploring is to integrate them to the so called Long Short Term Memory (LSTM) networks. LSTMs are a model of Recurrent Neural Network (RNN) that captures the sequential information of the input data, a highly desirable feature for classification of videos. The LSTM architecture could be used to process the CNN extracted features, using the proposed methods in this work, from a fixed number of frames, improving the real-time classification.

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 from CNPq, FAPESP (Grant #2015/19222-9) through the DéjàVu project, and CAPES, through the DeepEyes project. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

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, Cyberpsychology, Behavior, and Social Networking 15 (1) (2012) 13–23.

[2] M. Fleck, D. Forsyth, C. Bregler, Finding naked people, in: European Conference on Computer Vision (ECCV), Vol. 1065, 1996, pp. 593–602.

[3] D. Forsyth, M. Fleck, Identifying nude pictures, in: IEEE Workshop on Applications of Computer Vision, 1996, pp. 103–108.

[4] D. Forsyth, M. Fleck, Automatic detection of human nudes, International Journal of Computer Vision (IJCV) 32 (1) (1999) 63–77.

[5] H. Zheng, M. Daoudi, B. Jedynak, Blocking Adult Images Based on Statistical Skin Detection, Electronic Letters on Computer Vision and Image Analysis (ELCVIA) (2004) 1–14.

[6] M. Jones, J. Rehg, Statistical color models with application to skin detection, International Journal of Computer Vision (IJCV) 46 (1) (2002) 81–96.

[7] H. Rowley, Y. Jing, S. Baluja, Large scale image-based adult-content filtering, in: International Conference on Computer Vision Theory and Applications (VISAPP), 2006, pp. 290–296.

[8] S. Lee, W. Shim, S. Kim, Hierarchical system for objectionable video detection, IEEE Transactions on Consumer Electronics 55 (2) (2009) 677–684.

[9] H. Bouirouga, S. El Fkihi, A. Jilbab, D. Aboutajdine, Skin detection in pornographic videos using threshold technique, Journal of Theoretical and Applied Information Technology 35 (1) (2012) 7–19.

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

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

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

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

[14] S. Avila, N. Thome, M. Cord, E. Valle, A. Araújo, Pooling in image representation: The visual codeword point of view, Elsevier Computer Vision and Image Understanding (CVIU) 117 (5) (2013) 453–465.

[15] E. Valle, S. Avila, A. da Luz Jr., 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, pp. 625–638.

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

[17] I. Laptev, On Space-Time Interest Points, International Journal of Computer Vision (IJCV) 64 (2-3) (2005) 107–123.

[18] H. Wang, C. Schmid, Action Recognition with Improved Trajectories, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 3551–3558.

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

[20] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems (NIPS), 2014, pp. 568–576.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[22] M. Moustafa, Applying deep learning to classify pornographic images and videos, in: 7th Pacific-Rim Symposium on Image and Video Technology (PSIVT), 2015.

[23] Q. Le, W. Zou, S. Yeung, A. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3361–3368.

[24] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 35 (1) (2013) 221–231.

[25] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1725–1732.

[26] I. E. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia, John Wiley & Sons, Inc., 2004.

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

[28] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior Recognition via Sparse Spatio-Temporal Features, in: IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72.

[29] C. Harris, M. Stephens, A Combined Corner and Edge Detector, in: Alvey Vision Conference, 1988, pp. 189–192.

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

[31] 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, pp. 106–114.

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

[33] A. Lopes, S. Avila, A. Peixoto, R. S. Oliveira, M. Coelho, A. Araújo, A bag-of-features approach based on Hue-SIFT descriptor for nude detection, in: European Signal Processing Conference (EUSIPCO), 2009, pp. 1552–1556.

[34] H. Zuo, W. Hu, O. Wu, Patch-Based Skin Color Detection and Its Application to Pornography Image Filtering, International Conference on World Wide Web (WWW) (2010) 1227–1228.

[35] A. Zaidan, N. Ahmad, H. Abdul Karim, M. Larbani, B. Zaidan, A. Sali, On the multi-agent learning neural and Bayesian methods in skin detector and pornography classifier: An automated anti-pornography system, Elsevier Neurocomputing 131 (2014) 397–418.

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

[37] F. Nian, T. Li, Y. Wang, M. Xu, J. Wu, Pornographic Image Detection Utilizing Deep Convolutional Neural Networks, Elsevier Neurocomputing 120 (2016) 283–293.

[38] 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, pp. 49–54.

[39] C. Caetano, S. Avila, W. R. Schwartz, S. J. F. Guimarães, A. Araújo, A mid-level video representation based on binary descriptors: A case study for pornography detection, Elsevier Neurocomputing 213 (2016) 102–114.

[40] A. Ulges, C. Schulze, D. Borth, A. Stahl, Pornography detection in video benefits (a lot) from a multi-modal approach, in: ACM International Workshop on Audio and Multimedia Methods for Large-Scale Video Analysis, 2012, pp. 21–26.

[41] M. Goodale, D. Milner, Separate visual pathways for perception and action, Elsevier Trends in Neurosciences 15 (1) (1992) 20–25.

[42] K. Soomro, A. R. Zamir, M. Shah, UCF101: A Dataset of 101 Human Actions Classes From Videos in the Wild, Tech. rep., CRCV-TR-12-01 (2012).

[43] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: A large video database for human motion recognition, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2556–2563.

[44] B. Horn, B. Schunck, Determining optical flow, in: International Society for Optics and Photonics Technical Symposium East, 1981, pp. 319–331.

[45] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High Accuracy Optical Flow Estimation Based on a Theory for Warping, in: European Conference on Computer Vision (ECCV), 2004, pp. 25–36.

[46] A. Grange, P. de Rivaz, J. Hunt, VP9 Bitstream & Decoding Process Specification, http://www.webmproject.org/vp9/#draft-vp9-bitstream-and-decoding-process-specification (2016).

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252.

[48] 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, pp. 2909–2912.

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

[50] F. Wilcoxon, Individual Comparisons by Ranking Methods, Biometrics Bulletin 1 (6) (1945) 80–83.

[51] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM International Conference on Multimedia (MM), 2014, pp. 675–678.

[52] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST) 2 (2011) 1–27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[53] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, SURF: Speeded up robust features, Computer Vision and Image Understanding (CVIU) 110 (3) (2008) 346–359.

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

[55] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: Theory and practice, International Journal of Computer Vision (IJCV) 105 (3) (2013) 222–245.

[56] 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, pp. 1–12.

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

[58] Media Detective, http://mediadetective.com/.

[59] Snitch Plus, http://www.hyperdynesoftware.com/.

[60] PornSeer Pro, http://www.yangsky.com/products/dshowseer/porndetection/PornSeePro.

[61] 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, pp. 349–353.

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

[63] 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, pp. 31–36.

[64] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint arXiv:1409.1556 (2014) 1–10.

[65] X. Chang, Y. Yang, E. P. Xing, Y.-l. Yu, Complex event detection using semantic saliency and nearly-isotonic SVM, in: ACM International Conference on Machine Learning (ICML), 2015, pp. 1348–1357.

[66] X. Chang, Y. Yang, E. P. Xing, Y.-l. Yu, Searching persuasively: Joint event detection and evidence recounting with limited supervision, in: ACM Conference on Multimedia (MM), 2015, pp. 581–590.