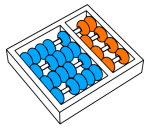
Universidade Estadual de Campinas
Instituto de Computação



Daniel Henriques Moreira

Sensitive-Video Analysis

Análise de Vídeo Sensível

CAMPINAS
2016

Daniel Henriques Moreira

Sensitive-Video Analysis

Análise de Vídeo Sensível

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Anderson de Rezende Rocha
Co-supervisor/Coorientador: Prof. Dr. Siome Klein Goldenstein

Este exemplar corresponde à versão final da Tese defendida por Daniel Henriques Moreira e orientada pelo Prof. Dr. Anderson de Rezende Rocha.

CAMPINAS
2016

Agência(s) de fomento e nº(s) de processo(s): CAPES, 1572763; CAPES, 1197473

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

M813s

Moreira, Daniel Henriques, 1983-
Sensitive-video analysis / Daniel Henriques Moreira. – Campinas, SP : [s.n.], 2016.

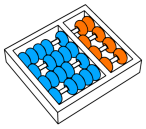
Orientador: Anderson de Rezende Rocha.
Coorientador: Siome Klein Goldenstein.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Visão por computador. 2. Reconhecimento de padrões. 3. Vídeo digital - Classificação. 4. Análise de imagem. 5. Pornografia na internet. 6. Violência no cinema. I. Rocha, Anderson de Rezende,1980-. II. Goldenstein, Siome Klein,1972-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Análise de vídeo sensível
Palavras-chave em inglês:
Computer vision
Pattern recognition
Digital video - Classification
Image analysis
Internet pornography
Violence in motion pictures
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Siome Klein Goldenstein [Coorientador]
Cristina Nader Vasconcelos
William Robson Schwartz
Roberto de Alencar Lotufo
Ricardo da Silva Torres
Data de defesa: 19-07-2016
Programa de Pós-Graduação: Ciência da Computação

Universidade Estadual de Campinas
Instituto de Computação



Daniel Henriques Moreira

Sensitive-Video Analysis

Análise de Vídeo Sensível

Banca Examinadora:

- Prof. Dr. Siome Klein Goldenstein
  Universidade Estadual de Campinas
- Profa. Dra. Cristina Nader Vasconcelos
  Universidade Federal Fluminense
- Prof. Dr. William Robson Schwartz
  Universidade Federal de Minas Gerais
- Prof. Dr. Roberto de Alencar Lotufo
  Universidade Estadual de Campinas
- Prof. Dr. Ricardo da Silva Torres
  Universidade Estadual de Campinas

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 19 de julho de 2016

Aos meus pais, com carinho.
É mais fácil enfrentar águas turbulentas,
quando há a certeza de um porto seguro.

I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description "hardcore pornography" (...). But I know it when I see it...

(Potter Stewart)

Acknowledgements

Firstly, I would like to express my gratitude to my supervisors, Professors Anderson Rocha and Siome Goldenstein, for their diligent orientation. Professor Anderson, thank you for your present and continuous support, guidance, and encouragement. Professor Siome, thank you for sharing your immense experience and knowledge, and for having readily accepted to supervise me in the beginning of this journey, when I was only a stranger that had been out of Academy for more than five years.

Besides my supervisors, I would like to sincerely thank the members of my thesis committee, namely Professors Cristina Vasconcelos, William Schwartz, Roberto Lotufo, and Ricardo Torres, for their insightful comments and hard questions, which certainly helped me to improve the present work, and to widen my research perspectives.

I am really grateful to Dr. Sandra Avila, for having altruistically and competently supervised me so many times during this research. Sandra, thank you for sharing those sleepless nights before deadlines, for the partnership, and for being a true inspiration.

I thank the members of Projeto Sete: Professor Eduardo Valle, Dr. Vanessa Testoni, Daniel Moraes, and Mauricio Perez, for the stimulating discussions, commitment, and hard work. In addition, I also thank my fellow Recod labmates, for kindly sharing knowledge, books, CPU clocks, space, and time with me.

Thanks to Samsung and CAPES for the financial support, and thanks to the University of Campinas and to the Institute of Computing for offering such an excellent and fruitful academic research environment.

Last but not least, I would like to thank my friends, parents, brothers, aunts, and wife, for their ever present and absolute assistance. Special thanks go to Priscila, my wife, for sharing this journey with me, with love and patience.

From my whole heart, thank you very much for all your support!

Resumo

Vídeo sensível pode ser definido como qualquer filme capaz de oferecer ameaças à sua audiência. Representantes típicos incluem — mas não estão limitados a — pornografia, violência, abuso infantil, crueldade contra animais, etc. Hoje em dia, com o papel cada vez mais pervasivo dos dados digitais em nossa vidas, a análise de conteúdo sensível representa uma grande preocupação para representantes da lei, empresas, professores, e pais, devido aos potenciais danos que este tipo de conteúdo pode infligir a menores, estudantes, trabalhadores, etc. Não obstante, o emprego de mediadores humanos, para constantemente analisar grandes quantidades de dados sensíveis, muitas vezes leva a ocorrências de estresse e trauma, o que justifica a busca por análises assistidas por computador. Neste trabalho, nós abordamos este problema em duas frentes. Na primeira, almejamos decidir se um fluxo de vídeo apresenta ou não conteúdo sensível, à qual nos referimos como classificação de vídeo sensível. Na segunda, temos como objetivo encontrar os momentos exatos em que um fluxo começa e termina a exibição de conteúdo sensível, em nível de quadros de vídeo, à qual nos referimos como localização de conteúdo sensível. Para ambos os casos, projetamos e desenvolvemos métodos eficazes e eficientes, com baixo consumo de memória, e adequação à implantação em dispositivos móveis. Neste contexto, nós fornecemos quatro principais contribuições. A primeira é uma nova solução baseada em sacolas de palavras visuais, para a classificação eficiente de vídeos sensíveis, apoiada na análise de fenômenos temporais. A segunda é uma nova solução de fusão multimodal em alto nível semântico, para a localização de conteúdo sensível. A terceira, por sua vez, é um novo detector espaço-temporal de pontos de interesse, e descritor de conteúdo de vídeo. Finalmente, a quarta contribuição diz respeito a uma base de vídeos anotados em nível de quadro, que possui 140 horas de conteúdo pornográfico, e que é a primeira da literatura a ser adequada para a localização de pornografia. Um aspecto relevante das três primeiras contribuições é a sua natureza de generalização, no sentido de poderem ser empregadas — sem modificações no passo a passo — para a detecção de tipos diversos de conteúdos sensíveis, tais como os mencionados anteriormente. Para validação, nós escolhemos pornografia e violência — dois dos tipos mais comuns de material impróprio — como representantes de interesse, de conteúdo sensível. Nestes termos, realizamos experimentos de classificação e de localização, e reportamos resultados para ambos os tipos de conteúdo. As soluções propostas apresentam uma acurácia de 93% em classificação de pornografia, e permitem a correta localização de 91% de conteúdo pornográfico em fluxo de vídeo. Os resultados para violência também são interessantes: com as abordagens apresentadas, nós obtivemos o segundo lugar em uma competição internacional de detecção de cenas violentas. Colocando ambas em perspectiva, nós aprendemos que a detecção de pornografia é mais fácil que a de violência, abrindo várias oportunidades de pesquisa para a comunidade científica. A principal razão para tal diferença está relacionada aos níveis distintos de subjetividade que são inerentes a cada conceito. Enquanto pornografia é em geral mais explícita, violência apresenta um espectro mais amplo de possíveis manifestações.

Abstract

Sensitive video can be defined as any motion picture that may pose threats to its audience. Typical representatives include — but are not limited to — pornography, violence, child abuse, cruelty to animals, etc. Nowadays, with the ever more pervasive role of digital data in our lives, sensitive-content analysis represents a major concern to law enforcers, companies, tutors, and parents, due to the potential harm of such contents over minors, students, workers, etc. Notwithstanding, the employment of human mediators for constantly analyzing huge troves of sensitive data often leads to stress and trauma, justifying the search for computer-aided analysis. In this work, we tackle this problem in two ways. In the first one, we aim at deciding whether or not a video stream presents sensitive content, which we refer to as sensitive-video classification. In the second one, we aim at finding the exact moments a stream starts and ends displaying sensitive content, at frame level, which we refer to as sensitive-content localization. For both cases, we aim at designing and developing effective and efficient methods, with low memory footprint and suitable for deployment on mobile devices. In this vein, we provide four major contributions. The first one is a novel Bag-of-Visual-Words-based pipeline for efficient time-aware sensitive-video classification. The second is a novel high-level multimodal fusion pipeline for sensitive-content localization. The third, in turn, is a novel space-temporal video interest point detector and video content descriptor. Finally, the fourth contribution comprises a frame-level annotated 140-hour pornographic video dataset, which is the first one in the literature that is appropriate for pornography localization. An important aspect of the first three contributions is their generalization nature, in the sense that they can be employed — without step modifications — to the detection of diverse sensitive content types, such as the previously mentioned ones. For validation, we choose pornography and violence — two of the commonest types of inappropriate material — as target representatives of sensitive content. We therefore perform classification and localization experiments, and report results for both types of content. The proposed solutions present an accuracy of 93% in pornography classification, and allow the correct localization of 91% of pornographic content within a video stream. The results for violence are also compelling: with the proposed approaches, we reached second place in an international competition of violent scenes detection. Putting both in perspective, we learned that pornography detection is easier than its violence counterpart, opening several opportunities for additional investigations by the research community. The main reason for such difference is related to the distinct levels of subjectivity that are inherent to each concept. While pornography is usually more explicit, violence presents a broader spectrum of possible manifestations.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

We define *sensitive video* as any motion picture that may pose threats to its audience. Typical representatives include — but are not limited to — video pornography and scenes depicting violence.

Taking into account the easiness and the multitude of ways to produce, share, and send video streams over the Internet, it becomes clear that the diversity of content is untold. Within such diversity, it is not hard to imagine that some streams may be sensitive, due to inadequate audience, sex appeal, religious or cultural offensiveness. The reasons for the diffusion of such material may be related to negligence (e.g., people who are not aware of sharing their personal files), protest (e.g., the topless-based activism of Femen[1]), or even malice (e.g., pedophiles).

Sensitive content is alarming because it may be really harmful (e.g., violent media contribute to aggressive behavior in children, and desensitization to brutality [26]), and even illegal (e.g., child pornography [50]). Hence, there is a need for regulating its use over the Internet. However, the employment of human operators for constantly analyzing tons of sensitive streams often leads to stress and trauma [9], justifying the search for computer-aided analysis, for alleviating the job of moderators.

Notwithstanding, the automatic detection of sensitive video is a challenging and still open problem, mainly due to the following aspects:

Big-data nature The biggest video-sharing website on the Internet states that 300 hours of video are uploaded to its servers every minute[2]. From such number, we can have a vague yet remarkable notion of the big-data nature of the provided service. How to design more efficient solutions, for meeting such high demand?

Pervasiveness The same website estimates that hundreds of millions of hours are watched every day on its platform. From this total, more than 50% happens on mobile devices, attesting an increasingly high video pervasiveness. How to design more ubiquitous solutions, that can operate on the consumer side, even on devices with limited hardware?

[1] Cf. `http://www.femen.org`, accessed May 3rd, 2016

[2] Cf. `http://www.youtube.com/yt/press/statistics.html`, accessed May 3rd, 2016

Subjectivity The sensitiveness of some contents depends on complex cultural and social issues. For instance, while female topless is not offensive to South American indigenous societies, for some Muslims, the entire female body is considered intimate, except for the hands and the face. How to design more general solutions, that can be easily suited to a specific population, or sensitive concept?

Urgency The fast analysis of sensitive content is important in many scenarios. For instance, in forensic situations, the fast identification of inappropriate content among millions of files shall aid law enforcement by letting officers catch red-handed criminals. How to design faster yet effective solutions?

This work approaches computer-aided sensitive-video analysis, by considering the aforementioned open issues. In addition, we aim at investigating different forms of incorporating video temporal information, in a quest for more effective solutions to sensitive analysis.

1.1 Hypotheses and Goal

Given our interest in using temporal information for sensitive-video analysis, we state the following hypotheses to guide and justify the directions of this research:

H1 It is possible to efficiently use video temporal information for effective sensitive-content classification, regarding low-memory footprint[3] and small processing time[4], by combining simplified space-temporal video interest-point detection and description, with entire-footage representation through a single feature vector.

H2 It is possible to localize sensitive content within the video timeline by means of the classification and fusion of time-overlapping video snippets[5].

As one might observe, for the sake of research scope definition, we tackle the problem of sensitive-video analysis as either (i) a problem of classifying sensitive video content, or (ii) a problem of localizing sensitive content within the video timeline.

That helps us to define the goal of this research:

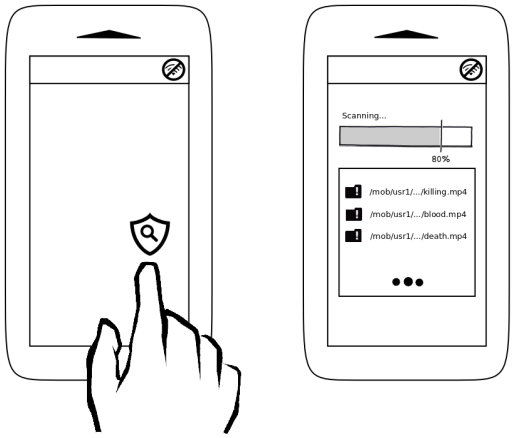
Goal Design and develop effective and efficient methods for sensitive-video classification, and for sensitive-content localization within the video timeline.

Furthermore, we choose pornography and violence — two of the commonest types of inappropriate material, specially for their relevance and negative impact on minors [81, 50, 95, 26] — as target representatives of sensitive content.

[3]Nowadays, we consider that a solution has low-memory footprint, if it, at least, is amenable to direct implementation on mobile devices, such as smartphones and tablets.

[4]Preferably close to real time.

[5]A snippet is any video excerpt.



(a) (b)

Figure 1.1: Application example of sensitive-video classification. In (a), the user activates a scanning app of sensitive content, and in (b), the app enlists the sensitive (e.g., violent) videos, with a progress bar depicting the scanning progress.

1.2 Application Examples

In this section, we define the problems of (i) sensitive-video classification, and of (ii) sensitive-content localization. Moreover, we present one application example for each problem type, in order to illustrate the utility of solving them. It is worth to mention that the application possibilities are far from being limited to the given examples.

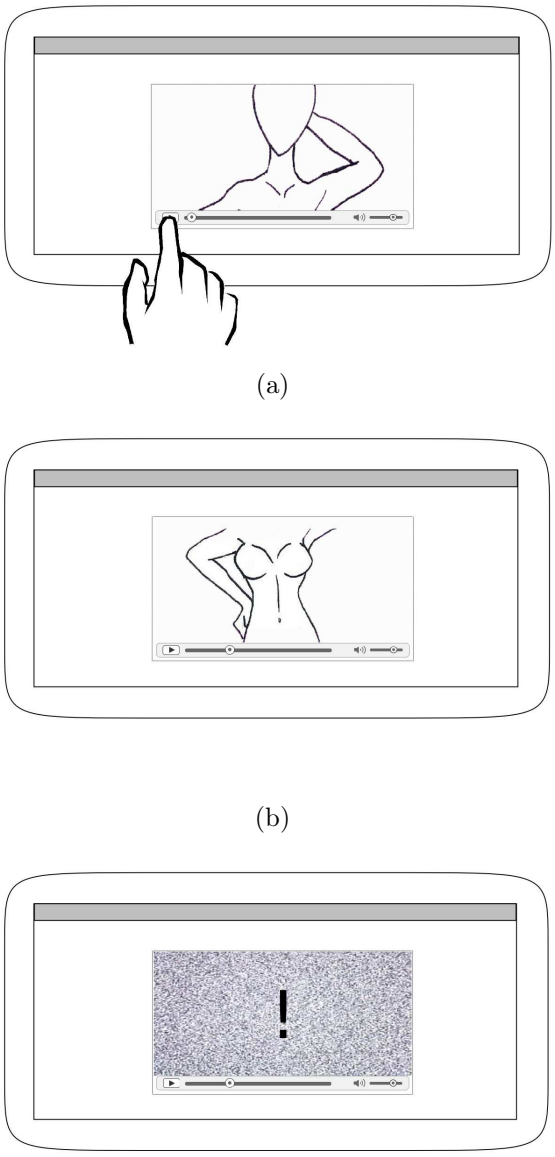
1.2.1 Sensitive-Video Classification

Sensitive-video classification is the decision problem of defining whether or not a given video stream has any occurrence of a particular target sensitive content. In other words, the related solution shall label a target stream as being representative of one of two classes: *sensitive* or *non-sensitive*.

Figure 1.1 depicts a possible application of a sensitive-video classifier. The action starts in (a), when a person (e.g., a forensic expert) activates a scanning app on a smartphone. The app finds all the video files — stored in the device — that present sensitive content (e.g., violence). In (b), the scanning progress can be checked by means of a progress bar, and the sensitive videos are iteratively enlisted. Please verify that the smartphone may stay offline during the entire process (what is shown through the offline icon, depicted in the top right corner of the device screen). It means that the classification process is performed locally, with no need of additional processing steps in external or remote machines, despite eventual memory and processing restrictions of the smartphone.

1.2.2 Sensitive-Content Localization

Sensitive-content localization is the search problem of finding sensitive scenes within a video timeline. In other words, the related system shall return the instants a video stream starts and ends displaying sensitive content.



(a)

(b)

(c)

Figure 1.2: Application example of sensitive-content localization. In (a), the user starts to play a chosen video, within a tablet, through a safe video player. In (b), the video that is being played is about to show sensitive content (pornographic). In (c), the pornographic scenes are properly censored.

Figure 1.2 depicts a possible application of a sensitive-content locator. The action starts in (a), with a person (e.g., a child) playing a chosen video, through a safe video player, which was installed in a personal tablet. In (b), the video content is about to depict sensitive (pornographic) scenes, which are properly prevented in (c), when the pornographic scenes are properly censored, according to a sensitive-scene localization process that works in the background.

1.3 Contributions

By verifying the stated hypotheses, and pursuing the aforementioned goal, this work contributes to the areas of Digital Forensics (e.g., Video Surveillance), Computer Vision (e.g., Video Content Description and Video Content Classification), and Content-Based Visual Information Retrieval (e.g., Video Content Filtering), with the following novelties:

End-to-end pipeline for efficient time-aware sensitive-video classification
Such pipeline consists of a three-level Bag-of-Visual-Words (BoVW) -inspired solution, which efficiently employs temporal information as an effective discriminative clue for the task of sensitive-content classification. It incorporates temporal information in the low and mid levels, by means of efficient local space-temporal descriptors (in terms of small processing time and low-memory footprint), and entire-footage mid-level feature pooling, respectively. It relies on Gaussian-Mixture-Models (GMM)-based codebooks, Fisher Vectors, and a linear Support Vector Machine (SVM), one of the most effective combinations that were ever reported in the BoVW-related literature. It is of general purpose, in the sense that it can be used — without step modifications — for the detection of diverse sensitive content types (e.g., gore scenes, child abuse, cruelty to animals, etc.), including our desired pornographic and violent ones. We validate the proposed pipeline for both pornographic and violent content classification. The pipeline and its results are under an ongoing process of scientific community's appreciation, regarding pornography classification [66], and violence classification [67].

Space-temporal video interest point detector and video content descriptor
Referred to as Temporal Robust Features (TRoF), such interest point detector and video descriptor constitute a lightweight space-temporal alternative, when compared to the more computationally intensive space-temporal solutions from the literature. It is fast and presents low-memory footprint, what makes it possible to run on limited hardware, such as mobile devices. To reach such efficiency, TRoF relies on a sparse strategy, which detects an optimized amount of space-temporal interest points within the video timeline. The detection process is Hessian-based, and relies on integral video and box filters for fast computation. The description process, in turn, is optimized by selecting only a small amount of video voxels around the previously detected space-temporal interest points. We validate TRoF for sensitive-video classification, and for sensitive-content localization. TRoF is also currently under an ongoing process of scientific community's appreciation, by means of the papers [66, 67].

High-level multimodal fusion pipeline for sensitive-content localization
Such pipeline is based on the combination of different and independent sensitive-snippet classifiers. Given that each snippet classifier can rely on a particular data modality (e.g., video frames, audio stream, video space-time, etc.), the pipeline has an important multimodal capability. Besides that, we recommend analyzing the content of different time-overlapping snippets, in order to provide a dense sampling

and a dense classification of the video timeline. The combination of classifiers is done by means of a late fusion of the sensitiveness classification scores that are returned by each snippet classifier. Scores that refer to the same video instant of interest are used to generate a single time-localized fusion feature vector. For deducing the fusion-vector configurations that better indicate sensitive and non-sensitive video moments, we employ machine-learning techniques. Similar to the pipeline for sensitive-video classification, the present sensitive-content localization pipeline is of general purpose; it can be used — without step modifications — for the detection of diverse sensitive content types (e.g., gore scenes, child abuse, pornography, violence, etc.). We validate it for both pornographic and violent content localization. The pipeline is subject to the deposit of two patents, one in the Brazilian National Institute of Industrial Property (INPI) [5], and the other in the United States Patent and Trademark Office (USPTO) [6]. In addition, it led us to reach second place in an international competition of violent scenes localization [4]. Finally, we intend to submit the solution to the scientific community's appreciation by means of a regular paper [65].

Large frame-level-annotated pornographic video dataset

Referred to as Pornography-2k dataset, it is a challenging set of 2,000 webvideos, which comprises 140 hours of video footage. Such dataset is useful for pornographic video classification, in the sense that it comprises 1,000 pornographic samples, and 1,000 non-pornographic samples, which vary from six seconds to 33 minutes. In addition, it is also useful for pornographic content localization, since we provide frame-level annotation for the 140 hours of video footage, of which 91.5 hours depict pornographic scenes, and 48.5 hours depict non-pornographic scenes. To the best of our knowledge, Pornography-2k is the first pornographic dataset in the literature that provides binary annotation (i.e., pornographic vs. non-pornographic) for every one of its frames. The dataset is available free of charge to the scientific community, upon request and the sign of a proper responsibility agreement, due to its sensitive content.

1.4 Accomplishments

In summary, the main results of this research are:

- Two patents, one in the Brazilian National Institute of Industrial Property (INPI) [5], and the other in the United States Patent and Trademark Office (USPTO) [6].
- Two journal publications, one under minor revisions [66], and the other in the final stages of preparation [65].
- Three conference papers, two already published [4, 64], and one under revision [67].
- Second-place award in an international competition of video violence localization [4].

1.5 Sponsorship

This research was 65% sponsored by Samsung Eletrônica da Amazônia Ltda., through the *Sensitive Media Project* under coordination of Prof. Anderson Rocha, and 35% sponsored by the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES). The two resulting patents are fully and exclusively licensed to Samsung Electronics. We herewith thank our sponsors for all the support during the research.

1.6 Thesis Roadmap

For a better understanding of the remaining parts of this thesis, we organized it as follows. In Chapter 2, we review the literature, regarding the foundations of this work. In the sequence, we divide the text into two major parts.

In Part I, we focus on the problem of sensitive-video classification, and on the verification of hypothesis *H1*. It comprises three chapters. In Chapter 3, we present the solutions that we are proposing for performing sensitive-video classification. In Chapter 4, we explain the experimental setup, and we report results for the classification of pornographic video, while in Chapter 5, we do so for the classification of violent videos.

In Part II, we focus on the problem of sensitive-content localization, and on the verification of hypothesis *H2*. Similarly to the previous part, Part II also comprises three chapters. In Chapter 6, we present the solutions that we are proposing for performing sensitive-content localization. In Chapter 7, we explain the experimental setup, and we report results for the localization of pornographic scenes, while in Chapter 8, we do so for the localization of violent scenes.

Finally, in Chapter 9, we present the conclusions of the research, and we elaborate on possible future work.

Chapter 2

Literature Review

In this chapter, we establish the foundations of this research. For that, we divide the state of the art of sensitive-video analysis in three sections. In Section 2.1, we elaborate on the problem of incorporating temporal information to the task of video content analysis, and how researchers have been tackling it. In Section 2.2, we survey the works that deal with pornographic content detection[1], while in Section 2.3, we review the literature that is related to the detection of violent content.

2.1 Temporal-Information Incorporation

This research is mainly founded upon Bag-of-Visual-Words (BoVW) approaches, to perform video content analysis. By doing so, we join the investigations of several other researchers — in the field of Content-Based Visual Information Retrieval — that have been betting on the BoVW model to reduce the semantic gap between the low-level visual data representation (e.g., pixels), and the high-level concepts one may want to recognize (e.g., violence and pornography).

The typical BoVW video analysis pipeline can have its operation properly framed in a three-layered representation. Within it, the (i) low-level layer refers to the video description, a process that commonly employs local descriptors to extract perceptual features directly from the pixel values. One level up, the (ii) mid-level layer aims at combining the low-level features into global video representations, with intermediate complexity. On top of that, the (iii) high-level layer deals with the challenge of learning and predicting the classes of the mid-level features.

Figure 2.1 depicts the typical BoVW framework, with the three levels properly chained in a low-to-mid and mid-to-high fashion, from left to right. 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. Firstly, in the so-called training phase, the visual codebook is constructed (or updated) for posterior reference, and the desired behavior of the system is learned from labeled video examples. Secondly, in the test phase, unknown videos are presented to the system; in this case, it must determine the video labels based on the codebook and classification model that

[1]Herein, we employ content *detection* and content *analysis* interchangeably.

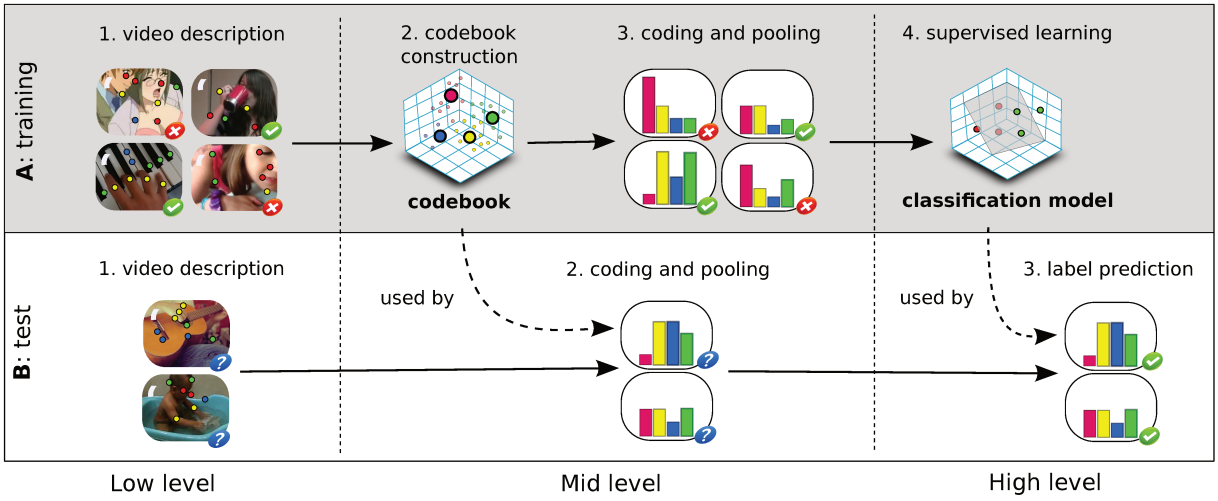


Figure 2.1: A typical three-level BoVW framework for video content analysis. On the top, the darker box depicts the training system operation, in which video labels are known in advance. On the bottom, the lighter box depicts the test operation, in which the codebook and classification model — previously learned in the training operation — are used by the system. Please notice that, in this case, the video labels are predicted only in the late stages.

were formerly learned.

As a result of such characteristic, Figure 2.1 depicts the workflow with two paths: the one related to the training system operation, depicted by the darker horizontal box, and the one related to the test operation, depicted by the lighter one.

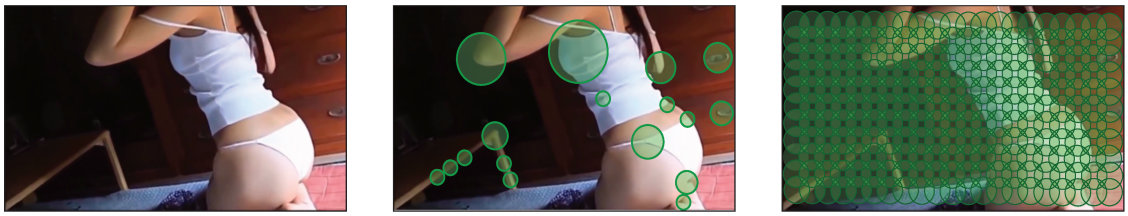
In the particular case of still-image analysis, several researches have been conducted in the direction of finding better strategies to implement each one of the aforementioned layers, as well as better methods to combine them [2, 21, 71, 16, 70]. However, in the case of analyzing motion pictures, to the best of our knowledge, it remains unclear what are the best ways to benefit from the time dimension that is inherent to videos.

Therefore, in the following sections, we delve into the alternatives we find in the literature, for each one of the BoVW levels, regarding the incorporation of temporal information.

2.1.1 Time-Aware Local Descriptors

In spite of the operation mode, the first step of a typical video-related BoVW framework is always connected to the task of video description (steps *A:1* and *B:1*, in Figure 2.1), which we call low-level stages. At this point, we must consider that each frame — delivered by a digital camera to a computer — corresponds to a collection of numbers that measure the amount of light that was incident on particular locations (pixels), within a photosensitive surface, at the very moment of capture. Thus, the inherent challenge is to extract useful information from such numbers.

Concerning such challenge, Tuytelaars and Mikolajczyk [87] early attested the success of the employment of local descriptors to the development of good computer vision systems. One can find in the literature several alternatives of local descriptors, with *Scale-*



(a) original (b) interest points (c) dense sampling

Figure 2.2: Illustration of interest points and dense sampling. Interest points provide focus on relevant visual phenomena (e.g., edges, corners, blobs, etc.), while dense sampling provides a systematic coverage of the image content.

Invariant Feature Transform (SIFT) [61] and *Speeded-Up Robust Features* (SURF) [11] 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 methods engineered to combine these features.

Keeping in mind the video nature as a sequence of frames in time, the conventional descriptors rely solely on the space domain of the frames, thus analyzing the pixel values strictly in the frame they occur. Such descriptors can be considered *static*, in the sense that they do not consider the video time dimension, neither the order in which the frames occur inside 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 the video duration.

In contrast with the static features, there are descriptors that interpret the frame pixels more like voxels. Pixel values are thus analyzed considering a third dimension, that is their position in video time. Such descriptors can be considered *time-aware*, in the sense that the feature vectors they deliver somehow encode the space-temporal information that is inherent to the video stream. For instance, *Space-Time Interest Points* (STIP) [56] and Dense Trajectories [93] are representatives of such type of descriptors.

Nevertheless, as a result of the addition of that third dimension in the description process, more data becomes available to be analyzed in each description step. Hence, that usually leads to a higher computational cost, both in terms of processing time and memory consumption. Anyway, here, one can easily perceive an excellent opportunity for incorporating temporal information. By using time-aware descriptors, it becomes possible to push temporal information early on in the low-level stages of the framework. Researchers in [83, 14, 90, 84] report to follow this path.

Regardless of being static or space-temporal, local descriptors can operate on image/video content in one of two ways. On the one hand, they may count on strategies for selecting interest points, according to the detection of relevant visual phenomena (e.g., edges, corners, blobs, etc.), to describe sparsely-localized feature vectors. That is the case of the works in [37, 60, 14, 90]. On the other hand, they may admit a dense sampling of the image/video space, in which the target content is systematically divided into patches of fixed size, which are placed on a regular grid (possibly repeated over multiple scales); all patches must then be described. Such strategy is employed in [89, 51, 8, 18]. Please refer to Figure 2.2 for an illustration.

2.1.2 A Single Bag for the Entire Segment

At the mid-level stages of the system operation, the main goal is to transform the previously extracted local descriptions into a global and richer video representation.

In the particular case of training operation, prior to the feature extraction itself, there is the necessity to construct the visual codebook, for posterior reference. Step *A:2* of Figure 2.1 is related to such task. There, the basic idea is to somehow split the space of low-level descriptions into multiple regions, being each region associated to a visual word. Thus, by the storage of these visual words, we have a representative codebook.

Strategies to construct the visual codebook may vary a lot. Most of researchers tend to follow the original solution of Sivic and Zisserman [82]. Therefore, they apply k-means clustering on the description space, in order to pick k prototypes (commonly the centers of the clusters) to represent the visual words [96, 100, 85, 14].

In a different fashion, other investigators manage to use simpler strategies, such as randomly sampling the description space, in order to raffle k representatives. That is the case of the works in [90, 84, 83]. Indeed, Nowak et al. [70] reported results produced by such type of random construction, with no significant loss of performance when compared to the systematic clustering approach.

Additionally, more sophisticated strategies can also be used, such as the application of an Expectation-Maximization (EM) algorithm to establish a Gaussian Mixture Model (GMM) on the low-level description space. In such cases, the centers of the GMM can be understood as the visual words. Deselaers et al. [37] report to follow this path.

Another possibility is the use of Random Forests (RF) [47], a combination of decision trees, which are individually built from the set of training descriptions. The inherent idea is that each tree node splits the description space in two, and that each tree leaf represents an actual cluster. Hence, the visual codebook corresponds to the set of all leaves, from all trees. Mironică et al. [63] report to use this strategy.

Regardless of the task of codebook construction, and common to both training and test phases (please see Figure 2.1), the main process of the mid-level feature extraction can be broken into two steps: coding and pooling (steps *A:3* and *B:2* in Figure 2.1). The coding step quantifies each low-level description with respect to its similarity to the words that compose the visual codebook[2]. The pooling step, in turn, aggregates the quantization obtained in the coding stage, by summarizing, usually in a single feature vector per video frame, how often the visual words are being manifested.

There are many ways to code and to pool the low-level descriptions [16, 71, 52, 7, 8]. Boureau et al. [16] surveyed on the traditional methods (e.g., hard- and soft-coding, and sum- and max-pooling). Perronnin et al. [71], in turn, experimented with the application of Fisher Vectors, to encode the average first and second order differences between the low-level descriptions, and the centers of a GMM-based codebook. Similarly, Jégou et al. [52] proposed a simplification of the Fisher Vectors, by relying only upon the first order differences, what they referred to as *Vectors of Locally Aggregated Descriptors* (VLAD). Finally, Avila et al. introduced *Bags Of Statistical Sampling Analysis* (BOSSA) [7] and

[2]In the case of a RF-based codebook, the coding step represents each low-level description through the leaves that are visited, when walking on the decision trees according to the low-level description content.

BossaNova [8], as peculiar pooling strategies.

Notwithstanding, the pooling step offers an interesting chance to incorporate temporal information into the mid level of the typical BoVW framework. The basic idea relies on answering the following question: instead of pooling the codes — obtained in the former coding step — per video frame, why not pooling and normalizing them per group of consecutive frames (a.k.a., video segment)? As a result, it becomes possible to gather a single feature vector for an entire video segment (i.e., a single bag), instead of gathering various, one for every described frame.

Depending on the intent of the video analysis (e.g., content localization, or classification), each segment may comprise only a single shot (in the case of Hollywood productions, in which there are many camera angles and scene cuts), a full scene (with all the consecutive events that happen in the same environment), fixed-length snippets, or even the entire video (useful for video clip classification). For instance, the authors of [99] established a single bag for each video shot, aiming at content localization. In [55, 29], the researchers established bags for fixed-length snippets, also focusing on localizing content, and in [18, 90], the authors reported to follow the strategy of entire video pooling, as their objective was to classify entire video clips.

2.1.3 Video-label Polling

Last but not least, at the high-level stages of the system operation, one finally has the video content properly coded as feature vectors. Thus, in training operation, the next step is to apply a method of supervised learning to deduce a good classification model, which is able to support the labeling of the input data. That is related to step *A:4*, in Figure 2.1. Once the classification model is defined, it becomes possible to predict the label of every given feature vector. Step *B:3* of Figure 2.1 refers to this moment.

Many machine learning solutions can be applied to this last classification process. However, Support Vector Machines (SVM) [91] are the most widely used technique in the BoVW literature, for both classification model learning, and label prediction. The difference among the publications rely mainly on the type of the kernel that is used to learn the separation hyperplane. For example, options may vary from Histogram Intersection (HI) kernels, to Gaussian Radial Basis Function (RBF), to Chi-Square (χ^2), and to Linear ones.

In the particular case of video content classification, in the commonest pipeline, a typical protocol of still-image classification is applied to predict the label of the video frames individually [19, 8, 7, 51, 60]. Therefore, a natural question is how to rely on such process, to label the entire video. The answer relies on ultimately incorporating the temporal information into the high level. For that, a voting scheme is adopted, in order to decide the label of the entire video. The class is thus assumed as being equal to the most voted label across the described frames. Please notice that such strategy exists as an alternative to the technique of establishing a single bag for the entire video.

2.2 Pornography Detection

Pornography consumption over the Internet has systematically increased in recent years [81]. However, contrary to the pervasiveness and availability of its hosting web, pornographic content cannot be disclosed to every audience, specially to minors, due to its highly sensitive nature. Moreover, some categories of porn are illegal, with child pornography being the obvious case [50]. Hence, pornography detection receives growing attention from law enforcement, and from the scientific community.

In this section, we survey the state of the art of pornography detection, by grouping the related work according to the features that they have in common.

2.2.1 Skin Detectors

The first efforts in the literature to automatically detect pornographic content, in digital images or videos, conservatively associated *pornography* with *nudity*. Hence, plenty of solutions were proposed, aimed at identifying naked bodies [73, 58, 43, 42]. In such works, the detection of human skin played a major role, commonly enhanced by the identification of human-body structures. A comprehensive survey on skin detection techniques can be found in [53].

Notwithstanding, it has long been reported that skin-detection-based pornography filters suffer from high rates of false positives, specially in situations of capturing activities with intense body exposure (e.g., swimming, sunbathing, boxing, etc.) [37]. That motivated the research for more effective solutions.

2.2.2 BoVW-Based Detectors

Here, we survey the works that applied BoVW-based techniques to perform pornography detection. Such solutions are close to the contributions of this research.

Multi-Categorical Porn

Aware of the advances promoted by BoVW approaches in the field of image recognition, Deselaers et al. [37] were the first to pose the pornography detection problem as an object classification one, rather than a skin detection or skin segmentation one. Thus, by the application of a task-specific visual vocabulary, they were able to conceive a BoVW model good at classifying images into five different categories of pornographic content: (i) inoffensive, (ii) with lightly dressed people, (iii) with partly nude people, (iv) with nude people, and (v) porn.

Nonetheless, a quick check on any pornographic website reveals an untold amount of image/video categories. If on the one hand it exposes the naïveté of designing solutions that rely solely on the detection of body and skin exposure (for example, one may easily find many sexual activities concerning people dressed with fetishist clothes), on the other hand, it reveals the complexity of any effort trying to establish reasonable and fully embracing multi-categorical approaches.

Nudity

In opposition to the multi-categorical strategy adopted in [37], some BoVW-field researchers insisted in viewing pornography as a matter of finding nude people.

This was the case of Lopes et al. [59], who employed a color descriptor (HueSIFT) in the low-level stages of their nudity classifier. Furthermore, they extended their solution to work with video content, by proposing a late voting scheme based on the classification of the individual video frames [60].

In the same direction of BoVW-based nudity classifiers, but this time betting on explicit skin detection to improve results, Steel [85] implemented a Gaussian skin masking for the isolation of image regions of interest (ROIs), before applying SIFT [61] as the low-level part of their method (what they called mask-SIFT).

Notwithstanding, nudity is a much simpler concept than pornography. It depends solely on the presence of naked people, whichever action they are taking. Pornography, on the contrary, is more subjective, and heavily dependent on socio-cultural aspects. Hence, it is not possible to guarantee that these nudity-detection-based works are enough for the effective detection of pornographic content in real scenarios.

Porn vs. Non-Porn

Still in opposition to the multi-categorical approach, but as a third distinct BoVW-based strategy, 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: it may involve cultural aspects and even personal value judgments. In spite of such complication, many of these researchers left, to the task of visual codebook construction, the opportunity for choosing the particular types of pornography one would want to classify.

For instance, Ulges and Stahl [89] 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 Discrete Cosine Transformation (DCT) in the YUV color space, before constructing their visual codebooks.

In the same sense of porn vs. non-porn, but yet influenced by the idea of combining skin detection with BoVW approaches, Zhang et al. [100] employed a skin-color-aware visual attention model to identify image ROIs, 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. [96] also used a color-aware visual attention model, which relied on the identification of salient and skin-colored faceless image ROIs. For a fast description, the researchers proposed the use of the SURF [11] descriptor.

Similarly, Zhuo et al. [102] proposed a BoVW approach that also focused on the fast description of formerly detected skin-colored regions, by employing the *Oriented fast and Rotated BRIEF* (ORB) descriptor [77].

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. [100], 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[3] human skin).

Adopting a Porn Definition

Attained to the importance and complexity of defining pornography, a series of works [7, 90, 84, 8, 19, 18] were inspired by the publication of Short et al. [81], and jointly adopted the concept of pornography as being *any explicit sexual matter with the purpose of eliciting arousal*. On top of that, as they tackled the classification of pornographic videos, they provided an interesting database composed of 800 webvideos (the Pornography-800 dataset [7], containing 400 pornographic, and 400 non-pornographic videos), which facilitated the efforts of comparing video pornography classifiers.

From such trend, by the occasions of proposing BOSSA [7] and BossaNova [8] (both extensions to the BoVW formalism), Avila et al. managed to solve the problem of classifying video pornography in the Pornography-800 dataset. 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 incorporated the video time dimension in the last stages of their pipeline, by applying a voting scheme based on the classification of the individual video frames.

Valle et al. [90], in turn, were pioneers at classifying pornography on the Pornography-800 dataset with the use of bags of space-temporal (STIP) features. Souza et al. [84] improved the results on the same database, by applying ColorSTIP — a color-aware version of the STIP detector [56] — and HueSTIP, a color-aware version of the STIP descriptor [56]. More than that, they innovated by keeping a single bag for the entire target video, instead of keeping a bag for each described video frame, prior to late voting schemes.

More recently, Caetano et al. [19, 18] also tackled the pornography classification problem related to the Pornography-800 dataset. In [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 [18], they improved the classification results by also establishing a single bag for the entire target video, instead of a bag for each extracted video frame.

Except for the works of Valle et al. [90] and Souza et al. [84], all the aforementioned BoVW-based solutions used bags of static features, which ignore significant and cogent information brought by video motion. However, motion information can be very revealing about the presence of pornographic content. That motivated the following works, as well

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

as ours.

2.2.3 Time-Aware Detectors

Regardless of the BoVW model, and having in mind the issue of employing descriptors more suitable to the non-static nature of video, publications other than [90, 84] were proposed in the literature, aiming at incorporating temporal information early on in the low-level description of video content.

For instance, Behrad et al. [12] tried to measure motion by analyzing the positions of skin-toned patches along the video frames. Aided by tree-based data structures that were used to register the temporal relation between distinct patches, they sought to pay attention to relevant volumes of skin, along the time dimension. Additionally, they extracted feature vectors that relied on the frequency domain of the frames, seeking to, somehow, code interesting skin motion.

Other publications relied on the content of the motion vectors intrinsically coded in the *Moving Picture Experts Group* (MPEG) video compression format [88, 51, 101, 39, 76]. Particularly in the cases of Ulges et al. [88], and Jansohn et al. [51], BoVW approaches were used to describe only the static visual features: the researchers, unfortunately, did not consider to apply BoVW to the motion-aware data.

Although effective in diverse tasks, space-temporal video detection approaches normally demand high computational power, thus impairing the final system performance, specially in terms of memory footprint, and spent processing time. In spite of that, none of the mentioned publications assessed performance, or observed efficiency, an important issue that we take into account in this work.

2.2.4 Third-Party Detectors

It is possible to purchase content-filter and crawler programs to inspect digital media for pornographic hints [15, 28, 86, 25, 97, 49, 74]. Some of these solutions indeed deal with visual content (image or video). For instance, MediaDetective [86] and Snitch Plus [49] are off-the-shelf products, which rely on the detection of human skin to find potential pictures or movies that may contain nude people.

Similarly, PornSeer Pro [97] is a free pornography classification system, which relies upon the identification of specific features (e.g., nipples, breasts, anuses, vaginas, lips, eyes, etc.) on individual video frames. Likewise, the work of Polastro and Eleuterio [74] (a.k.a., NuDetective) also adopts skin detection, and is supposed to be used by the Federal Police of Brazil, in forensic setups.

2.2.5 Summary

The comparison of pornography detectors from the literature is hardened by the absence of standardized datasets, groundtruths, and metrics. A myriad of publications present limited validation, except for the pornography classification methods that are proposed in [19, 18, 8, 84, 90], which report results on the two-class Pornography-800 dataset [8].

Indeed, very recently, Moustafa [68] reported results of fresh deep learning techniques on such dataset as well. Please refer to Table 2.1 for details concerning these results.

In addition, given our interest in BoVW, in Table 2.1, we summarize the works in the literature that applied BoVW-based solutions to classify pornography. From these works, Table 2.2 selects the ones that specifically tackled video classification. It summarizes in which level of the typical BoVW pipeline such strategies managed to incorporate the temporal information.

Finally, as one might observe, we were not able to find BoVW-based strategies in the literature that explicitly perform pornographic content localization within the video timeline. Furthermore, to the best of our knowledge, there is no properly annotated dataset to support the validation of such task either.

Table 2.1: BoVW-based pornography classifiers from the literature. Most results employed different protocols/datasets and are not directly comparable, except for the last five rows of the table, which employed the Pornography-800 dataset [8]. Moustafa [68] reported an accuracy of 94.1% on the same dataset, by using deep learning techniques.

	Reference	Media	Dataset (#pos/#neg)	Low level		Mid level		High level (SVM kernel)	ACC (%)
				Feature detector	Feature descriptor	Codebook	BoVW method		
Image	Lopes et al. [59]	Nude	90/90	SIFT blobs	Hue-SIFT	k-means	Traditional	Linear	84.6
	Steel [85]	Nude	1,500/1,500	Skin ROIs	Mask-SIFT	k-means	Traditional	RBF	*
	Deselaers et al. [37]	Porn	1,700/6,800	SIFT-based blobs	Difference of Gaussians	GMM	Traditional	HI	**
	Ulges and Stahl [89]	Porn	4,248/20,000	Regular grid	DCT	k-means	Traditional	χ^2	**
	Zhang et al. [100]	Porn	4,000/8,000	Skin ROIs	Color, texture, intensity	k-means	Traditional	Not reported	90.9
	Yan et al. [96]	Porn	20,000/70,000	Skin ROIs	SURF	k-means	Traditional	RBF	***
	Zhuo et al. [102]	Porn	8,000/11,000	Skin ROIs	ORB	k-means	Traditional	RBF	93.0
Video	Lopes et al. [60]	Nude	89/90	SIFT blobs	Hue-SIFT	k-means	Traditional	Linear	93.2
	Jansohn et al. [51]	Porn	932/2,663	Regular grid	DCT†	k-means	Traditional	χ^2	**
	Ulges et al. [88]	Porn	1,000/2,300	Regular grid	DCT†	k-means	Traditional	χ^2	**
	Avila et al. [7]	Porn	400/400	Regular grid	Hue-SIFT	k-means	BOSSA	χ	87.1
	Valle et al. [90]	Porn	400/400	STIP blobs	STIP	Random	Traditional	Linear	91.9
	Souza et al. [84]	Porn	400/400	Color-STIP blobs	STIP	Random	Traditional	Linear	91.0
	Avila et al. [8]	Porn	400/400	Regular grid	Hue-SIFT	k-means	BossaNova	χ^2	89.5
	Caetano et al. [19, 18]	Porn	400/400	Regular grid	Binary descriptors	k-medians	BossaNova	χ^2	90.9

Traditional BoVW mid-level representation is obtained with hard coding and average pooling — ACC: accuracy
*It uses False Positive Rate (FPR) as evaluation measure — **It uses Equal Error Rate (EER) as evaluation measure
***It uses Receiver Operating Characteristic (ROC) curve as evaluation measure — †It uses other low-level features, but not with BoVW

Table 2.2: Level of temporal-information incorporation in BoVW-based video pornography classifiers. A tick on the low level indicates the use of time-aware local video descriptors. A tick on the mid level indicates feature pooling and normalization for the entire video footage. Finally, a tick on the high level indicates majority voting on the label of the individual video frames.

Reference	Low level	Mid level	High level
Lopes et al. [60]			✔
Jansohn et al. [51]			✔
Ulges et al. [88]			✔
Avila et al. [7]			✔
Valle et al. [90]	✔	✔	
Souza et al. [84]	✔	✔	
Avila et al. [8]			✔
Caetano et al. [18]		✔	
Caetano et al. [19]			✔

2.3 Violence Detection

We now turn our attention to violence, which is a worldwide public health problem, constantly demanding efforts from authorities to provide the population with safer public places [95]. As a part of these efforts, experts have been investigating different forms of performing computer-aided violence detection on surveillance cameras, with the intent to support faster and more assertive official reactions, in situations of danger and crime occurrence, while alleviating the job of human operators.

Regarding the entertainment industry, the exposure to violence in media (including television, movies, music, etc.) represents a risk to the health of children, contributing to episodes of aggressive behavior, and desensitization to violence [26]. In this direction, researchers have been inspecting solutions to provide automated content filtering and rating, on movies and online video streams, with the aim of preventing the disclosure of violent material to inappropriate audiences.

In this section, we review the literature that is related to violent content detection, grouping the publications according to their application purpose, and to the features that they present in common.

2.3.1 Surveillance Detectors

Video streams from surveillance cameras are often silent and almost stationary, with well-behaved backgrounds, and people-centered foregrounds. Hence, surveillance-aimed works usually rely on background subtraction, people segmentation and tracking, and action recognition of fight-related concepts (e.g., punches, kicks, etc.).

In addition, due to their single-source nature, surveillance video streams do not present the notion of shots and scene cuts. As a result of that, the majority of works in the literature have been tackling the problem of violence surveillance as a matter of localizing events, within the stream timeline.

Nevertheless, just for an exceptional example, Hayashi et al. [46] were able to treat the problem as a matter of classifying video clips as violent or not. For that, as they wanted to detect assault-related events inside elevators, they suggested considering the frames between get-into and get-out events as single clips. To label these clips, they computed optical flow statistics for further decision making. With such strategy, they reported violence recall and precision, when testing a video dataset of their own.

Back to the trend of violence localization, for instance, Datta et al. [31] used background subtraction, people segmentation, estimation of the direction and magnitude of motion, among other methods, to feed a finite state machine and detect two standing people fighting. Mecocci and Micheli [62], in turn, suggested the use of background detection, and the analysis of the space-temporal complexity of local color conformations, to threshold on the amount of estimated motion. Both works reported only specific situations, in which their systems were able to localize violent acts along target stream timelines. No quantitative assessments were reported.

Zajdel et al. [98], in turn, aimed at detecting two to four people fighting, and vandalism against vending machines, inside train stations. They employed people segmentation and

interest point tracking to register human activity, which was quantified regarding a five-degree violent-activity scale. Additionally, moving trains were discarded by an optical-flow-based detector. In contrast to the previous work, they admitted microphones on the surveillance cameras, for exploiting complementary auditory features (e.g., pitch and spectral tilt). For the labeling of the audio data, they reported the use of decision trees. In the end, dynamic Bayesian Networks were used to implement a time-series model, which was responsible for ultimately fusing all the feature labels. For reporting the performance of their system, Zajdel et al. provided a 13-clip test dataset.

Surveillance detectors suffer from the drawback of not being suitable for detecting violence in broad-category movies, because movies usually present intense film transition, with variable pace rates. That is the main motivation for the next group of works.

2.3.2 Hollywood Detectors

Aware of the plot- and camera-oriented nature of Hollywood movies, many works in the literature have been taking advantage of the well-known film grammar of the movie industry. Although adopting different strategies, they have been similarly making use of at least one of the following aspects, for inferring scene nature: sound effects, visual effects, pace rate, and soundtrack.

Some researchers tackled the problem of classifying entire movie segments as violent or not. For instance, Gong et al. [45] classified movie shots by relying on the detection of gunshots, explosions, racing cars, screams, etc., through auditory features — such as bandwidth, pitch, and Mel-Frequency Cepstral Coefficients (MFCC) [32] — and by analyzing the scene and soundtrack pace rates, as indicators of frantic moments. For the final labeling of each feature type, they employed SVM classifiers, which were late fused by boosting techniques. With such strategy, Gong et al. reported violent-shot recall and precision, when testing four action Hollywood movies.

Giannakopoulos et al. [44] recommended the use of visual (motion vectors on frame blocks), and auditory features (e.g., Chroma [10] and MFCC). While visual features were fed to K-Nearest Neighbors (KNN) classifiers, auditory features were fed to a more complex combination of KNN classifiers and Bayesian Networks. In the end, a KNN-based late fusion method was used for returning the resulting class of each video segment. The authors reported the system recall and precision of gathering 9,000 violent one-second movie segments (which were extracted from ten movies).

Chen et al. [22], in turn, aimed at the detection of blood, fights, and injured people, by relying on visual clues only. For that, they suggested the segmentation of every movie into shots, which were grouped into scenes. From the scenes, they extracted video motion intensity (which was fed to an SVM classifier), and applied face and blood detectors. The authors reported violent-scene recall and precision, when testing four Hollywood movies.

In opposition to the classification of violent segments, some researchers tackled the problem of localizing violent events within the movie timeline. For example, Nam et al. [69] turned to the application of thresholds on the values of auditory and visual features. As auditory features, they explored the audio signal energy for detecting special effects of gunshots and explosions, while for visual features, they used pixel colors for detecting

fire and blood, and motion density for detecting frantic scenes. Similar to [31] and [62], Nam et al. made only qualitative assessments of their system performance, by pinpointing samples of violence localization, within movie timelines.

Cheng et al. [24], in turn, suggested the employment of only auditory features (e.g., bandwidth, volume, MFCC), for localizing gunshots, explosions, engines, helicopters, car breakings, etc. For that, they trained Hidden Markov Models (HMM), which were used to recognize the target sound events. To lately combine the used features, the authors suggested to seek specific combinations of sound events, that represented problem-domain situations of violence (e.g., *gunplay*, which was composed of gunshots, explosions, and engines). For modeling such situations, they adopted the concept of GMM-based *semantic contexts*. Besides presenting qualitative assessments of the detected events, Cheng et al. reported the recall and the precision of gathering semantic contexts, when testing five-minute segments that were extracted from five Hollywood movies.

In face of the current easiness of recording videos, and considering the growing offer of online amateur content, some of the aforementioned violence detectors may completely fail, due to the heterogeneity of material (regarding, for instance, illumination conditions, video and sound quality, erratic camera movement, and absence of plot or special effects). Given such situation, how could one automatically detect violent scenes in the broadest possible way? In this sense, some works in the literature appealed to the BoVW approach for designing and developing more general solutions.

2.3.3 BoVW-Based Detectors

We now turn our attention to some of these BoVW-based detectors, as they are relatively close (in concept) to the contributions of this research.

First Efforts

Souza et al. [83] proposed a motion-aware BoVW-based solution for classifying video shots as violent or not. The particularities of their strategy relied on the prior necessity of segmenting the target video streams into shots, as a very first step. The idea was to establish — after the hard coding of STIP-detected space-temporal low-level descriptions — one bag of features for each shot, as well as the further training and use of a linear SVM shot classifier. Experiments were conducted on a dataset comprising 400 webvideos (200 depicting aggressive behavior, 200 without hostility), and the motion-aware STIP-based solution was compared to a still-image SIFT-based counterpart, based on classification accuracy. By doing so, they were able to highlight the importance of using space-temporal features in violent content detection.

Similarly, Bermejo et al. [14] addressed video violence classification by applying a BoVW-based approach whose initial stages relied upon either STIP or Motion SIFT (MoSIFT) [23]. Again, the aim was to employ motion-aware low-level visual descriptors in the process. Experiments were conducted on a dataset that comprised 1,000 50-frame clips, which depicted hockey matches. Positive samples comprehended hockey fights, over which the authors reported violence classification accuracy.

The aforementioned work presented the limitation of reporting results on different datasets, with distinct metrics. Even the concept of violence was not the same, thus preventing direct comparison with other existing works in the literature, and a proper measure of the progress in the field. Problems such as this one motivated the MediaEval initiative, about which we shall discuss next.

MediaEval Initiative

By the occasion of proposing the *Violent Scenes Detection* (VSD) task, the *MediaEval Benchmarking Initiative for Multimedia Evaluation*[4] provided the scientific community with a unified violence dataset, with a common groundtruth — which reflected a clear understanding of the concept of violence — and standardized evaluation protocols. Since then, plenty of works were proposed in the literature, aiming at attending the VSD task. In the following, we focus on the researches that relied upon BoVW-related concepts for doing the job. For more details on the MediaEval initiative, and reviews about all MediaEval attendants, please refer to the reports in [80, 33, 34].

MediaEval Shot Classification

In its first years, the VSD task challenged participants to classify pre-segmented video shots as violent or not. In opposition to the works of Souza et al. [83] and Bermejo et al. [14], which used only visual features, a common trend among the VSD task attendants was the combination of visual and auditory features.

For instance, Acar et al. [1] calculated motion vectors from the shot frame blocks, and also extracted MFCC features from the shot audio streams. Curiously, is spite of training a first SVM shot classifier directly with the low-level motion vectors, in the case of the auditory features, however, these authors experimented with a bag-of-words approach. Hence, they applied k-means on the MFCC descriptors, for constructing an audio codebook, and established a Bag of Auditory Words (BoAW) per shot, prior to training a second SVM shot classifier. In the end, to provide a late fusion of features, they suggested to feed a third SVM classifier with the outputs of the previous ones.

In the same direction, Derbas and Quénot [36] proposed the use of Histograms of Optical Flow (HOF) [57] for describing STIP-detected space-temporal interest points, and MFCC for describing the audio stream. The most evident particularity of their approach relied on the early fusion of the low-level features, which were concatenated according to a randomly selected subset of all possible combinations, within a given video shot. By interpreting such concatenations as joint audio-visual features, the authors constructed codebooks with them, and established bags of audio-visual words, per shot, which were fed to SVM classifiers.

Lastly, aiming at performing fast shot classification, Mironică et al. [63] gave up using space-temporal features, in the particular case of the MediaEval dataset[5]. Instead, they employed fast global still-image frame description (e.g., based on Histograms of Oriented Gradients, HOG [30]), along with plenty of audio descriptors (e.g., MFCC, flux,

[4]Cf. `http://www.multimediaeval.org/`, accessed May 3rd, 2016.

[5]Mironică et al. [63] also applied space-temporal descriptors, but for other smaller datasets.

rolloff, etc.). In the mid level, codebooks were constructed with the support of Random Forests, and shots were represented by means of a VLAD-based approach. In the high-level, for each type of feature, one independent SVM shot classifier was trained. For ultimately fusing the outputs of these SVM classifiers, the authors recommended the use of a weighted-sum function.

For the sake of highlighting an open issue, none of the mentioned violent video classifiers had their efficiency analyzed, in terms of memory footprint and processing time. Except for Mironică et al. [63], all of them made use of space-temporal video detection approaches, which normally demand high computational power. In the particular case of Mironică et al. [63], although they aimed at fast shot classification, they assessed performance for the task of video genre classification only, but not for violence detection.

MediaEval Scene Localization

More recently, the VSD task challenged participants to localize violent scenes within the video timeline. For that, they maintained the two-class frame-level-annotated groundtruth, but did not provide any shot segmentation to the public.

That led to a major contrast between the previous BoVW-related classification works, and the further localization ones. Attendants of the past shot-classification task had often adopted the straightforward strategy of establishing a bag per provided shot, for further discrete classification. In opposition, attendants of the newly introduced content-localization task had to reckon with (i) the granularity of mid-level pooling (due to the absence of shots), and (ii) the method of online bag score fusion, for providing content classification with temporal continuity (a basic requirement for the localization task).

Regarding the granularity problem, given the many possibilities of video segmentation (frames, shots, time-overlapping snippets, etc.), in what unity should one pool the mid-level features, in order to provide bag labels that were more supportive of the task of content localization? One bag (and thus one label) per frame? One bag per second? Concerning the online bag score fusion, how should one combine the violence scores of the many discrete bags, in test execution, for providing a continuous answer?

Reasoning about all these open questions, Zhang et al. [99] kept the idea of segmenting the target streams into shots. For that, they employed a third-party shot boundary detection method. In the mid-level, for each type of feature (e.g., SIFT on regular grids, Dense Trajectories, and MFCC), they represented each shot by a proper Fisher Vector (equivalent to the notion of a bag). In the high-level, each set of feature-related Fisher Vectors was fed to a particular SVM classifier (i.e., they trained one SVM per feature type). Then, a weighted sum of classification scores was used for the final shot classification. Given that the labeled shots did not present time overlaps, Zhang et al. simplified the fusion of discrete bag scores. Their system just returned a time-sorted concatenation of the shot violence scores, when in test execution.

Contrary to [99], Lam et al. [55] opted for dividing the streams into non-overlapping five-second snippets. In the mid-level, for each type of feature (e.g., SIFT on regular grids, Dense Trajectories, and MFCC), each snippet was encoded as a Fisher vector, and as a bag of words. Besides that, the authors fed keyframes to a Deep Neural Network (DNN),

for obtaining a third alternative of mid-level representation (a further improvement on their original task attendance [54]). In face of plenty of mid-level representations (Fisher vectors, bags of words, and DNN outputs), one SVM classifier was trained for each feature type. To combine everything, a weighted sum of classification scores was performed, for the final snippet classification. In the end, in the online snippet score fusion, Lam et al. [55] proceeded as [99], configuring their solution to return a concatenation of the adjacent snippet violence scores.

Dai et al. [29], in turn, divided the target streams into non-overlapping fixed-length three-second snippets. In the mid-level, for some features (e.g., Dense Trajectories), they represented each snippet by a Fisher Vector. For other features (e.g., STIP and MFCC), they established conventional bags of words, one for each snippet. In face of such diversity of representations, they trained one SVM classifier for each feature type. Additionally, they fed some of the features to a DNN, that worked as a high-level classifier, equivalent to the SVMs. Once more, a weighted sum of classification scores was performed, for the final snippet classification. In contrast to the previous solutions, Dai et al. suggested a more complex strategy for the online bag score fusion. Snippet classification scores were first smoothed by a proper function. Then, each snippet received a label (violent or non-violent), according to a threshold on the smoothed scores. In the end, adjacent snippets with the same label were merged into a single segment, whose final violence score was set as the average of the merged scores.

The solution for violent-content localization we are proposing in this work is contemporaneous to the above-mentioned researches of [99, 55, 29]. With such strategy, we reached second place in the 2014 MediaEval VSD task competition, regarding the localization of violent scenes within webvideos. In opposition to [99, 55, 29], we recommend a late fusion of distinct time-overlapping-snippet classifiers, which shall rely upon different and complementary data modalities (e.g., video frames, audio stream, and video space-time). The combination of classifiers is done with machine-learning techniques, which are used to determine the best way of combining the classification scores that are returned by each snippet classifier (i.e., through a meta-learning procedure). In addition, the solution is amenable to the localization of sensitive contents other than just violence; we also validate it for pornography localization. For a complete description of the method, validation and experiments, please refer to Part II.

2.3.4 Summary

In Table 2.3, we summarize the works in the literature that applied BoVW-based solutions for detecting violence (regarding both classification and localization challenges). In contrast to the pornography-related solutions, all these works made use of space-temporal features, by employing low-level time-aware descriptors (e.g., STIP, MoSIFT, Dense Trajectories, and MFCC), and establishing a single bag per shot (or per interest snippet).

Table 2.3: BoVW-based violence detectors from the literature. Results are directly comparable if they share the same dataset.

	Reference	Dataset	Low level		Mid level		High level	MAP
			Feature detector	Feature descriptor	Codebook	Method	(SVM kernel)	
Classification	Souza et al. [83]	Violence-400	SIFT blobs; STIP blobs	SIFT; STIP	Random	Traditional BoVW	Linear	*
	Bermejo et al. [14]	Hockey Fights	STIP blobs; MoSIFT	STIP; MoSIFT	k-means	Traditional BoVW	HI	*
	Acar et al. [1]	MediaEval 2012	MFCC	MFCC†	k-means	Traditional BoAW	RBF	0.545
	Derbas and Quénot [36]	MediaEval 2013	MFCC and STIP	MFCC + HOF (early fusion)	k-means	Traditional BoVW	RBF	0.690
	Mironică et al. [63]	MediaEval 2013	Regular grid; MFCC, etc.	HOG; MFCC, etc.	Random Forests	VLAD	RBF	0.760
Localization	Zhang et al. [99]	MediaEval 2014	Regular grid; Dense Trajectories; MFCC	SIFT; Dense Trajectories; MFCC	GMM	Fisher Vectors	Linear	0.566
	Lam et al. [55]	MediaEval 2014	Regular grid; Dense Trajectories; MFCC	SIFT; Dense Trajectories; MFCC	k-means; GMM	Traditional BoVW; Fisher Vectors	Linear‡	0.564
	Dai et al. [29]	MediaEval 2014	STIP; Dense Trajectories; MFCC	STIP; Dense Trajectories; MFCC	k-means; GMM	Traditional BoVW (STIP, MFCC); Fisher Vectors (Dense Trajectories)	χ^2; Linear‡	0.630

Traditional BoVW and Traditional BoAW mid-level representations are obtained with hard coding and average pooling
MAP: mean average precision (MediaEval VSD task official metric)
*It reports accuracy as evaluation measure — †It uses other low-level features, but not with BoVW — ‡It also performs DNN-based classification

2.4 Final Remarks

In general, sensitive-content detection techniques are non-generalizable and purpose-dependent. For instance, most pornography detectors rely upon skin recognition, which might not be useful for detecting violence. Similarly, a multitude of violence detectors rely upon blood and special-effects recognition, which might not be useful for pornography detection. That hardens the duty of dealing with the high subjectivity of the target concepts. For example, it might be difficult to adapt skin-recognition-based pornography detectors to the reality of tropical countries, where body exposure is common and well-accepted. Under such circumstance, how should one proceed to reduce false negatives? In this work, we propose broader machine-learning general-purpose pipelines, that can be adapted to most of the sensitive contents one might want to detect (e.g., violence, pornography, child abuse, cruelty to animals, etc.). All one needs to do is to provide the algorithms with a properly annotated dataset, with enough sensitive and non-sensitive examples. Of course the notion of "enough" here depends on the difficulty of the problem, but often a few hours of each concept is enough for a good generalization of the designed detectors.

In the particular case of video pornography detection, the traditional approach extends the still-image classification process to video, by simply labeling the frames individually, and then performing majority voting to decide the label of the entire clip. That is a poor design, that does not take into account video motion, which might be very revealing about the sensitiveness of the target stream. Indeed, even well-known breakthrough video representations from the literature, such as Fisher Vectors, were never applied to the problem of pornography detection, to the best of our knowledge. Moreover, we were able to find in the literature only works that tackled video pornography classification. There is a lack of solutions for pornography localization, as well as standardized frame-level annotated datasets. In this work, we contribute to the scientific community by tackling these issues, by proposing more effective motion-aware solutions, and by releasing a large frame-level annotated pornographic dataset, which is fundamental for pornography localization.

In the particular case of video violence detection, there is already an available standardized dataset (MediaEval), and the available solutions in the literature have long been applying video representation techniques, such as space-temporal descriptors, and Fisher Vectors to the problem. Nevertheless, efficiency is not a commonly investigated matter, specially in terms of memory footprint, and spent processing time. It is not investigated also in the case of pornography detection. As a consequence, solutions are probably not ready to deal with the big-data and urgent nature of the sensitive-content detection task (given that efficiency is not a major concern), and they might not be amenable to run on hardware-limited mobile devices, to benefit from their pervasiveness. These are open issues in the literature that we also consider in this work.

In the following, we introduce and validate the methods we have mentioned thus far. More specifically, in Part I, we tackle the sensitive-video classification problem, while in Part II, we deal with sensitive-content localization.

Part I

Sensitive-Video Classification

Chapter 3

Getting Clues from Video Space-Time

Sensitive-video classification is the decision problem of defining whether or not a given video stream has any occurrence of a particular sensitive content. By definition, labeling a stream as positive means that the target sensitive concept is present within it. In opposition, labeling as negative indicates that the target concept is absent.

In this chapter, we introduce an end-to-end approach for time-aware sensitive-video classification, which is designed to be efficient (i.e., to be fast and to present low-memory footprint). The pipeline efficiency mainly relies upon a novel space-temporal interest point detector and video descriptor, namely Temporal Robust Features (TRoF), which is also introduced.

This chapter is related to hypothesis *H1* (please refer to Section 1.1), which states that it is possible to efficiently use video temporal information for effective sensitive-content classification, by combining simplified space-temporal video interest-point detection and description, with entire-footage representation through a single feature vector. It aims at the goal of designing and developing effective and efficient methods for sensitive-video classification. For that, we organized the text as follows. In Section 3.1, we detail the video classification pipeline, while in Section 3.2, we introduce the TRoF video descriptor. We then present final remarks related to the proposed solutions in Section 3.3.

3.1 Time-Aware Pipeline for Efficient Sensitive-Video Classification

Sensitive concepts such as pornography and violence represent high-level semantic categories, whose translations to visual characteristics are not straightforward. As already mentioned, to cope with such complexity, we propose to rely on BoVW-based strategies, for reducing the semantic gap between the low-level visual data representation (e.g., video frame pixels), and the high-level target sensitive concept.

Moreover, given our interest in performing effective and efficient time-aware sensitive video classification, we introduce a general-purpose end-to-end BoVW-based pipeline, which efficiently incorporates temporal information as an effective discriminative clue for the task of sensitive-video classification. We say that such pipeline is of general purpose, in the sense that it can be used — without step modifications — for the binary classification

(positive vs. negative) of diverse sensitive content (e.g., violence, pornography, gore scenes, child abuse, etc.). For employing the pipeline concerning a specific concept, all one needs to do is to provide a properly annotated training dataset, with positive and negative examples.

Figure 3.1 depicts the proposed pipeline, with the inherent three levels. As expected from a typical machine-learning solution, the pipeline can be executed either in (i) *training* mode (represented by the left larger column), or in (ii) *test* mode (represented by the right darker column).

In the former mode, the labels of the videos are known in advance, and are used for training the class-prediction capabilities of the system. For the sake of illustration, we start the training operation with only two videos (*video A*, positive, and *video B*, negative), but in a real-world application, it would involve much more samples. At this point, efficiency is not a major concern, since the system shall be trained only a few times (ideally just once). In the latter operation, in turn, the system shall efficiently predict the label of arbitrary videos (e.g., *video X*), with low-memory footprint, and small processing time. In the following sections, we detail each pipeline level, from low- to high-level stages.

3.1.1 Low-level Stage

First of all, for the sake of efficiency — and similar to Akata el al. [2] — 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 Figure 3.1, and considerably reduces the amount of data to be analyzed.

Given that we want to push temporal information early on in the low-level stage, we suggest the employment of local space-temporal descriptors, for the video description steps (*Steps A:2* and *B:2*). These descriptors usually deliver d_f-dimensional feature vectors that somehow encode the variation of the frame pixel values, regarding not only their spatial configuration, but also their disposition along the video timeline (i.e., pixels are analyzed as voxels). STIP [56] and Dense Trajectories [93] are typical representatives of such descriptors. However, if space-temporal data are not parsimoniously used, they lead to a high computational cost, in terms of both processing time and memory footprint. That clashes with our goal of designing efficient solutions, specially regarding the intention of deploying solutions on mobile devices. Hence, we introduce *Temporal Robust Features* (TRoF), a novel time-aware video descriptor, which saves computational resources, yet maintaining reasonable video description capability. In Section 3.2, we detail the TRoF detector and descriptor.

3.1.2 Mid-level Stage

In the mid level, the goal is to combine the low-level features into global video representations, with intermediate complexity, which are closer to the target high-level sensitive concept (e.g., violence, or pornography).

Firstly, for the sake of using the chosen mid-level representation — which we will shortly detail as being Fisher Vectors — we reduce the d_f-dimensional low-level feature

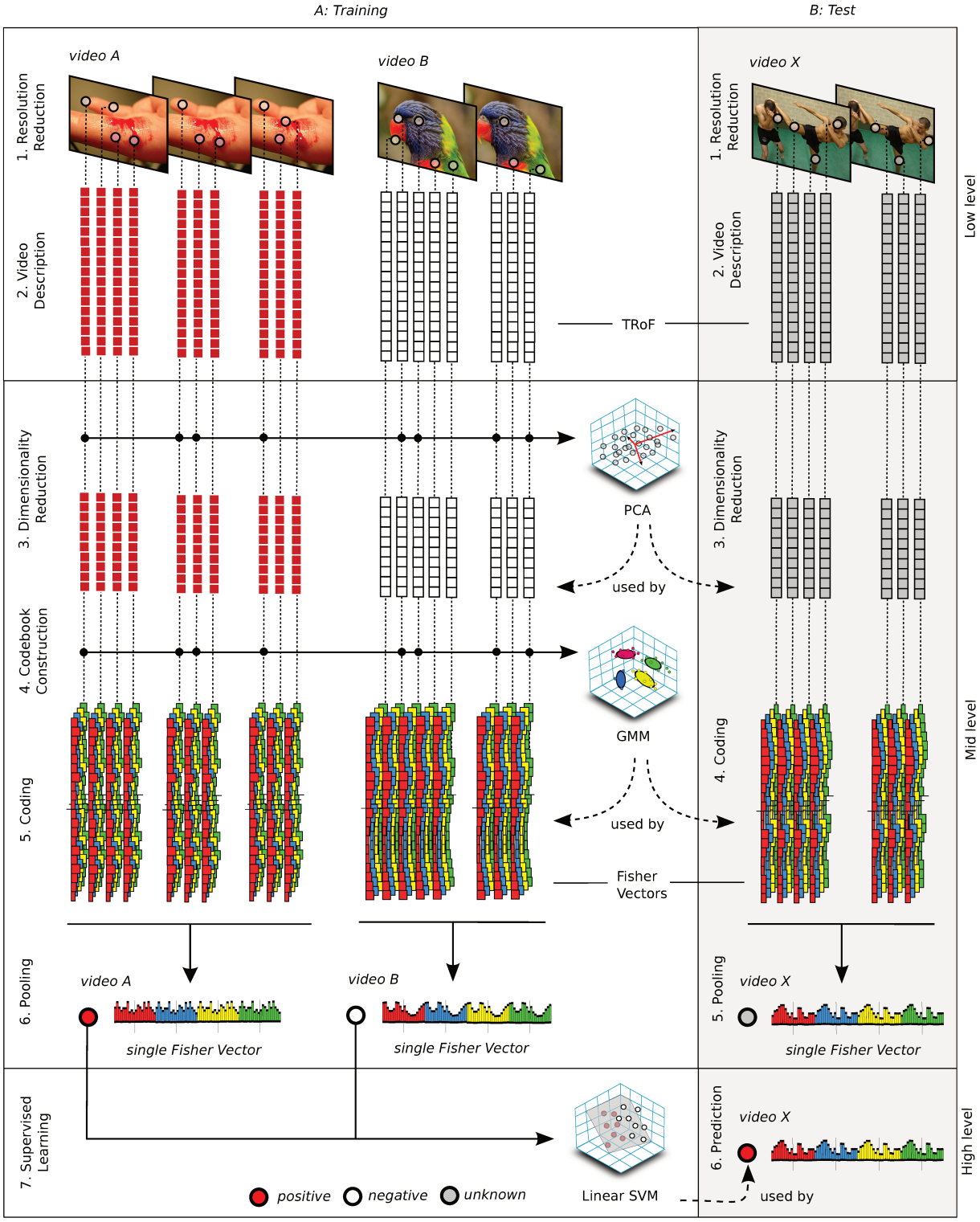


Figure 3.1: Three-level pipeline for efficient sensitive-video classification. On the left, the larger column depicts the training pipeline execution, in which video labels are known in advance, and are used for calculating the principal component analysis (PCA) transformation matrix (in *Step A:3*), generating the GMM codebook (in *Step A:4*), and training the linear SVM classification model (in *Step A:7*). On the right, the darker column depicts the test execution, in which the formerly learned models are used by the system, for predicting the class of arbitrary videos. This pipeline efficiently 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.

vectors to $p_f \leq d_f$ dimensions, with Principal Component Analysis (PCA). As pointed out by Sánchez et al. [79], the PCA dimensionality reduction is key to make the Fisher Vectors strategy work. That is related to *Steps A:3* and *B:3*, in the pipeline. More specifically, regarding *Step A:3* — in the particular case of training operation — 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 test 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.

In the sequence, as we want to benefit from breakthrough mid-level representations in the literature, we recommend the establishment of Fisher Vectors — one of the best mid-level representations in the literature of Computer Vision problems [21, 79] — for coding the video content with intermediate complexity (*Steps A:5* and *B:4*, in Figure 3.1). Roughly speaking, Fisher Vectors encode the average first- and second-order differences between the low-level descriptions, and the distributions of a GMM-based codebook [71].

Therefore, prior to the encoding step — and exclusive to the training operation — a GMM with c_{gmm} distributions is estimated through an EM algorithm, whose execution starts from random sampling features from the low-level PCA-reduced training feature space. At this point, similar to the PCA-sampling, we randomly select k_c PCA-reduced descriptions, with half of them coming from the positive training samples, and the other half coming from the negative ones. Such process is depicted in *Step A:4*, in which only some descriptions are used to generate the GMM codebook.

Once the coding step is concluded, each PCA-reduced low-level description is converted to a Fisher Vector with size $2 \times p_f \times c_{gmm}$, by definition [71]. For generating the final mid-level global video representations, we sum — for each video — all the Fisher Vectors along the time dimension (which is equivalent to establishing a single bag for the entire video, as explained in Section 2.1.2). As a result, each target video is represented by a single Fisher Vector, which is normalized by means of a 0.5-power normalization, followed by an ℓ_2-normalization, as recommended in [71]. By working with this reduced representation, we expect to alleviate computational costs, besides incorporating temporal information in the mid-level stage of the process. Such pooling step is represented through *Steps A:6* and *B:5*, in Figure 3.1.

3.1.3 High-level Stage

In the high level, many machine-learning algorithms can be used to infer a prediction function, for assigning labels to arbitrary videos (e.g., *porn* vs. *non-porn*, or *violent* vs. *non-violent* content). At this point, depicted by *Steps A:7* and *B:6* in Figure 3.1, we follow the literature and apply SVM (as explained in Section 2.1.3). We use a linear SVM, since it is well known that non-linear kernels do not improve classification performances for Fisher Vector representations [71]. In addition, a linear classifier is also of interest due to its recognized faster performance, when compared to non-linear ones.

Table 3.1: Parameters of the proposed sensitive-video classification pipeline.

Parameter	Meaning
f_r	Resolution, in pixels, to which the video frames are reduced.
p_f	Dimensionality of the low-level descriptions, after PCA reduction.
k_p	Quantity of descriptions sampled for PCA transformation calculation.
k_c	Quantity of PCA-reduced descriptions sampled for GMM estimation.
c_{gmm}	Quantity of GMM-codebook component Gaussians.

3.1.4 Parametrization Summary

The pipeline depicted in Figure 3.1 suggests the combined use of TRoF, PCA, GMM, Fisher Vectors, and a linear SVM for the final decision making. Nevertheless, it is noteworthy that these techniques can be replaced by alternative solutions, depending upon the application and the target system tradeoff between effectiveness and efficiency. Table 3.1 summarizes the pipeline parameters.

3.2 Temporal Robust Features (TRoF)

Local space-temporal features constitute a successful low-level representation for general action recognition [56, 93]. Nevertheless, one important factor that prevents their use in real-time applications is their high computational cost, regarding both processing time and memory footprint.

To solve this problem, we propose a fast yet-space-temporal alternative that can be implemented in limited hardware, such as mobile devices, and handheld video players. 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 sensitive-content classification task. For that, we investigated what type of hints we could observe in a video, and we singled out the motion information. To deal with the processing-time issue, we focus on employing fast image representations and manipulations, such as integral images, and box filters.

Therefore, (i) we custom-tailor a detector for finding relevant 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 3.2.1 introduces the TRoF detection method, while Section 3.2.2 explains its description approach. Finally, in Section 3.2.3, we demonstrate the TRoF detection capabilities, by means of synthetic test videos.

3.2.1 TRoF Detector

The TRoF detector is directly inspired by the still-image Speeded-Up Robust Features (SURF) detector [11], which is very fast. It relies on three major extensions of the original method, to use the video space-time: the employment of four-variable Hessian matrices, three-dimensional box filters, and the concept of integral video. In the following, we explain each one of these expansions.

Four-Variable Hessian Matrix

The original SURF detector [11] identifies interesting visual local structures (a.k.a., blobs) in an image, by means of 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 [11], 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 is 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. [11] 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 given 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. [94] 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. 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. Moreover, Willems et al. [94] suggest inspecting the spatial- and the temporal-scale search spaces separately. Hence, they propose the use of o_s five-layered spatial scale octaves, and o_t five-layered temporal scale octaves. Even though they give neither clues on the actual values used for the candidate standard deviations, nor how these values may be combined[1], we can stipulate that they must compute at most $o_s \times 5 \times o_t \times 5$ Hessian values, for every voxel.

In a similar fashion, we also extend the Hessian matrices, but with a different formulation, which is fundamental for real-time operation. In Equation 3.1, we express the content of a *four-variable* space-temporal Hessian matrix $H(x, y, t, \sigma_{st})$, such as we

[1]Source codes and executables are no longer available and, due to a lack of details in Willems et al.'s paper [94], we could not reproduce their method, making direct comparisons impossible.

Table 3.2: Four initial space-temporal octaves for the TRoF detector. The Increase Factor (IF) expresses the inter-layer scale increase. Values are measured in pixels.

Octave	Scales				IF
1	$9 \times 9 \times 9$	$15 \times 15 \times 15$	$21 \times 21 \times 21$	$27 \times 27 \times 27$	6
2	$15 \times 15 \times 15$	$27 \times 27 \times 27$	$39 \times 39 \times 39$	$51 \times 51 \times 51$	12
3	$27 \times 27 \times 27$	$51 \times 51 \times 51$	$75 \times 75 \times 75$	$99 \times 99 \times 99$	24
4	$51 \times 51 \times 51$	$99 \times 99 \times 99$	$147 \times 147 \times 147$	$195 \times 195 \times 195$	48

are adopting in this work. Within it, $L_{xx}(x, y, t, \sigma_{st})$ is the convolution of the Gaussian second-order derivative $\partial^2 G(x, y, t, \sigma_{st})/\partial xx$ with the voxel $\mathbf{x}(x, y, t)$ of the target video. Similarly, $L_{xy}(x, y, t, \sigma_{st})$ refers to the convolution of $\partial^2 G(x, y, t, \sigma_{dt})/\partial xy$ with the voxel $\mathbf{x}(x, y, t)$, and so forth for L_{xt}, L_{yt}, L_{yy}, and L_{tt}.

$$H(x, y, t, \sigma_{st}) = \begin{bmatrix} L_{xx}(x, y, t, \sigma_{st}) & L_{xy}(x, y, t, \sigma_{st}) & L_{xt}(x, y, t, \sigma_{st}) \\ L_{xy}(x, y, t, \sigma_{st}) & L_{yy}(x, y, t, \sigma_{st}) & L_{yt}(x, y, t, \sigma_{st}) \\ L_{xt}(x, y, t, \sigma_{st}) & L_{yt}(x, y, t, \sigma_{st}) & L_{tt}(x, y, t, \sigma_{st}) \end{bmatrix}. \tag{3.1}$$

As one might observe, we propose using a single standard deviation σ_{st} for both space and time. At this point, differently from Willems et al. [94], and for a matter of simplification, we adopt a joint strategy that — as a relaxation — lets us variate the scale of the detectable blobs faster and closer to the former proposition of Bay et al. [11]. We thus apply o four-layered space-temporal scale octaves (our first detection parameter), of increasing Gaussian standard deviations with dual nature (spatial and temporal). As a result, it becomes necessary to compute only $o \times 4$ Hessian values, for every candidate voxel (less than the $o_s \times 5 \times o_t \times 5$ values from Willems et al. [94]).

To support such significant scale search space reduction, we extend the four-layered octaves that were settled by Bay et al. [11] — 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, the first space-temporal octave starts with a scale of $9 \times 9 \times 9$ voxels, and it presents an inter-layer increase of six voxels, for both space and for time. The resulting space-temporal octave thus comprises four scales, with $9 \times 9 \times 9$, $15 \times 15 \times 15$, $21 \times 21 \times 21$, and $27 \times 27 \times 27$ voxels, respectively. Table 3.2 details the proposed scales for four consecutive space-temporal octaves.

At first glance, the employment of a joint scale σ_{st} may sound counterintuitive, given the distinct nature of space and time. However, preliminary experiments revealed that, besides the advantage of enabling real-time video description, thanks to the scale-space simplification, such strategy works on par with scale-separated solutions, in the case of detecting inappropriate content. That happens because of the nature of the problem that we intend to solve. While Willems et al. [94] aimed at action recognition, a duty that is fundamentally of specialization nature, we are interested in violence classification, a generalization task that does not require a precise detection of repeatable interest points.

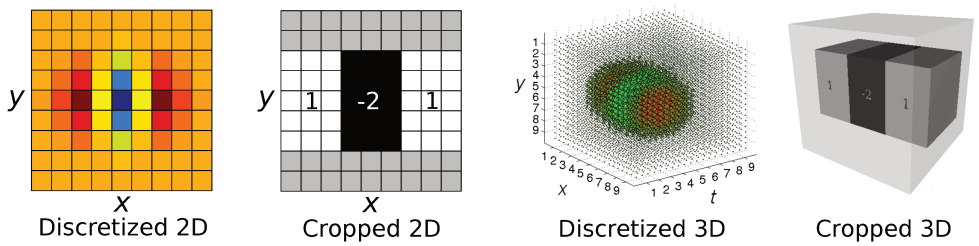


Figure 3.2: A visualization of the derivative filters $\partial^2 G(\mathbf{x}, \sigma)/\partial xx$, and their approximations. (a) The original two-dimensional filter, with its discretized and cropped versions. (b) The respective three-dimensional versions. The rightmost cuboid filter is one of the six filters used by the TRoF detector to support the calculation of Hessian matrices.

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 four-dimensional neighborhood, considering the immediate Hessian neighbors along the x-, y-, t-, and σ_{st}-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_{st} 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 one detection parameters s, which defines the initial sampling step in both spatial and temporal directions, for selecting the points where to calculate the Hessian values. We also recommend to double that step 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 parameter, one must consider that larger values of s 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.

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 with the integral image of the target image.

Figure 3.2(a) depicts 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, Figure 3.3(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 four-variable Hessian matrices (please refer to Equation 3.1), hence the related Gaussian second-order derivatives are three-dimensional (spreading across the x, y, and t directions of the video space-time), with space-temporal scale σ_{st} in all directions. We approximate these derivatives with cuboid filters.

Figures 3.2(b) and 3.3(b) show two of the six Gaussian filters, in both discretized and

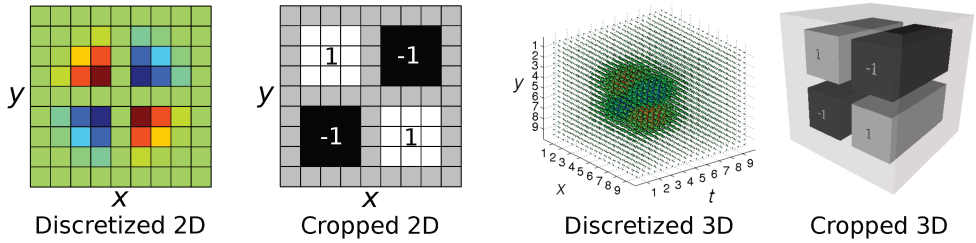


Figure 3.3: A visualization of the derivative filters $\partial^2 G(\mathbf{x}, \sigma)/\partial xy$, and their approximations. (a) The original two-dimensional filter, with its discretized and cropped versions. (b) The respective three-dimensional versions. The rightmost cuboid filter is one of the six filters used by the TRoF detector to support the calculation of Hessian matrices.

cuboid cropped versions. The other remaining four cuboid filters can be easily deduced by simply applying the proper rotations. The $9 \times 9 \times 9$ filters represent approximations of Gaussians with $\sigma_{st} = 1.2$.

In all elements of Figures 3.2 and 3.3, 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 to red, and green closer to blue. Cropped box filters, in turn, are approximations, with values explicitly shown on the images. As adopted in [11], gray positions have zero value, while white areas are positive, and black are negative.

Integral Video

The original SURF detector relies on *integral images* [27] to quickly perform image convolutions. In the case of TRoF, which operates within the video space-time, we must extend the concept of an integral image to the idea of an integral video, by considering three dimensions rather than two.

Equation 3.2 states the value of an integral video $V_\Sigma(\mathbf{x})$ at a space-temporal location $\mathbf{x}(x, y, t)$, as suggested by Willems et al. [94]. It is 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.

$$V_\Sigma(\mathbf{x}(x, y, t)) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} \sum_{k=0}^{k \leq t} (i, j, k). \tag{3.2}$$

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 Figure 3.4 is given by Equation 3.3.

$$V = (A + C) - (B + D) - (A' + C') + (B' + D'). \tag{3.3}$$

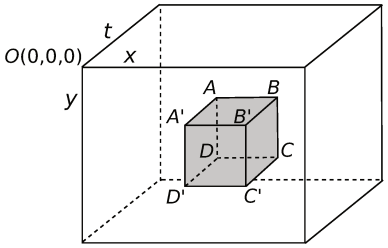


Figure 3.4: Integral video representation. The outer box represents the video space-time, with the x axis associated to the width, the y axis to the height, and the t axis to the video duration. The inner gray box represents the cuboid region, which is calculated by Equation 3.3, as suggested by Willems et al. [94]

Table 3.3: Parameters of the TRoF detector. Parameter s is measured in pixels. The remaining parameters refer to scalar quantities.

Variable	Meaning
o	Quantity of analyzed space-temporal scale octaves.
s	Initial video space-time sampling step.
c	Quantity of frames within each analyzed integral video.
b	Quantity of extracted interest blobs per integral video.

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 third 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, damage 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 (fourth and last TRoF parameter).

Table 3.3 summarizes the four parameters we have designed for the TRoF detector.

3.2.2 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 space-temporal scale σ_{st}. The next step refers to the description process to represent these elements.

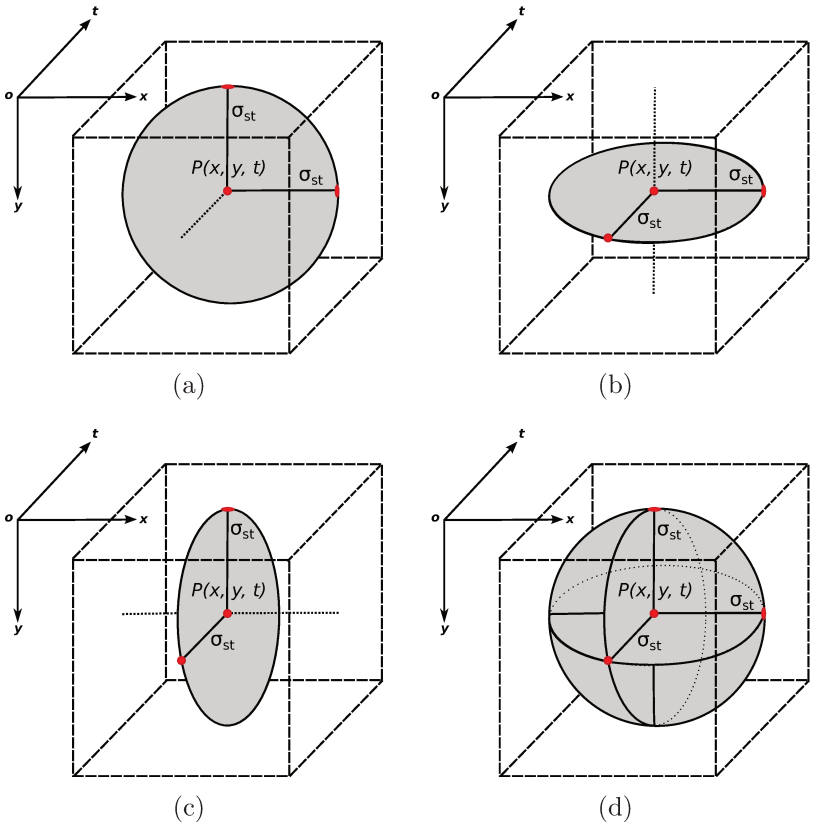


Figure 3.5: SURF-based TRoF described blob planes. The solid gray circles are conventional SURF blobs, which are all centered at position $P(x, y, t)$, and present a space-temporal scale of $2\sigma_{st}$. P and σ_{st} come from a formerly detected interest point. (a) SURF blob that is projected onto 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.

At this point, our goal is to perform an efficient and effective time-aware description of the previously detected space-temporal TRoF blobs, with low-memory footprint. With respect to 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 blob-centralized temporal $[x, t]$- and $[y, t]$-planes. As for effectiveness, we suggest the use of SURF [11] 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 [30]) can alternatively be used as well.

Figures 3.5(a-c) depict 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. Figure 3.5(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 a space-temporal scale of $2 \times \sigma_{st}$.

As previously mentioned, alternatively, each plane of interest can be described by a HOG block, divided — for instance — into 4×4 inner cells. Fig. 3.6(a-c) depict each

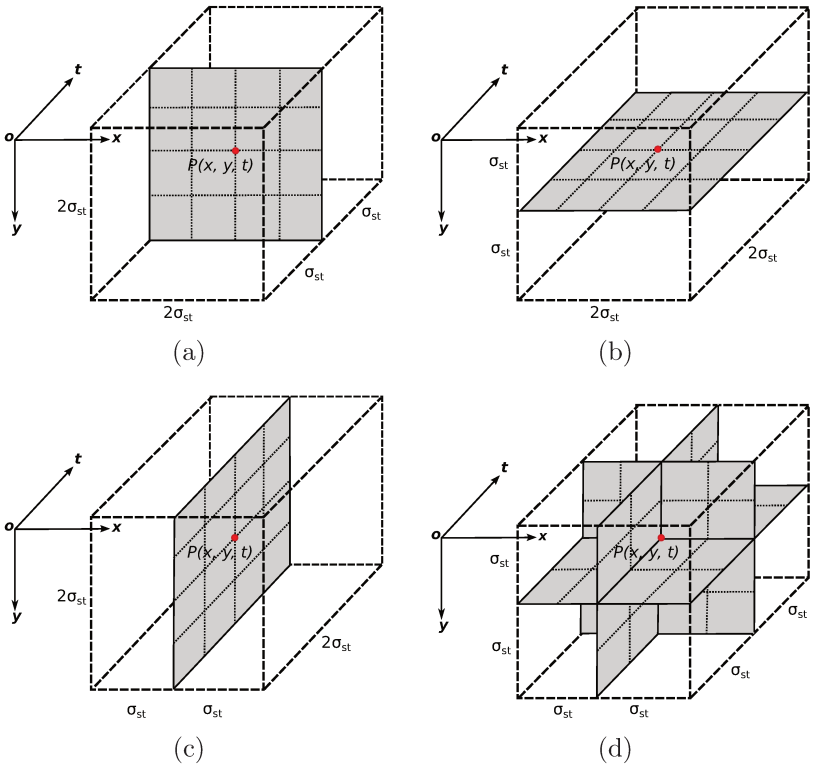


Figure 3.6: HOG-based TRoF-described blob planes. The solid gray rectangles are HOG description blocks, which are all centered at the position $P(x, y, t)$, and present a space-temporal scale of $2\sigma_{st}$. Each HOG block is divided into 4×4 inner cells, which are represented by internal dashed rectangles. P and σ_{st} come from a formerly detected interest point. (a) HOG block that is projected onto the $[x, y]$ plane. (b) HOG block that is projected onto the $[x, t]$ plane. (c) HOG block that is projected onto the $[y, t]$ plane. (d) Resulting space-temporal structure, which is formed by the union of the three HOG blocks.

one of these HOG blocks, in the form of solid gray rectangles. As one might observe, each rectangle is properly divided by 4×4 dashed subrectangles, which represent the HOG inner cells. Fig. 3.5(d) depicts the structural union of these three HOG blocks. For a low-memory footprint, we can limit the number of gradient histogram bins that are calculated in each HOG cell to four. Thus, each HOG block shall deliver four values for each one of its 4×4 inner cells, leading to a total of 64 description values, in a similar fashion to the 64-dimensional SURF blobs.

With the intent to register eventual correlations among the three flat SURF (or eventually HOG) blobs, that could be helpful to distinguish sensitive and non-sensitive material, we propose to 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 TRoF descriptor outputs a set of 192-dimensional feature vectors, for every target video stream or particular video snippet of interest.

For the sake of illustration, Figures 3.7(a-c) depict the visual content of the voxels that are described in each one of the three orthogonal planes of an eventually detected TRoF blob. Figure 3.7(a) contains the voxels belonging to the $[xy]$-plane, which — by being purely spatial — is the only one that is visually intelligible to humans. Figure 3.7(b),

(a) (b) (c)

Figure 3.7: 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 [11] (or alternatively HOG [30]) descriptor.

Table 3.4: Synthetic videos created to demonstrate the TRoF detection capabilities. The videos depict moving particles at different scales, which perform basic trajectories (e.g., vertical, horizontal, diagonal, zig-zag), along with static elements.

Video	Content	Depiction
horizontal	Two different-size circles that move independently and horizontally.	Figure 3.8
vertical	Two different-size circles that move independently and vertically.	Figure 3.9
diagonal	Two different-size circles, and two star-shape objects, separated by a diagonal static line. The circles move independently over the line, while the stars are static.	Figure 3.10
zig-zag	One star-shape object that moves in varied directions.	Figure 3.11

in turn, contains the voxels belonging to the $[xt]$-plane, while Figure 3.7(c) contains the voxels described in the $[yt]$-plane. We consider only these three images for applying a SURF (or alternatively HOG) descriptor.

3.2.3 TRoF Detection Capability

In order to visualize the quality of the TRoF detection process, we created four synthetic videos that depict moving particles at different scales, performing basic trajectories (e.g., vertical, horizontal, diagonal, and zig-zag), along with static elements. In Table 3.4, we describe the content of these videos, while in Figures 3.8–3.11, we depict the TRoF interest points that were detected along four frames of each one of these videos.

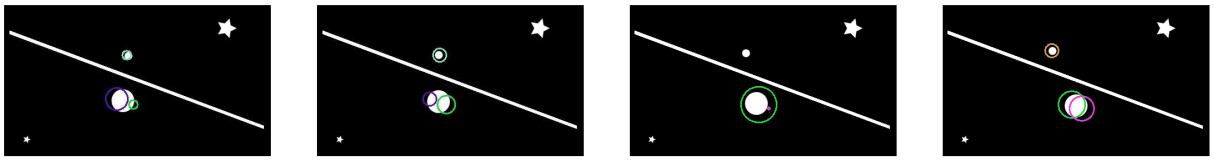
As one might observe, within these videos, the white circles, line, and stars correspond to the original video content, prior to the detection process. The colored circumferences, in turn, refer to the detected space-temporal blobs. As expected, the TRoF detector pays more attention to the moving objects, and describes their space-temporal neighborhood with scale invariance.

(a) frame 1001 (b) frame 1004 (c) frame 1008 (d) frame 1016

Figure 3.8: TRoF blob detection on four frames sampled from the *horizontal* video. The video depicts two white circles that have distinct scales and move horizontally. Colored circumferences refer to the detected space-temporal blobs.

(a) frame 969 (b) frame 977 (c) frame 986 (d) frame 1003

Figure 3.9: TRoF blob detection on four frames sampled from the *vertical* video. The video depicts two white circles that have distinct scales and move vertically. Colored circumferences refer to the detected space-temporal blobs.



(a) frame 746 (b) frame 750 (c) frame 758 (d) frame 768

Figure 3.10: TRoF blob detection on four frames sampled from the *diagonal* video. The video depicts two white circles that have distinct scales and move diagonally. All other white elements are static. Colored circumferences refer to the detected space-temporal blobs.

(a) frame 523 (b) frame 525 (c) frame 528 (d) frame 532

Figure 3.11: TRoF blob detection on four frames sampled from the *zig-zag* video. The video depicts one star-shape white object that moves in varied directions. Colored circumferences refer to the detected space-temporal blobs.

3.3 Final Remarks

The application scenarios of sensitive-video classification often require the analysis of myriad motion pictures (revealing a big-data nature), which are easily shared over the Internet, and cheaply disclosed by ubiquitous mobile devices (attesting an increasingly high data pervasiveness). Furthermore, sensitive concepts (such as violence, or pornography)

are usually subjective; in most cases, there is not a universal consensus on what is inappropriate, and in which situations. To make things harder, there are situations in which the detection of sensitive content is urgent. That happens, for instance, in surveillance scenarios, in which the fast identification of violent acts may be determinant for saving lives, or in Forensic scenarios, in which the fast identification of child pornography, among millions of files, may allow catching red-handed criminals.

For those reasons, computer-aided sensitive-video classification is a challenging and still open problem. Increasing effectiveness is always a goal; for that, we bet on using time-aware strategies, since it has long been proven that space-temporal approaches improve the final system video classification accuracy [83, 14, 90, 84]. For dealing with the big-data and urgency aspects of sensitive-video classification, we focus on designing fast solutions. For dealing with the pervasiveness aspect, in turn, we focus on designing solutions that present a low-memory footprint; such solutions are more suited to run locally, on mobile devices, thus benefiting from their ubiquity. Lastly, for mitigating the target-concept subjectivity, we suggest designing more generalizable and concept-independent solutions, by means of BoVW-based approaches.

In this vein, in this chapter, we presented an end-to-end BoVW-based time-aware pipeline for sensitive-video classification, whose internal components were selected with the aim of providing small processing time, and low-memory footprint. The presented approach is of general purpose, in the sense that it can be easily adapted to other binary classification (positive vs. negative) sensitive content problems (e.g., violence, pornography, gore scenes, child abuse, etc.). In addition, the efficiency gains strongly rely upon the low-level video description, thanks to the use of TRoF — a novel space-temporal interest point detector, and video descriptor — which was also introduced in this chapter. Tables 3.1 and 3.3 summarize the parameters of the pipeline, and of TRoF, respectively.

In the next two chapters, we validate the pipeline and the TRoF video descriptor, for both pornography (in Chapter 4) and violence (in Chapter 5) sensitive concepts.

Chapter 4

Pornography Classification: Experiments

In this chapter, we validate the BoVW- and TRoF-based sensitive-video classification pipeline that was introduced in Chapter 3, for the particular case of pornography detection. We compare the pipeline to commercial pornography detection solutions, and to other BoVW-based solutions that rely upon either a well-established still-image descriptor (namely, SURF [11]), or state-of-the-art space-temporal video descriptors (namely STIP [57], and Dense Trajectories [93]).

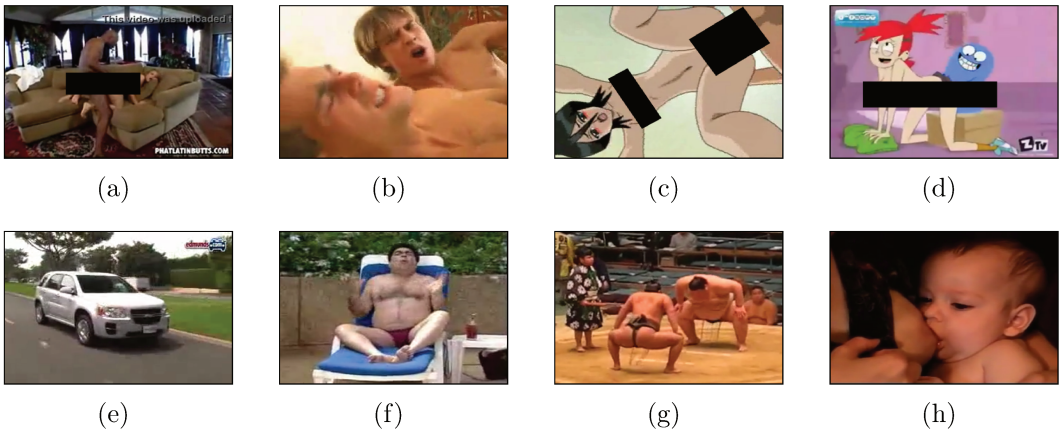
For that, in Section 4.1, we explain the adopted experimental setup, in terms of dataset, experimental protocol, metrics, available commercial solutions, BoVW-based parametrization, and implementation details. In the sequence, in Section 4.2, we report the experimental results, while in Section 4.3, we present some final remarks.

4.1 Experimental Setup

As explained in Section 2.2, previous work in the pornography classification literature presented limited validation, with no standardized datasets or metrics, except for the published methods in [19, 18, 8, 84, 90], which used the Pornography-800 dataset [8], with 800 videos. Hence, aiming at providing a larger standardized validation benchmark, we augmented that dataset to 2,000 videos, of which 1,000 are pornographic, and 1,000 are non-pornographic. In Section 4.1.1, we introduce this new dataset. Afterwards, in Section 4.1.2, we explain the experimental protocol and the metrics we use to evaluate the results, while in Section 4.1.3, we present the third-party solutions we compare to the proposed pipeline. Next, in Section 4.1.4, we detail the BoVW-based experimented solutions, which follow the sensitive-video classification pipeline that was introduced in Chapter 3.

4.1.1 Pornography-2K Dataset

The Pornography-2K dataset is an extended version of the Pornography-800 dataset, originally proposed in [8]. The new dataset — introduced in this work — comprises



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

(e) (f) (g) (h)

Figure 4.1: Frames sampled from the Pornography-2K dataset. On the top row, we show representative sensitive content, including pornographic cartoons. The black censor boxes were added by us, in the understanding that this text 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 fail in labeling samples (c-d), (f-h). In (c-d), we do not have live-action skin, despite of having pornographic material. In (f-h), we have non-pornographic cases with plenty of body exposure.

nearly 140 hours of 1,000 pornographic, and 1,000 non-pornographic videos, which vary from six seconds to 33 minutes.[1]

Concerning the acquisition of pornographic material, unlike Pornography-800 [8], we did not restrict ourselves to pornography-specialized websites. Instead, we also explored general-public and general-purpose video networks[2], 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 the non-pornographic content, we proceeded similar to Avila et al. [7]. 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). Figure 4.1 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.

[1]This dataset is the result of a joint effort made by professors Anderson Rocha, Siome Goldenstein, and Eduardo Valle, along with several other colleagues, namely Dr. Sandra Avila, Dr. Vanessa Testoni, Mauricio Perez, Daniel Moraes, and the author of this thesis.

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

4.1.2 Experimental Protocol and Metrics

In face of the 2,000-samples Pornography-2K dataset, and due to the very time- and resource-consuming experiments, we apply a variation of the 5×2-fold cross-validation protocol [38] for data folding and validation, with less folds, which we refer to as 3×2-fold protocol. It consists of randomly splitting the dataset into two same-size class-balanced folds, three times. In each time, training and test sets are switched, leading to six independent experiments, for a given classifier. Additionally, we submit the exact same six folds to each candidate classifier, allowing us to perform proper paired statistical tests later on. In this work, we employ the non-parametric pairwise Wilcoxon signed-rank test with Bonferroni's correction [35], whenever it is convenient to compare different classifiers with some statistical confidence.

For assessing the performance of the pornography classifiers, we report the *normalized accuracy* rate (ACC), and the F_2 *measure* (F_2). Prior to explaining these metrics, we need to define some basic concepts, namely *true positives*, *false positives*, *true negatives*, *recall*, *specificity*, and *precision*. For the sake of analogy, consider the test dataset as a sample space, whose videos are elements. Violent videos are positive elements, and non-violent videos are negative elements. Let $\#positive$ be the quantity of positive elements, and $\#negative$ be the quantity of negative ones.

True positives are the elements that a classifier labels as positive, and that are really positive. Let $\#true\_positive$ be the quantity of true positives, on the occasion of evaluating a classifier. False positives, on the contrary, are the elements that a classifier labels as positive, but are negative. Let $\#false\_positive$ be the quantity of false positives. True negatives, in turn, are the elements that a classifier labels as negative, and that are really negative. Let $\#true\_negative$ be the quantity of true negatives.

Recall, also known as true positive rate (TPR), expresses how good is a classifier, in identifying the positive elements of a sample space. It is given by Equation 4.1.

$$recall = TPR = \frac{\#true\_positive}{\#positive}. \tag{4.1}$$

Specificity, also known as true negative rate (TNR), expresses how good is a classifier, in identifying the negative elements of a sample space. It is given by Equation 4.2.

$$specificity = TNR = \frac{\#true\_negative}{\#negative}. \tag{4.2}$$

Finally, precision expresses how many elements are truly relevant (e.g., positive), among the ones that a classifier identifies as such. It is given by Equation 4.3.

$$precision = \frac{\#true\_positive}{\#true\_positive + \#false\_positive}. \tag{4.3}$$

Now, we are ready to present classification accuracy (ACC) and F_2 measure (F_2). ACC is the mean of TPR and TNR, as depicted in Equation 4.4. It tells us the hit rate of the methods, regardless of the class labels. A higher accuracy indicates a higher capability of

separating the target videos in pornographic and non-pornographic samples.

$$ACC = (TPR + TNR)/2. \tag{4.4}$$

F_2, in turn, is the weighted harmonic mean of recall and precision, which gives twice more weight to recall than to precision, by means of a $\beta = 2$ parameter. Equation 4.5 depicts the original F_β formula:

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall}, \tag{4.5}$$

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

4.1.3 Third-party Pornography Classifiers

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 [86], Snitch Plus [49], PornSeer Pro [97], and NuDetective [74].

MediaDetective and Snitch Plus are both off-the-shelf commercially available programs[3]. NuDetective, in 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, nipples, etc.).

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.

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

4.1.4 BoVW-based Pornography Classifiers

The proposed video classification pipeline, introduced in Chapter 3, is evaluated through different techniques. Specifically, we explore various methods of low-level local video description. In this section, we first describe the BoVW-based experimental setup followed by a brief detailing of the local descriptors we apply to our experiments.

In accordance with the pipeline that was presented in Chapter 3, we first pre-process the dataset, by resizing the video frame resolution to 100 thousand pixels, keeping the original aspect ratio (i.e., we use $fr = 100,000$ pixels). For this task, we use a cubic image interpolation.

In the sequence, regardless of the low-level descriptors (which are d_f-dimensional), we apply PCA to reduce by half their dimensionality (i.e., we use $p_f = \frac{d_f}{2}$), as it is done in the literature [93, 79, 71][4] For the calculation of the PCA transformation (reference eigenvalues and eigenvectors), we randomly sample 10 million descriptors from the training set, with half of them coming from the negative set, and the other half coming from the positive set (i.e., we use $k_p = 10,000,000$). These descriptors are then aggregated into an image/video-level signature.[5]

In order to make the comparisons fair, we use the same mid-level representation for all evaluated techniques. Therefore, we follow the pipeline recommendation, and extract Fisher Vectors [71], with the implementation provided by the VLFeat C++ API [92]. The visual codebook is modeled with a GMM, whose parameters are estimated over 10 million randomly sampled PCA-reduced descriptions (i.e., $k_c = 10,000,000$, with half coming from the positive training set, and half coming from the negative training set), by means of VLFeat C++ API [92]. By default, we use 256 Gaussians, as suggested in [71] (i.e., $c_{gmm} = 256$).

In the high level, classification is performed by SVM classifiers, using the LIBLINEAR library [41]. We apply grid search to find the best c-SVM parameter, during training. In Table 4.1, we summarize the values of the parameters that define the experimental setup.

Speeded-Up Robust Features (SURF)

To provide a controlled baseline for the space-temporal techniques, we extract SURF descriptors [11], which operate over static images only, with the OpenCV C++ API [17].

Thereby, for the sake of processing time, we use the `I`-frames from the video footage, which are extracted with the FFmpeg library [13]. Next, we discard 10% of the image borders to remove possible watermarks. 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 (baseline referred to as SURF-MJV). It means that the temporal information is incorporated at the high-level stage only.

[4]For the sake of additional investigation, we report in Appendix A the impact of variating the amount of PCA dimensionality reduction over the proposed TRoF-based solution. According to the results, we verify that the classification accuracy is not overly sensitive to the exact number of PCA components, even if the associated cumulative data variance is as low as 50%.

[5]For this task, and for the previous one (of resizing), we use the OpenCV C++ API [17].

Table 4.1: Parameter values of the experimented pornographic video classification pipeline. The dimensionality of low-level descriptions is given by d_f, and depends upon the chosen video description technique.

Parameter	Value	Meaning
f_r	$100,000$	Resolution, in pixels, to which the video frames are reduced.
p_f	$\frac{d_f}{2}$	Dimensionality of the low-level descriptions, after PCA reduction.
k_p	$10,000,000$	Quantity of descriptions sampled for PCA transformation calculation.
k_c	$10,000,000$	Quantity of PCA-reduced descriptions sampled for GMM estimation.
c_{gmm}	256	Quantity of GMM-codebook component Gaussians.

We also propose adding temporal information at the mid-level stage (baseline referred to as SURF-MLP), by pooling the mid-level features over the entire video.

With SURF, by definition, the low-level descriptions are 64-dimensional (i.e., $d_f = 64$). As a consequence, after the PCA reduction (with $p_f = \frac{d_f}{2} = 32$), and considering $c_{gmm} = 256$, the mid-level SURF-based Fisher Vectors are of length $2 \times p_f \times c_{gmm} = 16,384$ dimensions.

Space-Time Interest Points (STIP)

STIP [57] was the first local descriptor designed for analyzing the video space-time. Roughly speaking, the STIP detector [56] 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 of Laptev [57], using default values.

With STIP, the low-level descriptions are 162-dimensional (i.e., $d_f = 162$). As a consequence, after the PCA reduction (with $p_f = \frac{d_f}{2} = 81$), and considering $c_{gmm} = 256$, the mid-level STIP-based Fisher Vectors are of length $2 \times p_f \times c_{gmm} = 41,472$ dimensions.

Dense Trajectories (DTRACK)

Dense Trajectories represent the current state of the art in the field of time-aware local descriptors. In general terms, the dense trajectories [93] describe movement by 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 (MBH).

To extract the dense trajectories from the video files, we use the code provided by Wang et al. [93], with default values. It is worth to mention that, to the best of our knowledge, dense trajectories have never been applied to the task of pornography classification.

Table 4.2: Parameter values of the experimented TRoF detector. Parameter s is measured in pixels. The remaining parameters regard quantities.

Variable	Value	Meaning
o	4	Quantity of analyzed space-temporal scale octaves.
s	4	Initial video space-time sampling step.
c	250	Quantity of frames within each analyzed integral video.
b	3,000	Quantity of extracted interest blobs per integral video.

With DTRACK, the low-level descriptions are 426-dimensional (i.e., $d_f = 426$). As a consequence, after the PCA reduction (with $p_f = \frac{d_f}{2} = 213$), and considering $c_{gmm} = 256$, the mid-level DTRACK-based Fisher Vectors are of length $2 \times p_f \times c_{gmm} = 108,800$ dimensions.

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 SURF descriptors to represent the TRoF blob content (please refer to Section 3.2).

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 fastly, we sample one in every four video voxels, in all directions (i.e., $s = 4$), and apply four space-temporal scale octaves (i.e., $o = 4$), to perform the Hessian calculations.

Finally, we extract 3,000 blobs at every 250 integral video frames ($b = 3,000$). Preliminary experimental results showed that 250 frames, four voxels, four octaves, and 3,000 blobs represent a good compromise between effectiveness and efficiency. In Table 4.2, we summarize the chosen values for TRoF detector's four parameters.

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 and exploratory analysis.

With TRoF, the low-level descriptions are 192-dimensional (i.e., $d_f = 192$). As a consequence, after the PCA reduction (with $p_f = \frac{d_f}{2} = 96$), and considering $c_{gmm} = 256$, the mid-level TRoF-based Fisher Vectors are of length $2 \times p_f \times c_{gmm} = 49,152$ dimensions.

TRoF is currently implemented in C++, and relies upon the OpenCV C++ API [17].

4.2 Results

In Table 4.3, we present the results for pornographic video classification on the Pornography-2K dataset. We report the normalized accuracy rate (ACC) and the F_2 measure (F_2), both averaged over six folds. Additionally, we report the average true positive (TPR) and

Table 4.3: Results of video classification on the Pornography-2K dataset. We report the average performance on 3 × 2 folds. In all cases, the standard deviation is smaller than 0.1.

	Solution		TPR (%)	TNR (%)	ACC (%)	F_2 (%)
Third-party	Snitch Plus [49]	skin	43.43	91.30	67.37	47.88
	MediaDetective [86]		63.30	80.40	71.85	65.53
	NuDetective [74]		59.70	85.50	72.60	62.94
	PornSeer Pro [97]		74.10	84.10	79.10	75.61
BoVW-based	SURF-MJV [11]	static	94.50	82.80	88.65	92.34
	SURF-MLP [11]		91.30	91.73	91.52	91.38
	STIP [56]	temporal	92.77	94.80	93.78	93.14
	DSTIP [56]		94.13	94.27	94.20	94.16
	DTRACK [93]		95.37	96.20	95.78	95.52
	TRoF		93.00	93.80	93.40	93.54
	DTRoF		95.10	95.87	95.48	95.25

TPR: true positive — TNR: true negative rate — ACC: accuracy — F_2: F_2 measure

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

Figure 4.2: Failure examples of the skin-detection-based solutions. Frames (a-d) were sampled from four videos of the Pornography-2K dataset, in which the third-party solutions fail, in opposition to the success of the BoVW-based approaches. In (a-b), we present false positive examples. In (c-d), we present false negative examples. The black censor boxes were added by us, in the understanding that this text can reach a broad audience, including underage readers; they are not present in the original material.

average true negative (TNR) rates, to give the reader a broader view of the classification results.

4.2.1 Third-Party Solutions

As one might observe in Table 4.3, 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. Figure 4.2 depicts some frames that were sampled from four videos of the Pornography-2K dataset, in which the skin-detection-based solutions fail, in opposition to the success of the BoVW-based approaches. Figures 4.2 (a-b) depict false positive cases, while Figures 4.2 (c-d) depict false negative cases.

Table 4.4: Pairwise comparison between TRoF and the other approaches that are not simultaneously dense and space-temporal. We report the statistical tests for all 3×2 folds, considering ACC, and using the paired Wilcoxon test with Bonferroni's correction.

		STIP	SURF-MLP	SURF-MJV
TRoF	p-value	1.000	0.013	0.013
	conclusion	•	✔	✔

• not statistically different ✔ TRoF is better

Table 4.5: Pairwise comparison between DTRoF and the other dense space-temporal approaches. We report the statistical tests for all 3×2 folds, considering ACC, and using the paired Wilcoxon test with Bonferroni's correction.

		DTRACK	DSTIP
DTRoF	p-value	1.000	0.045
	conclusion	•	✔

• not statistically different ✔ TRoF is better

The strength of BoVW-based techniques is further prominent 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 45% and 68% with respect to ACC and F_2, respectively.

4.2.2 BoVW-Based Solutions

Among the BoVW-based solutions, the use of space-temporal local video descriptors lead 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.

In Table 4.4, we present the statistical comparison between the sparse application of TRoF (which is one contribution of this work), and each one of the following approaches: SURF-MJV, SURF-MLP, and STIP. For the sake of this analysis, we left the better-performing dense space-temporal approaches out. As expected, TRoF — which is space-temporal — presents better ACC, with statistical difference and 95% of confidence, when compared to the still-image approaches (SURF-MJV and SURF-MLP). In addition, TRoF and STIP are not statistically different, in spite of TRoF being more efficient, as we report in Section 4.2.3.

The use of dense space-temporal video description also leads to more effective classifiers. For instance, the three best solutions (DSTIP, DTRACK, and DTRoF) rely upon a dense description of the video space-time. In Table 4.5, we present the statistical comparison between the dense application of TRoF (DTRoF), and each one of the other two experimented dense space-temporal approaches (DSTIP and DTRACK). As one might

observe, DTRoF presents better ACC, with statistical difference and 95% of confidence, when compared to DSTIP, while it is not statistically different from DTRACK, in spite of being more efficient than both counterparts, simultaneously in terms of computational time and memory footprint. Please refer to Section 4.2.3 for more details about efficiency.

4.2.3 Efficiency Results

Despite of the higher effectiveness, space-temporal and dense strategies often lead to inefficient classifiers, specially with respect to the 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, at least with the mobile configurations available in the market as of 2016.

Figure 4.3(a) depicts the correlation between the accuracy and the computational time that is spent to perform an 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 three hours 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 a 2-GHz 12-core Intel(R) Xeon(R) processor (E5-2620), with 24 GB of RAM. Figure 4.3(b) correlates the F_2 measure with the computational time.

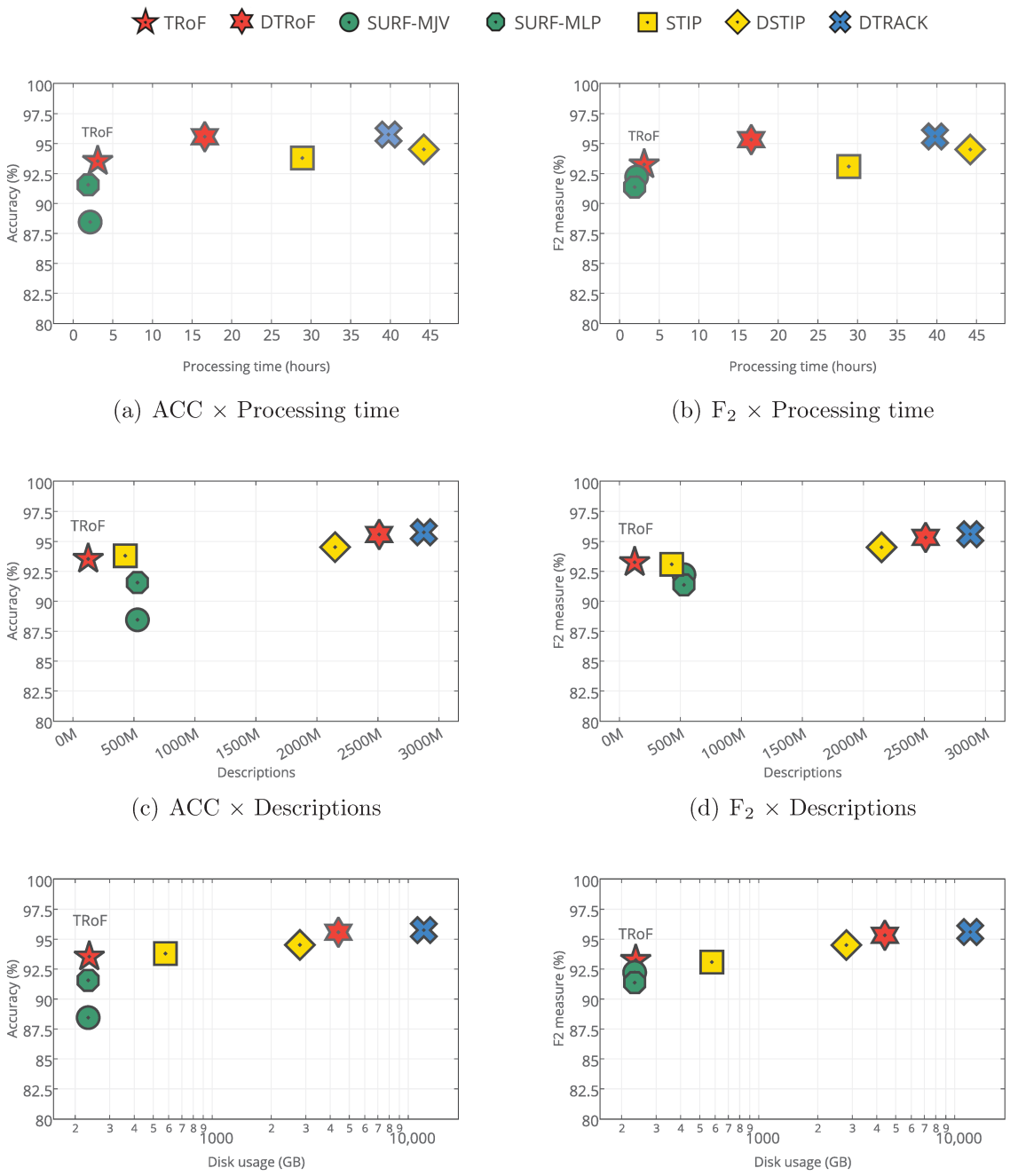
Likewise, Figure 4.3(c) shows the correlation between the accuracy and the quantity of descriptors extracted from the entire Pornography-2K dataset, for each BoVW-based solution. Figure 4.3(d), in turn, correlates the quantity of descriptors with the F_2 measure.

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. Figure 4.3(e) depicts the correlation between accuracy and disk usage, in a lin-log chart, for a better representation. Figure 4.3(f) depicts the correlation between F_2 measure and disk usage.

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.

In Figure 4.4, we detail 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 three hours 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).



(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

Figure 4.3: Performance of BoVW-based classifiers on the Pornography-2K dataset, putting effectiveness (vertical axes) in perspective with efficiency (horizontal axes). On the left, effectiveness refers to classification accuracy, while on the right, it refers to the F_2 measure. On the top row, efficiency regards computational time spent to classify over three hours of video footage (same system for all methods). On the middle row, efficiency concerns the number of descriptors extracted for the entire dataset. On the bottom row, efficiency refer to (log scale) the disk storage space for the entire dataset. In all charts, the best solutions are at the top-left corner.

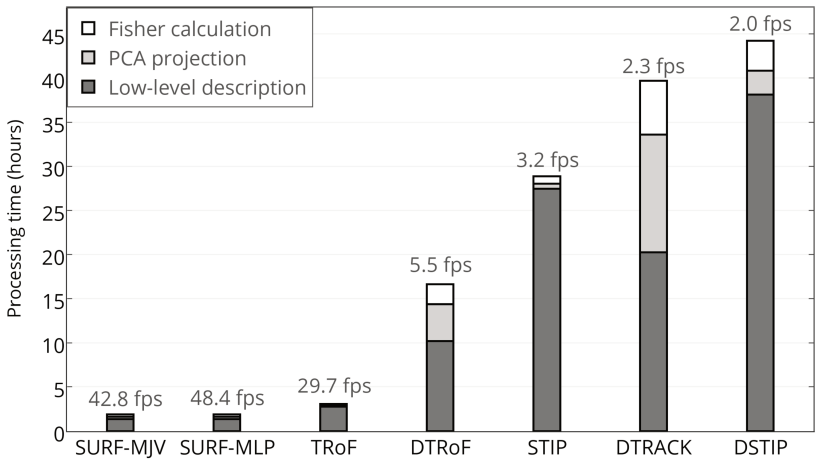


Figure 4.4: Breakdown of the processing time spent by each BoVW-based classifier. The computational times refer to the amount of time used to classify over three hours of randomly chosen video footage, under the same system. That time is divided among three subtasks: low-level video description, PCA 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.

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 (through 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.

4.3 Final Remarks

Pornographic video classification is a hard problem, in which traditional methods often employ still-image techniques: they label frames individually, often supported by skin detectors, prior to a global decision. Frame-based approaches, however, ignore significant cogent information brought by motion, leading to a reduction in their effectiveness (in terms of accuracy and F_2 measure). That justifies the use of space-temporal video descriptors, whose most representative alternatives from the literature were evaluated in this chapter.

Indeed, to the best of our knowledge, it was the first time that the dense application of STIP [56] and Dense Trajectories [93] were evaluated to solve the problem of pornography classification. Similarly, for the first time, the Fisher Vector representation was used as a mid-level stage in pornographic content classifiers.

Experiments confirmed that the incorporation of space-temporal information leads to more effective video-pornography classifiers, and also ratified that a dense low-level video

description increases the system effectiveness (in terms of accuracy and F_2 measure), but at prohibitive reductions in efficiency (in terms of computational time and memory footprint). Such drawback makes it impractical to apply dense strategies or other conventional space-temporal approaches on hardware-limited mobile devices, such as tablets and smartphones.

In such context, TRoF reveals itself as an interesting alternative for dealing with the effectiveness vs. efficiency tradeoff, in spite of not being statistically different from the more resource-consuming STIP alternative. Moreover, a dense application of TRoF is also noteworthy: it performs with no statistical difference from the Dense Trajectories, in spite of presenting three times less memory footprint, and being twice as faster.

Finally, the evaluation of the proposed sensitive-video classification pipeline took interesting steps in the direction of advancing the state of the art in pornography classification. The first is the acquisition of the Pornography-2K dataset, a new challenging benchmark, with 140 hours of video footage. The second is the evaluation of third-party classifiers. Among such solutions, we included two commercial programs, which are based upon skin detectors. We verify, through experimentation, that they are far from being reliable, in face of the task at hand.

Chapter 5

Violence Classification: Experiments

In this chapter, we validate the BoVW- and TRoF-based sensitive-video classification pipeline that was introduced in Chapter 3, for the particular case of violence detection. We analyze the pipeline performance in contrast to other BoVW-based solutions that rely upon either a well-established still-image descriptor (namely HOG [30]), or space-temporal video descriptor (namely STIP [57]), with the aim of verifying whether or not the proposed solution is able to efficiently use video temporal information (concerning hypothesis *H1*). Additionally, for the sake of investigation, we compare the proposed approach with the standout works of Mironică et al. [63] (who currently report the best results over the MediaEval 2013 VSD dataset [33]), and of Derbas and Quénot [36] (who officially took first place in the MediaEval 2013 subjective violence classification competition [33]).

For that, in Section 5.1, we explain the adopted experimental setup, in terms of dataset, experimental protocol, and metrics (which are inherited from the MediaEval 2013 VSD task [33]), in addition to BoVW-based parametrization, and implementation details. Thereafter, in Section 5.2, we report the experimental results, while in Section 5.3, we present some final remarks.

5.1 Experimental Setup

As explained in Section 2.3, in the last few years, progress in violence detection has been quantified mainly due to the MediaEval VSD task [80, 34, 33], which provides a common groundtruth and standard evaluation protocols to the scientific community. To benefit from such advantages, we use the MediaEval benchmark for conducting the experiments. Hence, in Section 5.1.1, we present some details of the MediaEval 2013 dataset [33], which refers to the last VSD task edition that had evaluated violence classification (further editions aimed at violence localization [80]). Next, in Section 5.1.2, we explain the competition experimental protocol, and the metrics used to evaluate the results, while in Section 5.1.3, we present the BoVW-based experimented solutions, which follow the sensitive-video classification pipeline that was introduced in Chapter 3.

Table 5.1: MediaEval 2013 VSD dataset summary. The 25-title dataset is divided into an 18-title *Training* set, and a seven-title *Test* set. The competition provides annotations for segmenting the titles into shots, which are individually labeled as violent (content that one would not let an eight-year old child see) or non-violent. Nearly 20% of the shots are violent.

	Title	#shots	violent shots (#)	violent shots (%)
Training	01. Armageddon	3,562	466	13.08
	02. Billy Elliot	1,236	66	5.33
	03. Dead Poets Society	1,583	23	1.45
	04. Eragon	1,663	453	27.23
	05. Fight Club	2,335	516	22.09
	06. Harry Potter V	1,891	329	17.39
	07. I am Legend	1,547	497	32.12
	08. Independence Day	2,652	640	24.13
	09. Kill Bill	1,597	650	40.70
	10. Leon	1,547	437	28.24
	11. Midnight Express	1,677	239	14.25
	12. Pirated of the Caribbean I	2,534	670	26.44
	13. Reservoir Dogs	856	304	35.51
	14. Saving Private Ryan	2,494	1195	47.91
	15. The Bourne Identity	1,995	257	12.88
	16. The Sixth Sense	963	53	5.50
	17. The Wickerman	1,638	193	11.78
	18. The Wizard of Oz	908	22	2.42
	Total	**32,678**	**7,010**	**21.45**
Test	01. Fantastic Four I	2,002	717	35.81
	02. Fargo	1,061	256	24.12
	03. Forrest Gump	1,418	238	16.78
	04. Legally Blond	1,340	0	0.00
	05. Pulp Fiction	1,686	496	29.41
	06. The Godfather	1,893	198	10.45
	07. The Pianist	1,845	371	20.10
	Total	**11,245**	**2,276**	**20.24**

5.1.1 MediaEval 2013 Violent Scenes Detection Dataset

The MediaEval 2013 VSD dataset comprises 25 Hollywood movie titles of diverse genres, from extremely violent, to musical. Shot[1] segmentation is provided for all the movies — including the ones belonging to the test set — as a part of the dataset annotation, and the resulting segments are individually annotated as containing or lacking violent scenes, which "one would not let an eight-year old child see" [33]. The shot annotation had been carried out by seven human assessors, with varied ages and cultural backgrounds, and the shot segmentation had been obtained through a proprietary software.

In Table 5.1, we summarize the content of the MediaEval 2013 VSD dataset. The values were collected by [33]. Competitors and other interested people are supposed to purchase the movie titles at their own expenses. The MediaEval initiative provides only the annotations, which come separated into a training set, regarding 18 movies that are segmented into 2,678 shots, and a test set, comprising seven movies that are segmented into 11,245 shots. Approximately 20% of all shots are violent.

For the sake of illustration, Figure 5.1 depicts some violent frames from the MediaEval 2013 VSD dataset.

[1] A shot is a temporal sequence of frames that are captured in the same plane, by the same camera.

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

(e) (f) (g) (h)

Figure 5.1: Violent frames sampled from the MediaEval 2013 VSD dataset. All images are copyrighted, and therefore belong to the respective movie studios.

5.1.2 Experimental Protocol and Metrics

The VSD task motivation is the development of systems that may help users choose suitable titles for their children, by retrieving the most violent movie parts, for parental preview [33]. As a consequence, competitors' solutions are compared from the perspective of retrieval: the highest performing systems are the ones that return the largest number of violent shots, at the first positions of the top-k retrieved shots, properly ranked by violence classification confidence.

For achieving that, the MediaEval initiative suggests using the Mean Average Precision (MAP) at the 100 top ranked violent shots (MAP@100), as the official evaluation metric. Equation 5.1 shows the mathematical formula of MAP@k:

$$MAP@k = \frac{1}{q}\sum_{i=1}^{q} AP@k(i), \tag{5.1}$$

where k is the quantity of shots within the rank of retrieved shots ($k = 100$), and q is the quantity of system queries for obtaining ranked violent shot lists. In the VSD case, $q = 7$, which is the number of titles within the test set (i.e., each query is related to retrieving shots from a specific test movie title). $AP@k(i)$, in turn, is the average precision of the i-th query, when returning a k-shot ranked list, as it follows:

$$AP@k(i) = \frac{1}{k}\sum_{j=1}^{k} precision(i,j), \tag{5.2}$$

where $precision(i,j)$ is the system precision when retrieving the top-j violent shots, regarding the i-th query.

Relying upon the MAP@100 metric, the MediaEval 2013 VSD task adopts a straightforward protocol. Participants must report results over the seven-title test dataset, with a label (violent or non-violent) and a confidence classification score for each one the shots that are defined in the annotations. Clearly, the test dataset must not be used in any

Table 5.2: Parameter values of the experimented violent video classification pipeline. The dimensionality of low-level descriptions is given by d_f, and depends upon the chosen video description technique.

Parameter	Value	Meaning
f_r	$100,000$	Resolution, in pixels, to which the video frames are reduced.
p_f	$\frac{d_f}{2}$	Dimensionality of the low-level descriptions, after PCA reduction.
k_p	$1,000,000$	Quantity of descriptions sampled for PCA transformation calculation.
k_c	$1,000,000$	Quantity of PCA-reduced descriptions sampled for GMM estimation.
c_{gmm}	256	Quantity of GMM-codebook component Gaussians.

system training step. The best solutions are the ones that report the highest values of MAP@100. For assessing the MAP@100, the MediaEval initiative provides a Perl script for free, which we use in our experiments.

5.1.3 BoVW-based Violence Classifiers

Similar to the case of video pornography classification, we evaluate the proposed video classification pipeline — for the particular case of violence — through different techniques. Specifically, we explore various methods of low-level local video description. In this section, we first describe the BoVW-based experimental setup, and we next provide a brief detailing of the used local descriptors.

As explained in Chapter 3, we first preprocess the dataset, by resizing the video frame resolution to 100 thousand pixels, keeping the original aspect ratio. In the sequence, regardless of the low-level descriptors, we apply PCA to reduce by half their dimensionality. For the calculation of the PCA transformation, we randomly sample one million descriptors from the training set, with half of them coming from the negative set, and the other half coming from the positive set. These descriptors are then aggregated into a video-level signature. For such tasks, we use the OpenCV C++ API [17].

To make the comparisons fair, we use the same mid-level representation for all evaluated techniques. Therefore, we follow the proposed pipeline, and extract Fisher Vectors [71], with the implementation provided by the VLFeat C++ API [92]. The visual codebook is modeled with a GMM, whose parameters are estimated over one million randomly sampled PCA-reduced descriptions, with half coming from the positive training set, and half coming from the negative training set), by means of VLFeat C++ API [92]. By default, we use 256 Gaussians, as suggested in [71] (i.e., $c_{gmm} = 256$).

In the high level, classification is performed by SVM classifiers, using the LIBLINEAR library [41]. We apply grid search to find the best c-SVM parameter, during training.

In Table 5.2, we summarize the values of the parameters that define the experimental setup. They are slightly different from the pornographic case: for the sake of saving training processing and experimentation time, we sample one million features from the low-level description space (i.e., $k_p = k_c = 1,000,000$), in opposition to the ten million pornographic samples.

Histograms of Oriented Gradients (HOG)

To provide a controlled baseline for the use of TRoF, we extract HOG descriptions [30], which operate over static images only, with the OpenCV C++ API [17]. Thereby, for the sake of saving processing time, we use the `I`-frames from the video shots, which are extracted with the FFmpeg library [13]. Next, we discard 10% of the image borders, to remove possible watermarks. HOG descriptions 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.

Each patch is described by a single HOG block, which is divided into 4×4 HOG cells. Each cell is described by eight bins, leading to $4 \times 4 \times 8$ description values per patch. Hence, the obtained HOG feature vectors are 128-dimensional.

In addition, we add temporal information at the pipeline mid-level stage, by pooling the mid-level features over the entire video. Considering the 128 dimensions of HOG, allied with the PCA reduction of $p_f = d_f/2 = 64$, and $c_{gmm} = 256$, the mid-level HOG-based Fisher Vectors are of length $2 \times p_f \times c_{gmm} = 32,768$ dimensions.

Space-Time Interest Points (STIP)

For the sake of saving experimental time and also based on our previous results for pornography classification, we choose STIP [57] as the representative of well-established space-temporal local descriptors, instead of Dense Trajectories [93], which are very time- and memory-consuming.

For the experiments, we extract both sparse — i.e., 3D-Harris-detected (STIP) — and dense STIP (DSTIP) descriptors, with the code provided by Laptev [57].

Considering that STIP descriptions are 162-dimensional (i.e., $d_f = 162$), after the PCA reduction (with $p_f = d_f/2 = 81$), and employing $c_{gmm} = 256$, the mid-level STIP-based Fisher Vectors are of length $2 \times p_f \times c_{gmm} = 41,472$ dimensions.

Temporal Robust Features (TRoF)

In accordance with the chosen baseline static local descriptor (which is HOG), we employ HOG descriptors to represent the TRoF blob content (please refer to Section 3.2).

For detecting the three-dimensional blobs of interest, with the TRoF detector, we use the same configuration that is presented in Table 4.2, due to the reported good performance, in face of the pornography classification problem (please refer to Section 4.2.2, for details). We therefore calculate the integral video at every 250 frames of the target video (i.e., $c = 250$). For each obtained integral video, to swiftly describe a video, we sample one in every four video voxels, in all directions (i.e., $s = 4$), and apply four space-temporal scale octaves (i.e., $o = 4$), to perform the Hessian calculations. Thereafter, we extract 3,000 blobs at every 250 integral video frames ($b = 3,000$).

Finally, with TRoF, the low-level descriptions are 192-dimensional (i.e., $d_f = 192$), regardless of relying upon SURF or HOG. As a consequence, after the PCA reduction (with $p_f = d_f/2 = 96$), and considering $c_{gmm} = 256$, the mid-level TRoF-based Fisher Vectors are of length $2 \times p_f \times c_{gmm} = 49,152$ dimensions.

Table 5.3: Results on the MediaEval 2013 VSD dataset. All MAP and MAP@100 values were obtained with the competition evaluation tool [33].

Solution	Modalities	MAP@100	MAP
Mironică et al. [63]	audio & video	*	0.760
Derbas and Quénot [36]	audio & video	0.690	0.673
DHOG	video only	0.459	0.390
STIP	video only	0.541	0.465
DSTIP	video only	0.588	0.512
TRoF	video only	0.508	0.460

*Authors did not report MAP@100.

5.2 Results

In Table 5.3, we present the results for violence video classification on the MediaEval 2013 VSD dataset. We report the MAP@100 over the seven-title test dataset, in addition to the MAP@{*all shots*} (referred to as MAP).

Although not our initial aim, we compare the pipeline results with the works of Mironică et al. [63] (who report the best results over the MediaEval 2013 VSD dataset [33]), and of Derbas and Quénot [36] (who took first place in the MediaEval 2013 subjective violence classification competition [33], for the sake of investigation.

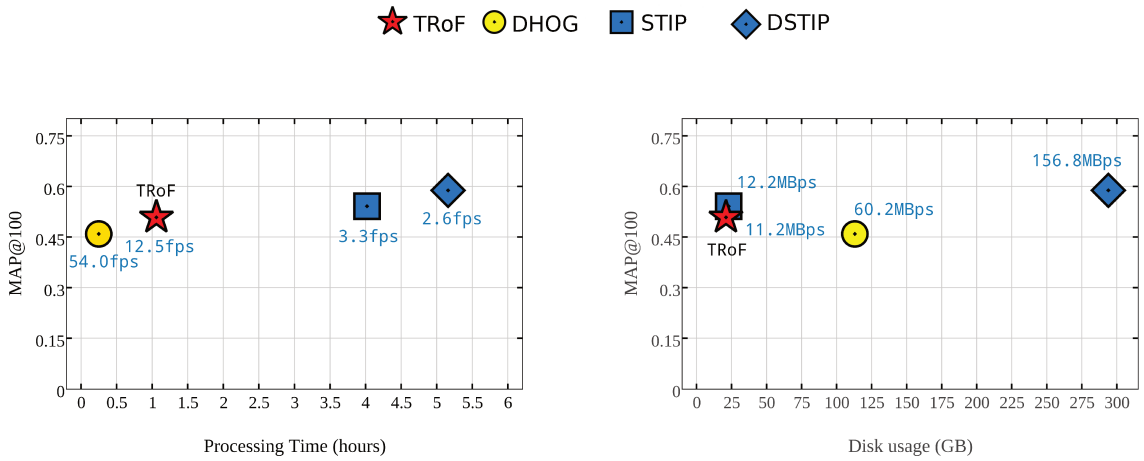
As explained in Section 2.3.3, Mironică et al. [63] give up using space-temporal features, in the particular case of the MediaEval 2013 VSD dataset, due to the high computational complexity, which makes such descriptors inefficient for large-scale collections. Instead, they apply global still-image description (based on HOG and color histograms), along with audio features (e.g., MFCC), which they claim being fast.

Derbas and Quénot [36], in turn, employ a combination of four descriptors, namely MFCC (auditory), SIFT and color texture (still-image aimed), and STIP.

In opposition to these works, we limited the solutions to the use of a single modality (either DHOG, STIP, DSTIP, or TRoF), in order to investigate how one may deal with the effectiveness vs. efficiency tradeoff. Single modal solutions presented reasonable classification effectiveness, with the dense space-temporal solution (DSTIP) presenting the highest MAP@100 (0.588), as expected. In the same direction, the still-image approach (DHOG) presented the worst results, confirming that the space-temporal information improves violence classification. TRoF, in turn, obtained a MAP@100 value of 0.508, however, it presents the best performance, in terms of processing time and memory footprint, as we shall demonstrate in the next section.

5.2.1 Efficiency Results

For investigating the required computational time, Fig. 5.2(a) depicts the correlation between MAP@100 and the computational time spent to classify a selected portion of 30 minutes of video content, for each experimented classifier. Internal values indicate the processing frame rate, in frames per second (fps). The highest the rate, the better



(a) MAP@100 × Processing time

(b) MAP@100 × Disk usage

Figure 5.2: Performance of BoVW-based classifiers on the MediaEval 2013 VSD dataset, putting effectiveness (vertical axes) in perspective with efficiency (horizontal axes). On the left, efficiency refers to the computational time spent to classify 30 minutes of video footage (same system for all methods). Internal values indicate the processing frame rate, in frames per second (fps). The higher the rate, the better. On the right, efficiency refers to the disk storage space for the seven-titles test dataset. Internal values indicate the amount of generated description in megabytes per second of footage (MBps). The smaller the amount, the better. In both charts, the best solutions are at the top-left corner.

the solution. As one might observe, TRoF leads to a rate of 12.5 fps, in spite of being space-temporal.

Likewise, for evaluating the strategies in terms of memory footprint, we also correlate MAP@100 with respect to the total disk space that is spent to store the low-level feature vectors of the seven-title test dataset, which comprises 15 hours of video footage. Fig. 5.2(b) shows the correlation between MAP@100 and disk usage. Internal values indicate the amount of generated description in megabytes per second of footage (MBps). The smaller the amount, the better.

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

In addition, although we do not have the proper time and disk usage measurements of the work of Derbas and Quénot [36], we can still infer their performance from the implemented STIP solution, because it is included in their adopted fusion strategy. STIP can be seen as a lower-bound to the time and disk usage of such solution. With respect to the work of Mironică et al. [63], even though they claim that their solution is fast, they do not report any efficiency results concerning the MediaEval dataset.

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

5.3 Final Remarks

Violent video classification is a problem that has gained attention from the scientific community, due to its relevance. Specially in the last few years, progress in the field has been quantified mainly due to the MediaEval VSD task [34, 33]. Among the proposed solutions, the typical approach relies upon multimodal video characterization, with the compulsory use of space-temporal descriptors as one of the employed modalities. That happens because it has long been proven that space-temporal descriptors — such as STIP and Dense Trajectories — improve the effectiveness of violence detectors.

In this vein, although the general perception in the literature dictates that space-temporal techniques are normally computationally expensive and present a high-memory footprint, the research on violence detection, in general, lacks proper performance evaluation.

Regardless of that, the fast detection of violent content is important in surveillance setups (in which, for instance, the real-time identification of violent events shall be determinant for saving lives), and in forensic scenarios (in which the fast identification of violent content among millions of files shall allow catching red-handed criminals). Moreover, if automated violence detection is transparently performed in low-memory devices (such as smartphones and tablets), it might ubiquitously protect audiences, without harming the user experience.

In this context, in this work, we evaluated the use of a sensitive-video classification approach (please refer Chapter 3), conceived for being not only effective, but also efficient, in spite of relying upon space-temporal video description. That is possible mainly due to the use of TRoF (please refer to Section 3.2), a novel and effective space-temporal interest point detector and video descriptor, which is computationally fast and presents low-memory footprint.

For filling the lack of performance evaluation, we report the performance of the proposed solution in face of both pornography (in Chapter 4) and violence classification problems. The results have shown that the TRoF usage allied with SURF yields a processing frame rate capacity of almost 30 fps, which is ideal for real-time video description. In addition, the alternative TRoF combination with HOG yields a processing frame rate capacity of nearly 12 fps: four times faster than STIP, and more than five times faster than Dense Trajectories.

The performance of TRoF is possible mainly due to two innovations. First, a four-variable space-temporal Hessian matrix, which uses a space-temporal standard deviation that is shared between space and time, for detecting the scale of interesting phenomena. And second, a fast description of the detected space-temporal interest point, which yields a compact description in $\Re^{192}$. As the shared space-temporal scale parameter is key for the performance gain, one might wonder when it is interesting to apply it. In our experience, we have learned that whenever we have a generalization problem (e.g., generalizing video motion to more general concepts, such as pornography and violence) this representation is appropriate. On the other hand, when we have a specialization problem (e.g., using video motion for detecting specific actions, such as walking, running, jumping, etc.), possibly an untangled representation for the space and time scales is more appealing.

This chapter closes the first part of this work, and is related to hypothesis *H1* (please refer to Section 1.1), which states that it is possible to efficiently use video temporal information for effective sensitive-content classification, by combining simplified space-temporal video interest-point detection and description, with entire-footage representation through a single feature vector. According to the presented results, we found strong evidence that such hypothesis is confirmed.

Part II

Sensitive-Content Localization

Chapter 6

From Many to One: Combining Multimodalities

Sensitive-content localization is the search problem of finding sensitive scenes within a video timeline. In other words, a system that performs sensitive-content localization must return the instants a video stream starts and ends to display sensitive content.

In this chapter, we introduce a high-level multimodal fusion approach to sensitive-video localization, which relies on the combination of different sensitive-snippet classifiers. Roughly speaking, a *snippet* is any video segment that is smaller than the entire video, and that preserves the temporal order of the scenes. It may start and it may end at any video time, as long as the starting time precedes the ending time. In addition, each snippet classifier may rely on a particular data modality (e.g., video frames, audio stream, video space-time, etc.) — that defines the multimodal nature of the solution.

The proposed fusion approach ultimately resulted in the filing of two patents, one in the Brazilian National Institute of Industrial Property (INPI) [5], and the other in the United States Patent and Trademark Office (USPTO) [6]. Moreover, it helped us to reach second place within the international MediaEval violence-detection 2014 competition (in the so-called generalization task, which considered webvideos [4]). Finally, the approach also resulted in a regular journal paper, currently in its final stages of preparation [65].

This chapter is related to hypothesis *H2* (please refer to Section 1.1), which states that it is possible to localize sensitive content within the video timeline, by means of the classification and fusion of temporal-overlapping video snippets. It aims at the goal of designing and developing effective methods for sensitive-content localization. For that, it is organized as follows. In Section 6.1, we explain the high-level multimodal fusion pipeline of sensitive-snippet classifiers, that was designed for sensitive-content localization. Thereafter, in Section 6.2, we detail the challenges and the solutions for performing snippet classification, as a basic requirement of the proposed solution. Finally, in Section 6.3, we present the final remarks of this chapter.

6.1 High-Level Multimodal Fusion of Snippet Classifiers

In the video-classification problem, formerly addressed in Part I, the solutions are supposed to attribute a label to an entire well-defined *video unit* (for instance, a label for an entire video *shot*, or a label for an entire video *file*). That makes the application of BoVW-based approaches straightforward: just establish a bag per video unit of interest, for a further label-prediction learning (in training system execution), or for a further discrete classification (in test system execution).

However, in the current content-localization problem, in which the solutions are supposed to point out when a stream starts or ceases to display inappropriate content, there is not such a clear definition of video unit of interest to be labeled. Hence, in face of such absence, how could one still benefit from the use of BoVW-based classification approaches? For instance, given the many possibilities of video segmentation (e.g., frames, shots, scenes, etc.), in what unit shall one pool the mid-level features, in order to provide bag labels that are more supportive of the task of content localization?

In this work, we tackle such problem by pooling and normalizing consecutive features, as long as they belong to a same fixed-length video segment (a.k.a., a snippet). The reasons for taking this path and also the means for doing it are discussed in Section 6.2. As a consequence of such decision, we can admit that we have available *sensitive-snippet classifiers*, as initial resources for performing sensitive-content localization. In addition, considering that each snippet classifier can rely on a particular data modality (e.g., video frames, audio stream, video space-time, etc.), we highlight the multimodal potential of the proposed solution.

Figure 6.1 depicts a flowchart overview of the method that is proposed for performing sensitive-content localization. Each rounded rectangular box is an activity, and the solid arrows represent the precedence of activities. Dashed arrows represent a simple flow of data. As one might observe, we suggest a late fusion of snippet classifiers.

As pointed out by Atrey et al. [3], late-fusion strategies have the advantage of offering easier scalability, regarding the addition or subtraction of classifiers, when compared to early-fusion solutions. Besides that, early-fusion strategies present the drawback of having to combine low-level features from different modalities (e.g., visual and auditory), which certainly present distinct types of representation (for instance, in terms of dimension, scale, data type, etc.). In opposition, late-fusion solutions combine decisions at the semantic level, hence dealing with the same type of representation (e.g., classification scores, distances to decision hyperplanes, etc.). Due to the manipulation of data in more akin domains, late-fusion alternatives are usually more straightforward to be performed.

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

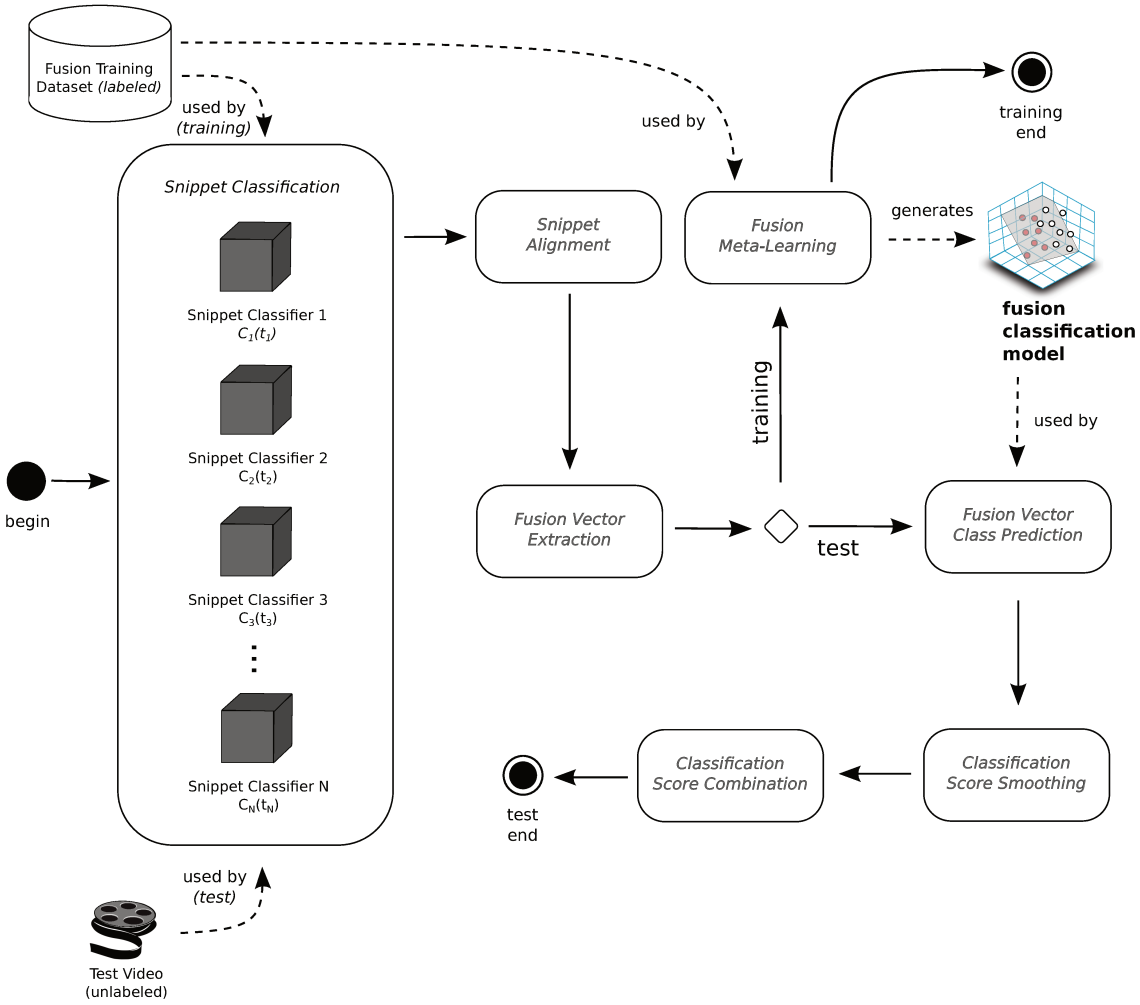


Figure 6.1: Sensitive-content localization method overview. Each rounded rectangular box is an activity, and the solid arrows represent the precedence of activities. Dashed arrows represent a simple flow of data. Depending on the type of system operation (training or test), the activity sequence may reach either the *training*, or the *test end*. The *Snippet Classification* activity is internally detailed, for depicting the use of N different *Snippet Classifiers*, as initial resources, which are properly represented as black boxes. Each snippet classifier $C_i(t_i)$ — with $i \in [1..N]$ — is an expert in predicting the sensitiveness of t_i-second-sized snippets.

As expected from most of the machine-learning techniques, the resulting fusing system may operate in one of two modes, namely training and test operation. It is depicted in Figure 6.1: depending on the type of system operation, the activity sequence may reach either the *training end*, or the *test end*. As it follows, in Section 6.1.1, we detail the training activity sequence (*Snippet Classification*, *Snippet Alignment*, *Fusion Vector Extraction*, and *Fusion Meta-Learning*), in which the desired system content-localization behavior is learned from the labeled *Fusion Training Dataset*. In Section 6.1.2, in turn, we explain the test activity sequence (*Snippet Classification*, *Snippet Alignment*, *Fusion Vector Extraction*, *Fusion Vector Class Prediction*, *Classification Score Smoothing*, and *Classification Score Combination*), in which an arbitrary unlabeled *Test Video* is presented to and analyzed by the system.

6.1.1 Training Activity Sequence

Figure 6.2 depicts the training activity sequence of the proposed fusion solution, by means of an illustrative toy case, with $N = 2$ snippet classifiers, and a *Fusion Training Dataset* that contains three videos (*Videos A, B, and C*, in the related diagram). Notwithstanding, in spite of the quantity of snippet classifiers, and of training videos, the aimed operation is always divided into four steps.

Snippet Classification

Step 1 refers to the *Snippet Classification* activity, in which the *Fusion Training Dataset* — represented by a hollow cylinder — is submitted to the snippet classifiers. The training dataset must be annotated at frame level, with the indications of the starting and ending times of the sensitive and non-sensitive sequences. The snippet classifiers, in turn, are represented by *black boxes*, in the sense that it is not important how they operate, considering the execution of the proposed fusion method. They can be implemented with BoVW-based approaches, in accordance with the scope of this research, but they are not necessarily limited to that. In fact, what they really need to do is to return a set of classified snippets, which are grouped per classifier (and thus per length t_i), and per training video. Please notice that, in the chosen notation, we represent each snippet by a hollow rectangle, containing the resulting classification score in the center, and a small chronometer on the lower right corner, to highlight their temporal nature. The widths of these rectangles are supposed to indicate their duration, which means that — for the sake of illustration — *Snippet Classifier 1* ($C_1(t_1)$) is able to classify snippets that are longer than the snippets analyzed by *Snippet Classifier 2* ($C_2(t_2)$). Actually, in the given example, $t_2 = 3\ t_1/5$.

Snippet Alignment

Step 2 refers to the *Snippet Alignment* activity, which is performed per training video: at such point, snippets coming from different streams cannot be mixed together yet. As one might observe in Figure 6.2, given that the snippets are defined by a starting and an ending time, they are all aligned along the video timeline, in order to reveal their coincidences.

In practice, the *Snippet Alignment* activity is performed as follows. For each classifier, the respective snippets are sorted according to their starting times. That leads to one sorted list of snippets per classifier. These lists are thus stored, for further use by a query function $q(t)$, which retrieves all the snippets, within all the lists, that coincide at a given instant of interest t. This is done through a binary search over each sorted list, which compares the instant of interest, and the bounds (starting and ending times) of the snippets.

Fusion Vector Extraction

Step 3, in turn, refers to the *Fusion Vector Extraction* activity. At this point, we want to generate a finite number of fusion vectors, which bundle the classification scores that were

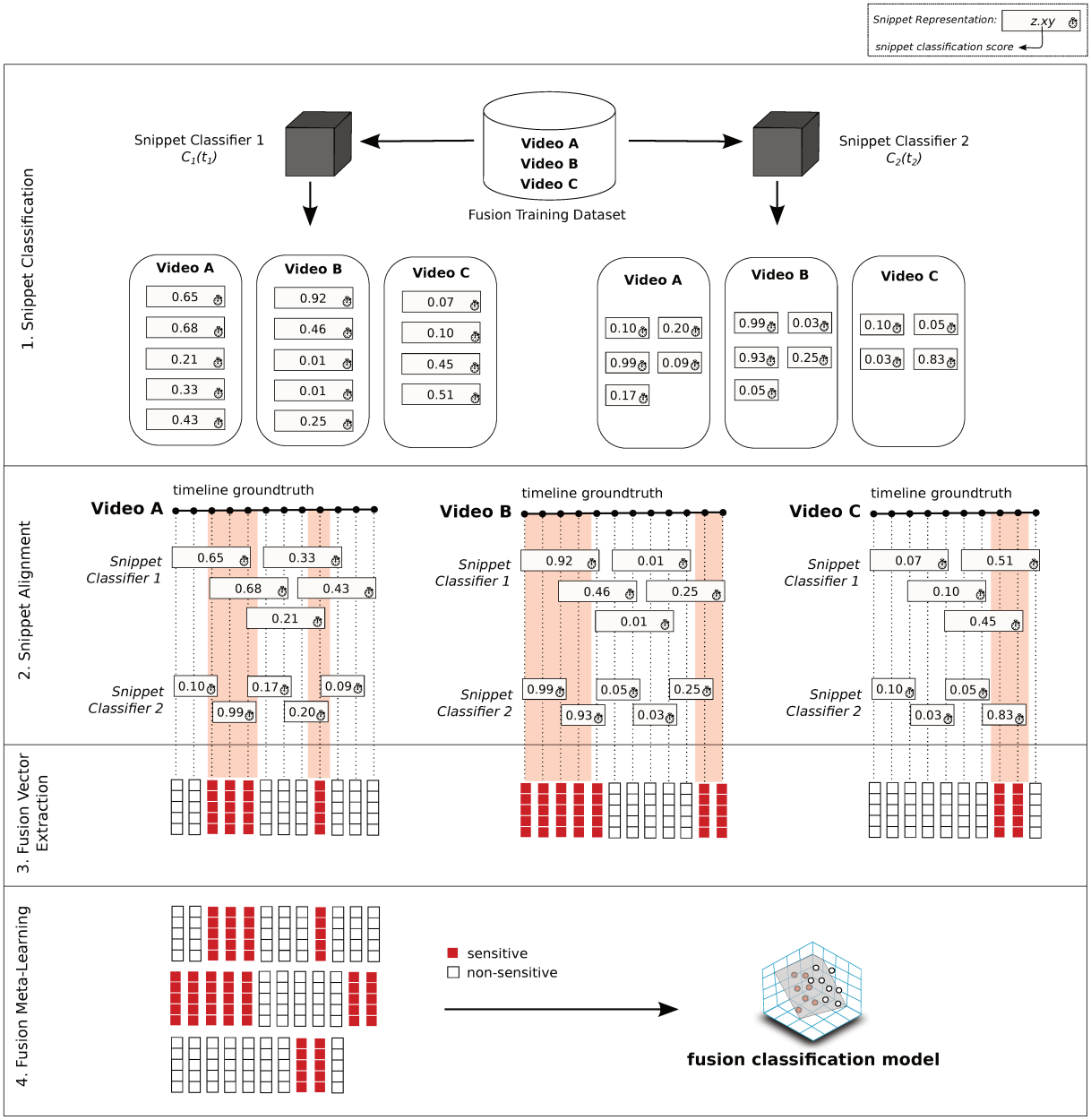


Figure 6.2: Toy-case instantiation of the proposed fusion training pipeline. The method starts with the *Fusion Training Dataset* (*Videos A, B, and C*), which is submitted to the different snippet classifiers that need to be fused: classifiers $C_1(t_1)$ and $C_2(t_2)$. The training dataset sensitiveness must be annotated at frame level. The method ends with a meta-learned classification model (*fusion classification model*), which must be stored for further use, during the test system operation. The size of the training dataset, and the quantity of combined snippet classifiers, can be larger than the given example, with no changes on the order of the depicted steps.

previously returned by the various snippet classifiers. For that, we sample the snippet alignments at every d seconds of video. Each second leads to an instant of interest t, which is fed to the aforementioned query function $q(t)$, and retrieves all the snippets that coincide at t.

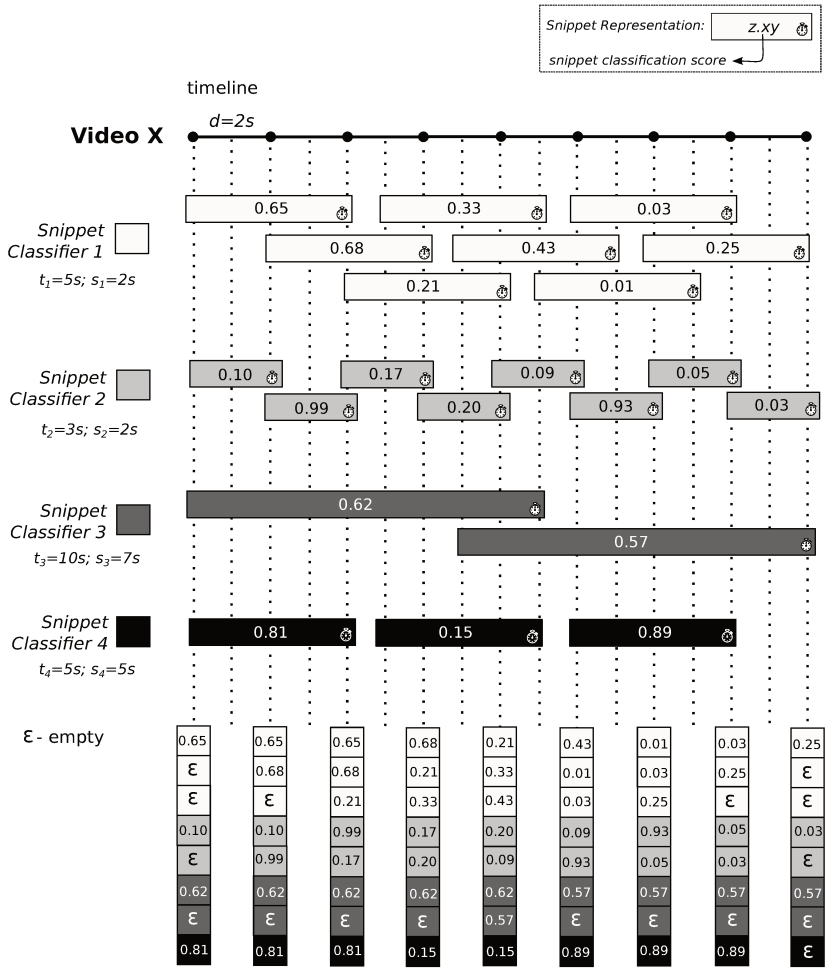


Figure 6.3: Extracting the combined confidence vectors for later fusion. In this example, four snippet classifiers are being combined, regarding the content of *Video X*. Fusion vectors are extracted every d seconds of video, and are filled with snippet classification scores. Missing values are indicated by ϵ. The different vector component colors indicate the source snippet classifier.

Figure 6.3 depicts the combination of fusion vectors, within the *Fusion Vector Extraction* activity, for a particular case of combining four snippet classifiers. As one might observe, for each video instant of interest (that is obtained in accordance to d), a fusion vector is extracted, containing the classification scores of coincident snippets. The coincident snippets must be sorted by source classifier and starting time, according to a predefined order of snippet classifiers. As a matter of fact, such order can be any, as long as it is repeated in the test system operation. In Figure 6.3, the colors of the fusion vector components indicate the snippet classifiers they are linked to, and therefore they reveal the fusion order.

The length l of every fusion vector is given by:

$$l = \sum_{i=1}^{N} \left\lceil \frac{t_i}{s_i} \right\rceil , \tag{6.1}$$

where N is the number of fused snippet classifiers, t_i is the length — in seconds — of the snippets for which classifier C_i is expert in predicting, and s_i is the step — in seconds — used to start a new snippet that is supposed to be analyzed by classifier C_i. Equation 6.2 calculates the size of the fusion vectors that are depicted in Figure 6.3, where $N = 4$:

$$\begin{aligned} l_{sample} &= \left\lceil \frac{t_1}{s_1} \right\rceil + \left\lceil \frac{t_2}{s_2} \right\rceil + \left\lceil \frac{t_3}{s_3} \right\rceil + \left\lceil \frac{t_3}{s_4} \right\rceil \therefore \\ &= \left\lceil \frac{5}{2} \right\rceil + \left\lceil \frac{3}{2} \right\rceil + \left\lceil \frac{10}{7} \right\rceil + \left\lceil \frac{5}{5} \right\rceil = 8. \end{aligned} \tag{6.2}$$

On the occasion of creating the fusion vectors, in the case of missing snippets (and thus missing classification scores), the respective vector components may be assumed as a value of complete uncertainty (e.g., 0.5, in the case of a normalized confidence score, which varies from zero — i.e., no confidence at all — to one — i.e., total confidence), or they can be interpolated. Missing vector components are represented by ϵ, in Figure 6.3.

Fusion Meta-Learning

Back to Figure 6.2, each discrete fusion vector that is obtained in *Step 3* is linked to an instant of interest, within the target video timeline. As one might observe, the labels of such vectors are deductible from the training dataset groundtruth, being either depicted in red, if the vector concerns a sensitive instant, or in white, if the vector lies within a non-sensitive segment. In the sequence, the *Fusion Meta-Learning* activity (*Step 4*) refers to the application of a machine-learning technique for generating a mathematical model that is able to predict the labels of unknown fusion vectors. Given that these fusion vectors are generated from previously machine-learned classification scores, we may say that we are performing a meta-leaning of the joint behavior of such scores.

In this work, we explore three implementation alternatives for the *Fusion Meta-Learning* activity: (i) score thresholding, as a baseline, (ii) Naïve Bayes Classifier [78], as a representative of generative strategies, and (iii) SVM [91], as a representative of discriminative strategies. In addition, all of them are conceived to return a confidence score, in the real interval [0..1], when classifying each fusion vector, which we refer to as *fusion score*. In the following, we give details of each one of these three fusion meta-learning methods.

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

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

$$fusion\_score(v) = \frac{\sum_{i=1}^{l} v[i]}{l}, \tag{6.3}$$

where l is given by Equation 6.1.

With such fusion score, we define the label of v as being:

$$label(v) = \begin{cases} positive, & \text{if } fusion\_score(v) \geq \tau; \\ negative, & \text{otherwise,} \end{cases} \tag{6.4}$$

where τ is the decision threshold.

Naïve Bayes Classifier As explained in [75], generative strategies for data learning usually establish a model of the joint probability of observations and labels, which are generalized by means of the Bayes theorem, for predicting the most likely label of an arbitrary unknown observation. In this work, we experiment with a simplified discrete naïve Bayes strategy [78].

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

$$b(v, i) = \begin{cases} 1, & \text{if } v[i] \geq \tau; \\ 0, & \text{otherwise.} \end{cases} \tag{6.5}$$

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

The mentioned prior and conditional probabilities (i, ii, and iii) constitute the *fusion classification model* (please refer to Figures 6.2 and 6.4). For predicting the probability of an arbitrary l-sized binarized vector b_j being positive, we apply the Bayes theorem:

$$p(positive|b_j) = \frac{p(positive) \times p(b_j|positive)}{p(b_j)}, \tag{6.6}$$

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

Complementarily, we calculate the probability of b_j being negative as 1.0 - $p(positive|b_j)$. For determining its label (positive or negative), we pick the one that is more probable (i.e., positive, if $p(positive|b_j) \geq 0.5$, or negative, otherwise). Additionally, we return $p(positive|b_j)$ as the resulting fusion score.

Support Vector Machine In contrast to the generative strategies, discriminative strategies focus on directly modeling the posterior probability of an observation belong to a target class [75]. Typical representatives include the solutions that aim at establishing the boundaries that better separate elements from different problem classes. The posterior probability, thus, can be estimated as a function of the element distance to the boundary. The farther away an element is from the boundary within the side of class x, the stronger the evidence of belonging to x.

SVMs [91] are popular representatives of such discriminative strategies. Roughly speaking, SVMs comprise supervised-learning methods that compute the optimal hyperplane that better separates a feature space into two classes. In addition, it is possible to transform the original feature space into another, in which the computed separation hyperplane is more effective for class separation. This is done implicitly, by means of a kernel function, which algebraically operates over the elements of the original feature space, to find their representatives into the new better-separable higher-order feature space.

In this work, we apply an SVM with a Radial Basis Function (RBF) kernel, for learning how to separate the fusion vectors into positive and negative samples. As pointed out in [48], RBF is a reasonable choice for SVM kernel, because it nonlinearly maps samples onto a higher dimensional space, so that, in the case of class elements being nonlinearly separable, the nonlinearity is handled. For reporting the fusion score (i.e., the SVM prediction confidence), we employ the standard Platt normalization [72], which calibrates the element distances to the decision hyperplane, conveniently returning a value in the real interval [0..1].

6.1.2 Test Activity Sequence

Figure 6.2 depicts the test activity sequence of the proposed fusion solution, by means of an illustrative toy case, with $N = 2$ snippet classifiers. Notwithstanding, in spite of the quantity of snippet classifiers, the aimed operation always starts with an arbitrary video (*Test Video D*), and is always divided into six steps. The initial three activities (represented by *Snippet Classification*, *Snippet Alignment*, and *Fusion Vector Extraction*) are the same from the training sequence. The only difference relies on the absence of timeline groundtruths — in the test case — with no impact on the refereed activities. The three remaining activities (*Fusion Vector Class Prediction*, *Classification Score Smoothing*, and *Classification Score Combination*), which are test-exclusive, are detailed as it follows.

Fusion Vector Class Prediction

Prior to this step, the target video (properly represented by *Test Video D*, in Figure 6.4) is supposed to have been segmented into snippets, which must have been classified during the *Snippet Classification* activity. In addition, the classified snippets must have been aligned

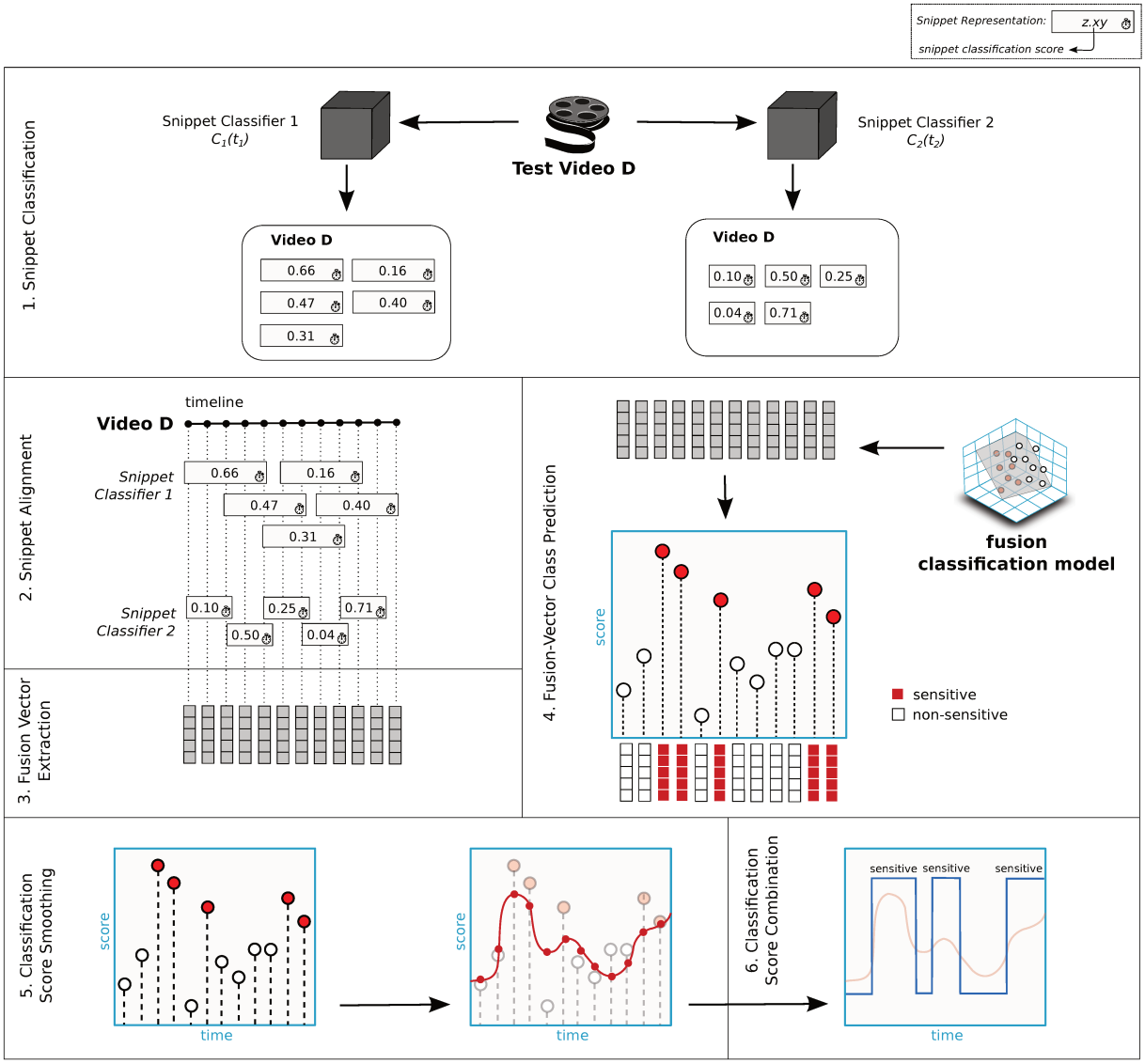


Figure 6.4: Toy-case instantiation of the proposed fusion test pipeline. The method starts with the unlabeled sample *Test Video D*, which is submitted to the different snippet classifiers that are fused: classifiers $C_1(t_1)$ and $C_2(t_2)$. The method ends returning the instants when *Test Video D* starts and ceases to display sensitive content, optionally enriched by confidence scores. The quantity of combined snippet classifiers can be larger than the given example, with no changes on the order of the depicted steps.

along the video timeline (during the *Snippet Alignment*), and thereafter combined into fusion vectors (during the *Fusion Vector Extraction* activity). In the particular case of the *Fusion Vector Extraction* activity, it is important to mention that the order in which the snippet classification scores are combined — for generating the fusion vectors — must be the same that was adopted in the training system operation (please refer to Figure 6.3, for more details).

As one might observe, in the beginning of *Step 4*, in Figure 6.4, the labels of the fusion vectors are unknown (what is represented by their gray colors). Hence, the system retrieves the *fusion classification model*, and predicts the labels of each fusion vector, with a proper confidence score. That justifies their red and white colors, in the end of

Step 4. As a result of that, considering that each fusion vector represents a discrete instant of interest within the target video timeline, the predicted labels actually classify the sensitiveness of every video instant of interest.

Strategies to perform the class prediction are a consequence of the chosen *Fusion Meta-Learning* solution, which — as already mentioned — may comprise SVM [91], Naïve Bayes Classifiers [78], etc.

Classification Score Smoothing

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

In this work, we propose the use of a unidimensional Gaussian blurring function, with standard deviation σ, which is convolved with the time-sorted sequence of classification scores. That leads to a more well-behaved sequence of scores, besides offering the opportunity of eliminating eventually incorrect predictions, according to the time-surrounding evidences. Dai et al. [29] report to adopt a similar solution, which relies upon an score-averaging convolution filter, instead of a Gaussian one.

Classification Score Combination

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

In the end, we come up with a continuous answer, which discriminates the instants the target video starts and ceases to disclose sensitive content.

6.2 Snippet Classification

In this work, we suggest to operate on the mid-level layer, in order to adequate the BoVW pipeline to the sensitive-content localization problem. As mentioned in Section 2.1.2, such layer is related to the combination of the low-level local features into broader representations, with intermediary complexity. In such process, we know that it is desirable to combine the features in such a way that they incorporate some semantic information from the classes of the problem. In addition, as it is reasonable to expect, features referring to

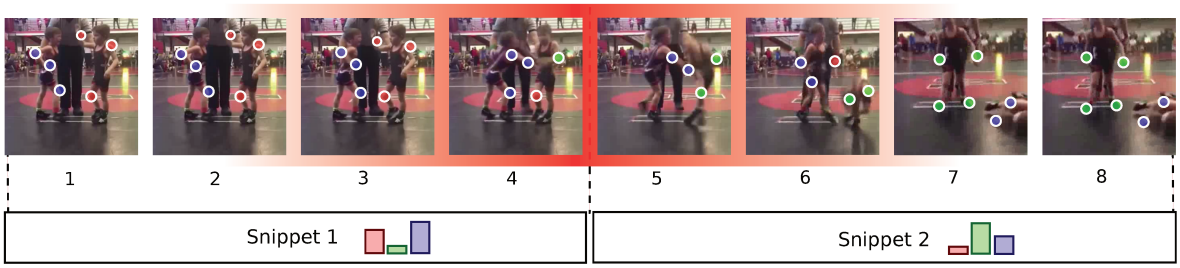


Figure 6.5: Non-overlapping bags of features. Each bag is obtained from a non-overlapping video snippet (either *Snippet 1*, or *Snippet 2*). Images 1-8 depict the frames of a video sequence of interest. Frames 3-6 depict a violent event. *Snippet 1* comprises frames 1-4, and the respective local features, which are represented by small colorful circles. *Snippet 2* comprises frames 5-8, and the respective features. Due to the non-overlapping nature of the snippets, the violent content is improperly split in the middle, and spread between the two bags.

the same sensitive captured moment shall be adjacent in time. Thus, a useful semantic information can be the temporal proximity of the features, along the video timeline.

With that in mind, we propose to pool and to normalize consecutive features, as long as they belong to a same fixed-length video segment (a.k.a., a snippet). Moreover, as we are looking for designing a more general-purpose solution, we do not assume anything about the target video stream, regarding number of camera sources, presence of scene cuts, amateurishness, or studio film grammar. Instead, we recommend to establish snippets that systematically overlap in time, as an effort to let sensitive events be entirely enclosed by at least one bag, in spite of eventually being split among the others.

6.2.1 Overlapping Snippets

Figure 6.5 depicts the situation in which the pooled snippets do not overlap, such as the usual strategies proposed in [99, 55, 29]. Images 1-8 represent the frames of a sample video sequence, whose frames 3-6 capture a sensitive event (actually, a violent event, regarding one kid punching another, properly highlighted in red). Over the frames, the colored circles illustrate eventually extracted local features, and the two non-overlapping rectangles that are positioned below represent the snippets that one might use, for establishing bags of such features. As one might observe, due to the non-overlapping nature of the bags, the violent motion is split in the middle and, therefore, it is entirely represented by none of the bags.

In contrast to Figure 6.5, Figure 6.6 illustrates the benefit of our suggestion of pooling overlapping snippets, with redundant content. The new overlapping *Snippet 1.5* is a video segment whose bag shares local features with the bags of *Snippets 1* and *2*. As a consequence, in spite of *Snippets 1* and *2* splitting the violent content, *Snippet 1.5* allows us to establish a bag that better represents the sensitive motion, without sudden mid fragmentation.

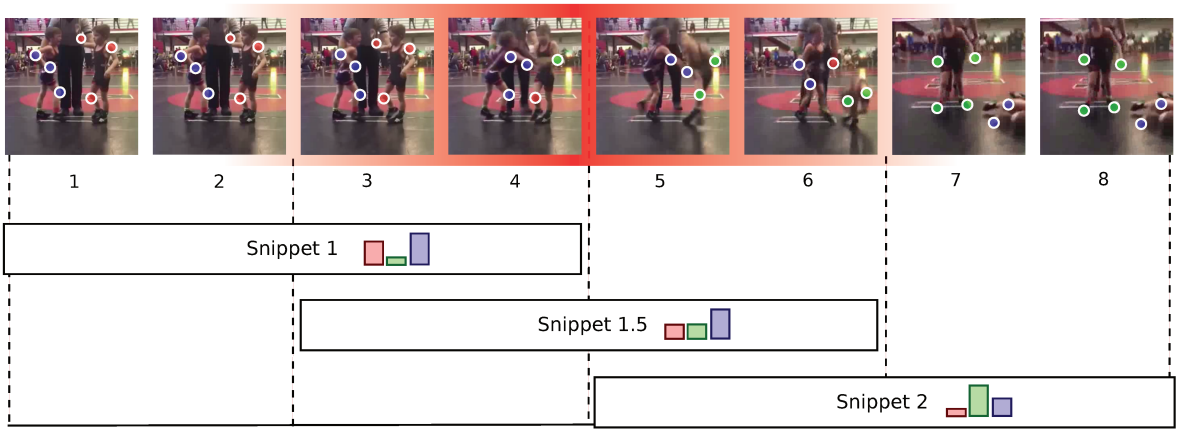


Figure 6.6: Overlapping bags of features. Each bag is obtained from a particular video snippet (either *Snippet 1*, or *Snippet 2*, or the overlapping *Snippet 1.5*). Images 1-8 depict the frames of a video sequence of interest, with frames 3-6 containing a violent event. *Snippet 1* comprises frames 1-4, *Snippet 2* comprises frames 5-8, and *Snippet 1.5* comprises frames 3-6. Contrary to Figure 6.5, in spite of *Snippets 1* and *2* splitting the violent content, the overlapping *Snippet 1.5* might result in a bag that captures most of the sensitive event.

Pooling and normalizing the mid-level features according to overlapping fixed-length snippets mean that we aim at training *snippet classifiers*. Prior to the training process, one must choose a length t, measured in seconds, for the duration of the snippets, and a sliding step s, also measured in seconds, to systematically start a new snippet along the video timeline. For the sake of illustration, let us admit $t = 10s$, and $s = 1s$. It means that, given a target video stream, we can group and analyze visual features inside windows of $10s$, that start at every second of movie.

The values of t and s may depend on the nature of the low-level video descriptions (e.g., space-temporal, static, etc.), and on the characteristics of the target sensitive content (e.g., violence, pornography, etc.). Nevertheless, in Section 6.1, we introduced a high-level fusion method of snippet classification outputs, which allows the combination of diverse classifiers that might have been trained with different sizes of t and s, despite of relying on the same low-level features. In addition, such fusion method also allows us to combine different low-level features and modalities (e.g., visual and auditory features) for the different snippets that are being considered.

6.2.2 Snippet Labeling

If the training dataset is annotated with a granularity of seconds, an extracted snippet may partially coincide with sensitive, and partially coincide with adjacent non-sensitive video segments. In such cases, what label can we assume, specially in the training process?

As a solution for that, we propose two percentile variables n and p. Regarding the n one, we can, for instance, assume it as $n = 100.0\%$, indicating that the extracted snippets are considered non-sensitive if they fall entirely out of all the given sensitive segments. On the other hand, we can assume $p = 75.0\%$, indicating that snippets coinciding in 75.0% with any of the sensitive sequences are meant to be labeled as positive.

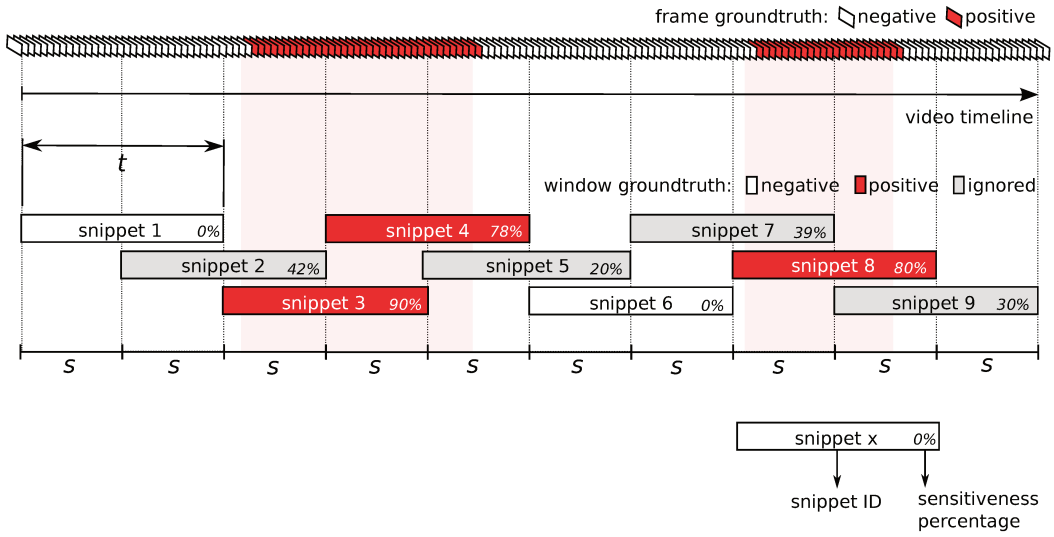


Figure 6.7: Extraction and labeling of snippets within a video stream. This configuration is supposed to be used in the training stage of a solution to the sensitive-content localization problem. For the sake of illustration, consider $s = t/2$, $p = 75.0\%$, and $n = 100.0\%$. White snippets are non-sensitive (negative), while red snippets are sensitive (positive). Gray snippets are not supposed to be used in the training process.

Table 6.1: Proposed variables to the scheme of snippet pooling. Variables t and s are measured in seconds. Variables n and p correspond to percentile values.

Variable	Meaning
t	Size of the snippets that group adjacent features, in the movie timeline.
s	Sliding step to start a new snippet, in the movie timeline.
n	Percentage of snippet falling out of sensitive segments, in order to be labeled as negative.
p	Percentage of snippet falling inside sensitive segments, in order to be labeled as positive.

Figure 6.7 depicts how the mentioned variables t, s, n, and p would impact on the extraction and on the labeling of snippets, in the case of $s = t/2$, $p = 75.0\%$, and $n = 100.0\%$. As one might observe, by adopting such strategy, we may not only obtain non-sensitive and sensitive snippets (respectively represented by white and red boxes), but may also gather dispensable ones (in gray), which might not be used in the training process.

Table 6.1 summarizes the four parameters. In the test system execution, when one is predicting the classes of the snippets, the variables n and p have no effect. Regarding variable t, we recommend the use of the same value that was employed in the training stage as we expect the previously trained classifier to be an expert in classifying snippets with size t seconds. Concerning variable s, we suggest the use of a small size, which may not necessarily be the same that was used in the training system operation. The idea here is to have as many snippet classifications as possible, along the target video timeline. We leave to the further high-level multimodal fusion process the task of combining the overlapping classification scores into a single continuous answer, as it is explained in

Section 6.1.

6.3 Final Remarks

The problem of sensitive-content localization is arguably more complex than the previously tackled problem of sensitive-video classification. It suffers from the same big-data-, pervasiveness-, subjectivity-, and urgency-related challenges, but with the worsening of having to find all the moments a stream starts and ceases to display sensitive content, instead of answering a single yes-no question. To be fair, we can admit that the localization problem is at least as difficult as the classification one.

In face of such complexity, in the case of the sensitive-content localization problem, we focused on designing an effective solution, and we ultimately aimed at dealing with the detected-content subjectivity. For that, we bet on relying upon the complementarity of distinct snippet classifiers, that shall individually use different data modalities (e.g., video frames, audio stream, video space-time, etc.). Moreover, we recommend a dense analysis of the target video streams, by means of time-overlapping snippets, and we propose a late fusion of snippet-classification scores, whose separation in positive and negative samples shall be learned with machine-learning techniques (i.e., we suggest a meta-learning strategy).

Therefore, as a result, we introduce a novel high-level multimodal fusion pipeline for sensitive-video localization. Such pipeline is of general purpose, in the sense that it can be easily adapted for the localization of diverse sensitive content (e.g., violence, pornography, gore scenes, child abuse, etc.); all one needs to do is to provide a new and soundly frame-level annotated dataset, with positive and negative examples for a comprehensive training stage. Furthermore, the pipeline allows us to easily combine the most suitable methods for analyzing each single modality, such as — for instance — hidden Markov models for audio, and SVMs for images, seamlessly. That would be much more difficult, if we had chosen an early-fusion approach.

In the next two chapters, we validate the fusion pipeline, for both pornography localization (in Chapter 7), and violence localization (in Chapter 8). We combine visual and auditory features, which are obtained with diverse low-level descriptors.

Chapter 7

Pornography Localization: Experiments

In this chapter, we validate the sensitive-content localization pipeline that was introduced in Chapter 6, for the particular case of pornography detection. We thus evaluate the combination of diverse snippet classifiers, which individually rely upon the description of different data modalities (from auditory to visual features).

For that, in Section 7.1, we explain the adopted experimental setup, in terms of dataset, experimental protocol, metrics, selected snippet classifiers, parametrization, and implementation details. Afterwards, in Section 7.2, we report the experimental results, while in Section 7.3, we present some final remarks.

7.1 Experimental Setup

As explained in Section 2.2, to the best of our knowledge, there is no video dataset in the literature that provides frame-level annotation for supporting the task of pornographic content localization. To cope with this, we annotated every frame of the 140-hour Pornography-2k dataset (previously introduced in Section 4.1.1), a process that we detail in Section 7.1.1. In Section 7.1.2, we explain the experimental protocol and the metrics we use to evaluate the results, while in Section 7.1.3, we present the multimodal snippet classifiers we have designed and selected for combination. Next, in Section 7.1.4, we define the parameters of the explored high-level fusion solutions, focusing on the *Fusion Meta-Learning* task alternatives (please refer to Section 6.1, for details about such task).

7.1.1 Pornography-2k Dataset Annotation

To facilitate the assessment of the strategies aiming at localizing pornography, we decided to take a step further with the Pornography-2K dataset: to annotate it at frame level.

In order to support the task of annotating the videos of the Pornography-2K dataset, we developed a tool to extract every frame of a given video, and thus show the images in a time-sorted and keyboard-controlled way. Therefore, by inspecting the frames one-by-one, and pressing the correct keys, one can easily annotate parts of the stream as positive or negative. Additionally, the task can be safely accelerated by increasing or decreasing the speed of showing images onto the screen, and the frames can be played in normal or reverse order.

(a) (b)

Figure 7.1: Interface of the frame annotation tool. In (a), frame 123 is being tagged as negative. In (b), frame 689 is being tagged as positive.

Figure 7.1 depicts the interface of the annotation tool, by showing two labeling situations. The program is fully keyboard-based, for a matter of agility. Annotation status is shown at the top, and available commands are displayed at the bottom. The current frame goes in the middle. Figure 7.1(a) shows a situation of labeling frame 123 as negative. Figure 7.1(b), in turn, shows the labeling of frame 689 as positive. The software is written in C++, and it relies upon the OpenCV C++ API [17].

As one might observe, we had 2,000 videos to annotate. With respect to the 1,000 negative videos, the annotation process was simple: they were automatically and entirely marked as negative frame sequences.

In turn, concerning the 1,000 positive videos, we had to inspect every frame of the dataset. For that, we recruited four RECOD[1] members to concurrently annotate the videos, with the help of the video annotation tool. Each one was responsible for 250 videos, that were randomly distributed. In order to equalize the situations one should consider positive, all annotators adopted the concept of pornography as of being "any explicit sexual matter with the purpose of eliciting arousal" [81]. Moreover, five videos were chosen at random and, prior to the major official annotation process, all four members dedicated some time to annotate these samples for further discussion, aiming at calibrating the opinions.

Table 7.1 brings the statistics of the annotated videos. As one might observe, the Pornography-2K dataset has a total of almost 140 video hours. From this total, 91h43min (65.54%) refer to pornographic content.

[1]RECOD is the Reasoning for Complex Data laboratory, which is hosted at the Institute of Computing (IC), of the University of Campinas (Unicamp). Cf. `http://www.recodbr.wordpress.com`, accessed May 3rd, 2016.

Table 7.1: Time statistics on the annotated pornographic videos. As one might expect, negative videos do not have positive sequences, only negative. Positive videos, in turn, might have non-pornographic frame intervals.

	Negative Sequences	Positive Sequences	Total
Negative Videos	40h25min	00h00min	40h25min
Positive Videos	07h49min	91h43min	99h32min
Total	48h14min	91h43min	139h57min

7.1.2 Experimental Protocol and Metrics

In face of the 140-hour Pornography-2K dataset, and due to the very time- and resource-consuming experiments, we apply, for pornography localization, the same variation of the 5×2-fold cross-validation protocol [38] that was used in Section 4.1.2: the 3×2-fold protocol. It consists of randomly splitting the dataset into two same-size class-balanced folds, three times, and in each time, training and test sets are switched, leading to six independent experiments, for each evaluated solution. In addition, in order to enable paired tests, we submit the exact six folds to each pornography locator. Therefore, whenever it is convenient to compare different locators with some statistical confidence, we employ the non-parametric pairwise Wilcoxon signed-rank test, with Bonferroni's correction [35]. Lastly, given the nature of our pipeline — in which we have two moments of data learning, (i) one related to the snippet classification learning, and (ii) the other related to the fusion meta-learning — we divide the training datasets into two disjoint parts: 60% for snippet classification learning, and 40% for fusion meta-learning.

Similar to the case of pornographic video classification, for assessing the performance of the pornography locators, we report the normalized accuracy (ACC), and the F_2 measure (F_2). However, the respective values are collected in a different granularity. In the former case of pornography classification, TPR, TNR, sensitiveness, and precision (which are basic for calculating ACC and F_2[2].) are collected *per video file*. It thus makes sense to think about the classification system performance, for instance, as failing to classify one in every five videos of interest. In the case of pornography localization, on the contrary, TPR, TNR, sensitiveness, and precision are collected *per second of video*. With that, we can report the performance of the localization system, for instance, as failing to localize one in every five seconds of pornographic video. As a consequence, on the occasion of localizing pornographic scenes, we verify whether or not the system is right at every second of video, therefore reporting the averages of ACC and of F_2. The reported values are not averaged over the number of video files anymore. Instead, they are averaged over the total amount of seconds that comprise all the analyzed videos together.

7.1.3 Multimodal Snippet Classifiers

The proposed high-level fusion pipeline, introduced in Chapter 6, is evaluated through different combinations of four distinct snippet classifiers. Two of these classifiers rely

[2]Please refer to Section 4.1.2, for the mathematical definitions

Table 7.2: Parameter values used in the training of the snippet classifiers. Variables t, $s_{positive}$, and $s_{negative}$ are measured in seconds. Variables n and p correspond to percentile values.

Variable	Value	Meaning
t	5.0	Size of the snippets.
s_p	6.0	Sliding step to start a new positive snippet.
s_n	2.5	Sliding step to start a new negative snippet.
n	100.0	Percentage of negative snippet falling out of positive scenes.
p	80.0	Percentage of positive snippet coinciding with positive scenes.

upon auditory features, namely MFCC [32] and prosodic features (fundamental frequency, voicing probability, and loudness contours). The remaining two rely upon visual features, of which one is representative of still image descriptors (namely, HOG [30]), and the other is representative of space-temporal descriptors (namely TRoF, which was introduced in Section 3.2).

All classifiers are trained to label snippets that are five-second long (i.e., $t = 5s$). In preliminary experiments, such duration showed a good compromise between the quantity of described snippets, vs. the information amount that constitutes each snippet. For instance, on the occasion of learning from a 10-second long positive scene, only one snippet of 10 seconds can fit such video excerpt. In opposition, if the snippets are five-second long, many of them can fit the scene, since we allow temporal overlap. Hence, shorter snippets result in more data to be learned. In the same direction, if we employ two-second long snippets, we can establish even more snippets within those 10 seconds. Nevertheless, a two-second snippet has less than half of the information that constitutes a five-second sample. Early investigations revealed that such shorter sequence is not enough for capturing discriminative pornographic phenomena.

In the training phase of all classifiers, we consider a snippet negative if it falls entirely out of positive scenes (i.e., $n = 100\%$). Similarly, we consider a snippet positive if it is at least 80% coincident with positive scenes (i.e., $p = 80\%$). For obtaining a balanced training set from the Pornography-2k dataset (i.e., a set with nearly the same amount of positive and negative snippet samples), we extract one positive snippet in every six seconds of a pornographic scene ($s_p = 6s$), and one negative snippet in every 2.5 seconds of a non-pornographic scene ($s_n = 2.5s$). In Table 7.2, we summarize such parameters.

In the test phase, we describe one five-second long snippet in every second of a video sequence (i.e., $t = 5s$, and $s = 1s$). That allows us to constitute one fusion vector per second, over the test dataset.

Regardless of the used low-level features, we employ Fisher Vectors [71] — one of the best mid-level representations [21] — for aggregating the low-level descriptions, within all the snippet classifiers. The codebooks are GMM-based, and each GMM is estimated over one million randomly sampled low-level descriptions (with 500,000 coming from the training positive scenes, and 500,000 coming from the training negative scenes). Moreover, each GMM is composed of 256 Gaussians, as suggested in [71]. The Fisher Vector

encoding, and GMM estimation are performed with the support of the VLFeat API [92].

In the high level, we apply linear SVM classifiers — as suggested in [71] — by means of the LIBLINEAR library [41]. We apply grid search to find the best c-SVM parameter, during the snippet classification training. Concerning the test phase, we use the option offered by LIBLINEAR [41], for returning the confidence scores of each class prediction. The confidence scores are normalized in the real interval [0..1]: the closer to one, the higher the certainty about the classification.

In the following, we detail the low-level description of each one of the four types of used snippet classifiers.

Mel-Frequency Cepstral Coefficients (MFCC)

MFCC features are used primarily for speech description [40], and a great deal of works in the literature have been using it for violent video content detection [1, 36, 99, 29, 63, 55]. In this work, we use it for pornographic content localization, through the OpenSmile library [40].

For extracting the MFCC features, we use the default parameters of OpenSmile. We therefore obtain 39-dimensional low-level auditory features in every 40 milliseconds of audio, without overlap. As a consequence, we associate each MFCC description to one video frame, allowing us to describe each stream at a rate of 25 frames per second.

Finally, prior to the following Fisher Vector encoding, we apply PCA to reduce the MFCC descriptions to 24 dimensions (as recommended in [54]). For the calculation of the PCA transformation (reference eigenvalues and eigenvectors), we randomly sample one million MFCC descriptions from the training set, with half of them coming from the negative set, and the other half coming from the positive set. For that, we use the OpenCV C++ API [17].

Prosodic Features (PROS)

In addition to MFCC, we extract prosodic features (PROS) as a second alternative for describing audio. Similar to MFCC, we employ the OpenSmile library [40] for obtaining three-dimensional features (fundamental frequency, voicing probability, and loudness contours) in every 40 milliseconds of audio, without overlap.

Prior to the following Fisher Vector encoding, we apply PCA to whiten the low-level descriptions (i.e., we maintain their three dimensions), instead of reducing their size, which is already small. Again, for the calculation of the PCA transformation, we randomly sample one million prosodic descriptions from the training set, with half of them coming from the negative set, and the other half coming from the positive set. For that, we apply the OpenCV C++ API [17].

Histograms of Oriented Gradients (HOG)

To provide a visual descriptor that relies solely on static images, we employ HOG [30] as the basis of one of the available snippet classifiers. Thereby, for the sake of processing

time, we extract two frames per second from the video footage. Next, we discard 10% of the image borders, to remove possible watermarks.

HOG descriptions 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. Each patch is described by a single HOG block, which is divided into 4×4 HOG cells. Each cell is described by eight bins, what leads to $4 \times 4 \times 8$ description values per patch. Hence, the obtained HOG feature vectors are 128-dimensional. For that, we apply the OpenCV HOG implementation [17].

Prior to the following Fisher Vector encoding, we apply PCA to reduce the size of the HOG descriptions by half, to 64 dimensions. Similar to MFCC and PROS, we randomly sample one million HOG descriptions from the training set, with half of them coming from the negative set, and the other half coming from the positive set, for supporting the calculation of the PCA transformation.

Temporal Robust Features (TRoF)

To provide a visual descriptor that captures video space-time properties, we employ TRoF as the basis of one of the available snippet classifiers. In addition, we employ HOG descriptors to represent the TRoF blob content (please refer to Section 3.2).

For detecting the three-dimensional blobs of interest, with the TRoF detector, we use the same configuration that is presented in Table 4.2, due to its good preliminary results, which are statistically equivalent to state-of-the-art video descriptors (please refer to Section 4.2.2, for details). We therefore calculate the integral video at every 250 frames of the target video (i.e., $c = 250$). For each obtained integral video, to describe video more quickly, we sample one in every four video voxels, in all directions (i.e., $s = 4$), and apply four space-temporal scale octaves (i.e., $o = 4$), to perform the Hessian calculations. Thereafter, we extract 3,000 blobs at every 250 integral video frames ($b = 3,000$).

Finally, prior to the following Fisher Vector encoding, we apply PCA to reduce the TRoF descriptions from 192 dimensions to half (i.e., 96 dimensions). As usual, for the calculation of the PCA transformation, we randomly sample one million TRoF descriptions from the training set, with half of them coming from the negative set, and the other half coming from the positive set.

7.1.4 Fusion Meta-Learning Solutions

As explained in Section 6.1, the fusion pipeline provides — for each video instant of interest — one fusion vector that encodes all the time-coincidental analysis of each snippet classifier. The *Fusion Meta-Learning* task, in turn, aims at learning mathematical models that are able to predict the label of arbitrary unknown fusion vectors, according to the labels of known vectors (from the training dataset). In this work, we explore three solutions for performing such task: (i) score thresholding, as a baseline, (ii) Naïve Bayes Classifier, as a representative of generative strategies, and (iii) SVM, as a representative of discriminative strategies. In addition, all of them are conceived to return a confidence score, in the real interval [0..1], when classifying each fusion vector, which we refer to as *fusion score*.

Regardless of the used fusion meta-learning method, in the test system operation, we always convolve a Gaussian window with standard deviation $\sigma = 5s$ (the size of each snippet) over the temporal sequence of obtained fusion scores, for smoothing. It is related to the *Classification Score Smoothing* task, which is presented in Section 6.1.2. In the end, the *Classification Score Combination* task is performed as pre-conceived: by assuming a fusion score threshold $t = 0.5$, we substitute all the time-adjacent scores equal to, or greater than $t = 0.5$, by their average value. Complementarily, all the time-adjacent scores smaller than $t = 0.5$ are replaced by their own average value.

As a consequence of each snippet classifier being an expert in classifying five-second snippets, and of a new snippet being evaluated in every second of movie, the resulting fusion vectors may present the following possible l sizes (in accordance to Equation 6.1): (i) ten, if two classifiers are combined, or (ii) 20, if four classifiers are combined[3]. Furthermore, in the case of eventually missing snippets — which are related to the ϵ value, in Figure 6.3 — the empty fusion vector components are filled with a linear interpolation of the present ones.

In the following, we give details of the parameters that are used for each one of the three mentioned fusion meta-learning methods. In all the cases, we always provide a class-balanced training dataset for learning the fusion of scores.

Score Thresholding (THR)

As aforementioned, each fusion vector has a size l, which is either ten, or 20. The score thresholding solution (THR) averages the l confidence scores that lie within each fusion vector component, and defines the ultimate label of the vector according to a threshold τ. For details, please refer to Equations 6.3 and 6.4.

Considering that we make the combined snippet classifiers return confidence scores that are normalized in the real interval [0..1] (please refer to Section 7.1.3), we employ $\tau = 0.5$. Therefore, if the average of fusion scores is greater than, or equal to $\tau = 0.5$, the fusion vector at hand is labeled as positive; otherwise, it is negative.

Naïve Bayes Classifier (NBC)

As a representative of generative strategies, we employ a discrete Naïve Bayes Classifier (NBC) for performing the *Fusion Meta-Learning* activity. As already mentioned, depending on the number of combined snippet classifiers, the resulting fusion vectors are ten- or 20-sized. In the case of the NBC solution, the value of l has a direct impact not only on the size of the fusion vectors, but also in the number of possible binarized fusion vectors, which is given by 2^l. We thus have either 1,024 or 1,048,576 possible binarized vectors. As explained in Section 6.1.2, the NBC strategy considers the frequencies of each possibility, within the test dataset, for deciding the label of an arbitrary vector, according to the Bayes theorem (Equation 6.6).

Again, as we are combining the confidence scores that are returned by the snippet classifiers, we use $\tau = 0.5$, when binarizing a target fusion vector. It is done as depicted in Equation 6.5.

[3]As detailed in Section 7.2, we explore only the combination of two, or four snippet classifiers.

Table 7.3: Results of pornography localization over the Pornography-2K dataset, without fusion of snippet classifiers. We report the average performance over the 3×2 cross-validation folds. In all experiments, the standard deviation is lower than 0.02.

		Solution	TPR (%)	TNR (%)	ACC (%)	F_2 (%)
No-fusion	audio	PROS	75.23	77.39	76.31	77.22
		MFCC	79.31	80.13	79.72	80.98
	image	HOG	89.06	85.44	87.25	89.65
		TRoF	89.58	83.36	86.47	89.89

TPR: true positive — TNR: true negative rate — ACC: accuracy — F_2: F_2 measure

Support Vector Machine (SVM)

As a representative of discriminative strategies, we apply an SVM with a Radial Basis Function (RBF) kernel, for learning how to separate the fusion vectors into positive and negative samples. Given that in the experiments, the value of l is either ten or 20, we opt for an RBF kernel, as recommended in [48].

For that, we employ the LIBSVM API [20] for training fusion vector classifiers, and for predicting the class of arbitrary fusion vectors. As defined in Section 6.1.2, the fusion scores are calibrated by the standard Platt normalization [72]. Moreover, for finding the parameters that lead to the best RBF SVM, we perform a grid-search with five-fold cross validation over the training dataset, as suggested in [48]. All these features are available in the LIBSVM API [20] implementation.

7.2 Results

In this section, we present the results of the proposed fusion pipeline, on the occasion of combining the snippet classifiers that were described in Section 7.1.3, in varied manners. First, in Section 7.2.1, we present the individual results of the snippet classifiers, in face of the problem of pornography localization, without combinations. Thereafter, in Section 7.2.2, we present the results of combining snippet classifiers that rely upon the same type of low-level feature (i.e., we provide the results of fusions of audio-based classifiers, and of fusions of image-based classifiers, separately). Next, in Section 7.2.3, we present the results of multimodal combinations. Finally, in Section 7.2.4, we present some graphs that depict the quality of the pornography localization, over two selected samples from the Pornography-2k dataset.

7.2.1 Single Solutions

In Table 7.3, we present the individual results of the snippet classifiers, without combinations, in face of the problem of pornographic content localization. We report the normalized accuracy rate (ACC) and the F_2 measure (F_2), both averaged over the 3×2 cross-validation folds. Furthermore, we report the average true positive (TPR) and true negative (TNR) rates, to give the reader a broader view of the localization results.

Table 7.4: Pairwise comparison of the pornographic snippet classifiers. We report the statistical tests for all 3×2 folds, considering ACC and using the paired Wilcoxon test with Bonferroni's correction.

	PROS	MFCC	HOG
MFCC	0.030 ✔	–	–
HOG	0.030 ✔	0.013 ✔	–
TRoF	0.030 ✔	0.013 ✔	0.558 •

p-values • not statistically different ✔ row solution is better

As one might observe, visual features are more suitable for the task, with static and space-temporal approaches showing close performance. Indeed, as it is presented in Table 7.4, TRoF and HOG snippet classifiers are not statistically different with respect to ACC. Besides that, PROS is the worst solution with 95% of confidence, being statistically different even to MFCC, which presents second worst results. Notwithstanding, if we take solely PROS into consideration, it is able to correctly classify one in every four seconds of video (ACC = 76.21%), starting with only three feature values in the low-level video description (due to the prosodic features). That shows a promising suitability for describing video in mobile devices, and for dealing with the tradeoff between efficiency and effectiveness. For instance, a solution with such accuracy may be useful for a quick parental check on mobile storage devices, prior to leaving it in minor hands.

Starting with these results, an interesting investigation is to verify how much effectiveness improvement a fusion of features will lead to. In the next section, we combine the auditory, and the visual features, according to their nature, for having a better understanding on how these modalities contribute to the localization of pornographic content. Later on (see Section 7.2.3), we take a step beyond and consider combining features of different modalities, as well.

7.2.2 Fusion of Solutions with Similar Nature

In Table 7.5, we present the results of combining same-nature solutions (i.e., PROS with MFCC, for being auditory, and HOG with TRoF, for being visual). Regardless of the type of fusion meta-learning (THR, NBC, or SVM), the combined visual features once again outperform the combined auditory features, as expected. Indeed, the single visual solutions (HOG and TRoF) are better than any combination of auditory features (PROS+MFCC). That is shown, for instance, in Table 7.6, in the particular case of the TRoF snippet classifier. TRoF is statistically better than THR-PROS+MFCC, NBC-PROS+MFCC, and SVM-PROS+MFCC, in terms of ACC, with 95% of confidence.

More important, however, is the fact that the fusion of specific features always result in better values for ACC and F_2 measure, when compared to the isolated use of these same features. That gives hints about the expected complementarity of the features, even though, at this point, they are still from similar nature. For example, in the case of auditory features (PROS and MFCC), the baseline THR fusion leads to an error reduction — regarding ACC — of nearly 27%, when compared to the solely PROS-based alternative, and of nearly 15%, when compared to the solely MFCC-based one. Similarly, in the case

Table 7.5: Results of pornography localization over the Pornography-2K dataset, with fusion of snippet classifiers that rely upon features of the same nature (auditory or visual). We report the average performance over the 3×2 cross-validation folds. In all experiments, the standard deviation is lower than 0.04.

	Solution	TPR (%)	TNR (%)	ACC (%)	F_2 (%)
THR	PROS + MFCC	82.84	82.74	82.79	84.21
	HOG + TRoF	93.95	87.54	90.74	93.92
NBC	PROS + MFCC	81.05	81.62	81.33	82.56
	HOG + TRoF	90.74	89.40	90.07	91.42
SVM	PROS + MFCC	82.19	82.05	82.12	83.59
	HOG + TRoF	90.57	90.01	90.29	91.33

TPR: true positive — TNR: true negative rate — ACC: accuracy — F_2: F_2 measure
THR: score thresholding — NBC: Naïve Bayes Classifier — SVM: Support Vector Machine

Table 7.6: Pairwise comparison of the TRoF pornographic snippet classifier and the fusions of audio-based snippet classifiers. We report the statistical tests for all 3×2 folds, considering ACC and using the paired Wilcoxon test with Bonferroni's correction. The solution based solely on TRoF is better than any experimented fusion of audio-based snippet classifiers.

TRoF	PROS + MFCC:	THR	NBC	SVM
	p-value	0.013	0.013	0.030
	conclusion	✔	✔	✔

✔ TRoF is better

Table 7.7: Pairwise comparison of the experimented fusions of HOG and TRoF pornographic snippet classifiers. We report the statistical tests for all 3×2 folds, considering ACC and using the paired Wilcoxon test with Bonferroni's correction.

HOG + TRoF:	THR	NBC
NBC	0.190 •	–
SVM	0.280 •	0.930 •

p-values • not statistically different

of visual features (HOG and TRoF), the baseline THR fusion yields an error reduction — regarding ACC — of around 27% and 31%, when compared to the solely HOG- and TRoF-based solutions, respectively.

Concerning the different types of fusion meta-learning (THR, NBC, or SVM), in the particular case of pornography, we see that the equivalent solutions (e.g., THR-HOG+TRoF, NBC-HOG+TRoF, and SVM-HOG+TRoF) present very close results, for both ACC and F_2 measure. In such direction, Table 7.7 presents the statistical comparison of the three types of fusion that we perform for combining HOG and TRoF pornographic snippet classifiers, for the sake of exemplification. As one might observe, the THR, NBC, and SVM alternatives are not statistically different, wit respect to ACC.

Table 7.8: Results of pornography localization over the Pornography-2K dataset, with multimodal fusion of snippet classifiers. We report the average performance over the 3×2 cross-validation folds. In all experiments, the standard deviation is lower than 0.02.

	Solution	TPR (%)	TNR (%)	ACC (%)	F_2 (%)
THR	MFCC + TRoF	92.65	87.51	90.08	92.76
	ALL	93.53	87.97	90.75	93.53
NBC	MFCC + TRoF	91.15	87.90	89.52	91.61
	ALL	91.62	88.61	90.18	92.04
SVM	MFCC + TRoF	90.87	89.15	90.01	91.47
	ALL	91.32	90.12	90.72	91.93

TPR: true positive — TNR: true negative rate — ACC: accuracy — F_2: F_2 measure
THR: score thresholding — NBC: Naïve Bayes Classifier — SVM: Support Vector Machine

Table 7.9: Pairwise comparison of HOG+TRoF, MFCC+TRoF, and HOG+TRoF+MFCC+PROS (ALL) fusions of pornographic snippet classifiers, all based on score thresholding (THR). We report the statistical tests for all 3×2 folds, considering ACC and using the paired Wilcoxon test with Bonferroni's correction.

	HOG+TRoF	MFCC+TRoF
MFCC+TRoF	0.060 •	–
ALL	1.000 •	0.130 •

p-values • not statistically different

In the following section, we report the results of multimodal fusion, and investigate whether or not the auditory and visual features are complementary for this particular problem.

7.2.3 Multimodal Fusion Solutions

In Table 7.8, we present the results of combining snippet classifiers that rely upon features of different nature (e.g., auditory and visual, a.k.a., multimodal solutions). As one might observe, we evaluate the combination of the best auditory feature with the space-temporal one (MFCC+TRoF), and alternatively, we evaluate a complete fusion, with all the four available snippet classifiers (referred to as ALL, therefore combining PROS, MFCC, HOG and TRoF). The former combination is the one in which the resulting fusion vectors have a size of l=20. In the other ones, in which always two classifiers are combined, the resulting fusion vectors are ten-sized (i.e., l=10).

In all cases, the multimodal combinations are not clearly better than exclusively combining visual features (HOG+TRoF solutions). For the sake of exemplification, Table 7.9 summarizes the statistical comparison of the THR-HOG+TRoF fusion solution (which is solely visual), and the THR-MFCC+TRoF and THR-ALL multimodal pornography locators. As one might observe, such strategies do not present statistical difference, besides presenting very close results. It indicates that the audio-based snippet classifiers do not

produce hits on the occasions in which the visual classifiers miss, and vice-versa. Hence, they may not be complementary.

The possible reasons for the not so impressive performance of the audio-based snippet classifiers may rely on the samples of the Pornography-2k dataset. Many of them depict amateur content, with amateur editing. For instance, it is common to find sexual footage whose moaning sounds are further covered with electronic music, for not denouncing ashamed spectators. Therefore, the herein stated observation about non-complementarity must be considered with a grain of salt. It is particularly true for the dataset we test in this work. However, it is possible that the cited features present complementary properties in face of other datasets, specially when considering professionally-edited and studio pornographic movies.

Finally, once more, the different explored meta-fusion techniques revealed to be fairly similar in terms of results. Anyhow, in the next section, we present a qualitative analysis of the performed localization, in which we point out some benefits and some drawbacks of using machine-learning techniques, in comparison to score thresholding.

7.2.4 Qualitative Evaluation

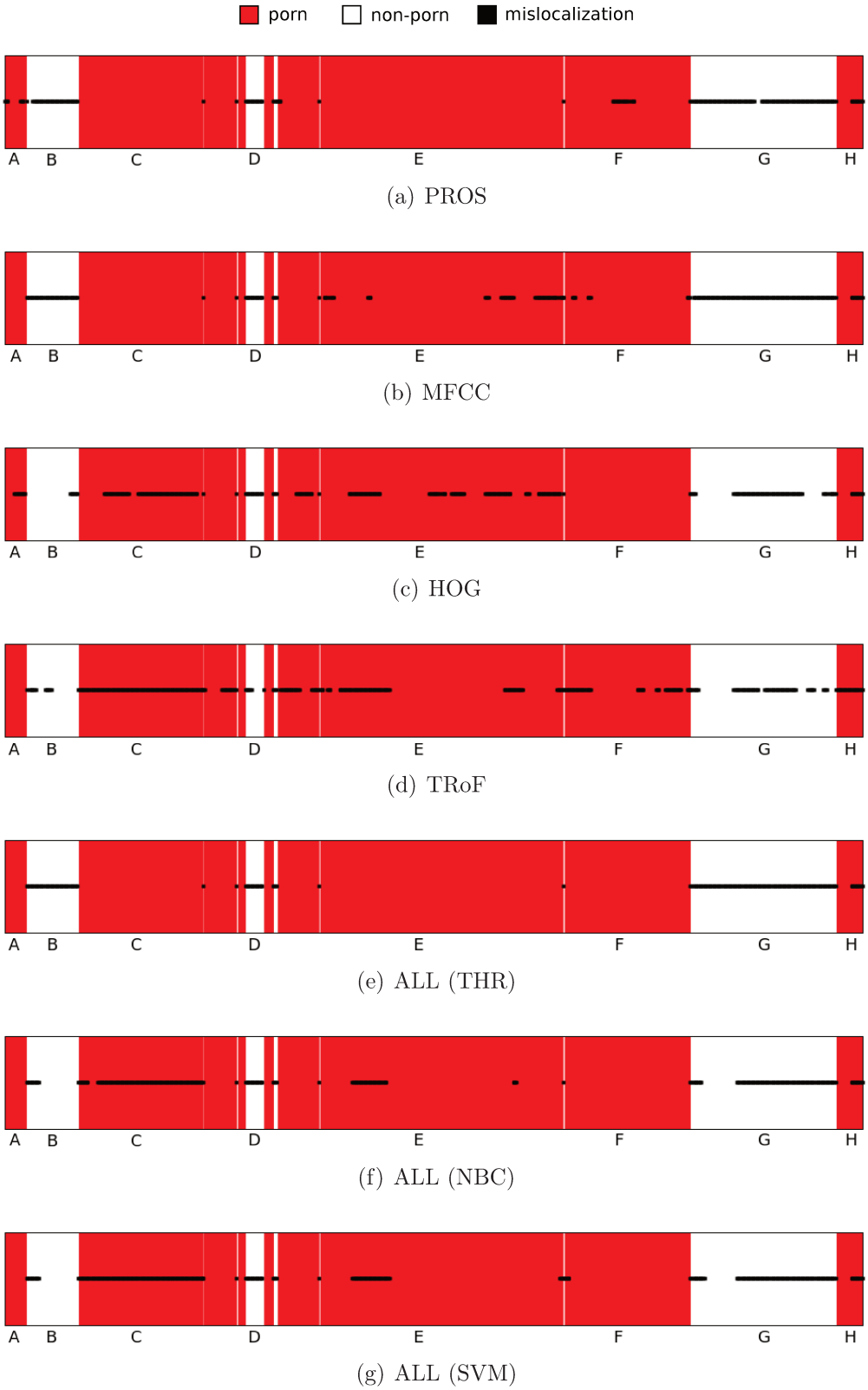
For the sake of illustration, in this section, we provide a qualitative evaluation of two video excerpts that were sampled from the Pornography-2k dataset.

Sample Excerpt 1

Figure 7.2 depicts the quality of pornography localization over a 4.5-minute long video footage, which was sampled from the Pornography-2k dataset, and whose results reveal a difficult case. Given that each row refers to the same footage, they individually represent the same timeline. Red and white areas depict the localization groundtruth: red for positive, and white for negative. As expected, these areas do not change along the boxes. Black dots, in turn, represent mislocalization. Hence, the lesser the quantity of black dots, the better the result of a solution. Moreover, some video segments are labeled with capital letters (from A to H), for further reference.

In Figure 7.2 (a–d), one can observe the localization quality of each single solution, with no fusion of features. In the particular case of the footage at hand, the audio-based snippet classifiers (PROS and MFCC) show a tendency of classifying the content as pornographic, with the PROS-based solution returning a more constant answer (i.e., with less label variations). That explains the higher quantity of hits over the C, and E positive and longer segments (which are, respectively, 40 and 78 seconds long), when compared to the image-based ones (HOG and TRoF), but at the cost of generating more mislocalization over negative segments (false positives), such as B and G (which are, respectively, 16 and 47 seconds long).

The image-based solutions (HOG and TRoF), in turn, result in answers that present more label transitions, and a better capability of detecting negative segments, at the cost of producing more false negatives (e.g., in segments C and E). The TRoF-based solution, in particular, is able to detect part of the more difficult six-second D segment, which is



(a) PROS

(b) MFCC

(c) HOG

(d) TRoF

(e) ALL (THR)

(f) ALL (NBC)

(g) ALL (SVM)

Figure 7.2: Localization quality over a 4.5-minute long Pornography-2k video sample. Each row depicts the same footage, and therefore the same timelime. Red and white areas depict the localization groundtruth: red for positive, and white for negative. Black dots represent the mislocalization of each technique: the lesser the quantity of black dots, the better the result.

mislocated by all the other solutions, including the multimodal ones (THR, NBC, and SVM). However, it provides more false positives in segments *C*, *F*, and *H*.

In opposition to Figure 7.2 (a–d), Figure 7.2 (e–g) depicts the localization quality of the combinations of all available snippet classifiers. Their difference relies upon the used fusion meta-learning technique: THR, NBC, or SVM. As one might observe, the THR solution finds the entire footage as positive, thus presenting a strong recall, but a weak precision. The machine-learning solutions (NBC and SVM), in turn, provide very close results, and a better precision than score thresholding: segments B and part of G are not mistaken as false positives.

Sample Excerpt 2

Figure 7.3 depicts the quality of pornography localization over a 1.5-minute long video footage, which was sampled from the Pornography-2k dataset, and whose results reveal an easy case. The notation is the same of Figure 7.2: each row refers to the same footage, and red and white areas depict the localization groundtruth: red for positive, and white for negative. Black dots, in turn, represent mislocalization, and some video segments are labeled with capital letters (from A to C), for further reference.

In Figure 7.3 (a–d), we show the localization quality of each single solution, with no fusion of features. As one might observe, contrary to the other results, the PROS-based strategy provides a good answer, except for some mislocalization in the points of transition, where the stream changes its sensitiveness (e.g., from segment A to B, and from B to C), and for some false negatives in the one-minute long positive segment B. The MFCC- and HOG-based ones, in turn, result in some additional false positives within the 23-second long negative segment A, while the TRoF-based alternative presents mislocalization only in the points of transition.

Regarding Figure 7.3 (e–g) — which depicts the localization quality of the combinations of all available snippet classifiers — the respective answers present better quality, when compared to the single solutions (they clearly present less black dots), as expected. In addition, while the SVM-based strategy produces some false negatives in the last transition (from segment B to C), the THR- and NBC-based ones perform better, with a few false negatives in the points of transition, and a perfect recall. The NBC-based one, indeed, provides the less quantity of false negatives, along the points of transition.

7.3 Final Remarks

The problem of pornographic content localization is prevailing and relevant. For instance, on the occasion of creating the Pornography-2k dataset, some positive samples were obtained through the biggest video-sharing website on the Internet, without much searching effort. Hence, nobody feels completely safe when their children go online.

Regardless of that, to the best of our knowledge, prior to this research, there was no work in the literature that had systematically approached the problem. There was a lack of a proper dataset, candidate solutions, and metrics, which we had to deal with. Posterior to filling such gap — and coming with the contribution of providing a frame-

(a) PROS

(b) MFCC

(c) HOG

(d) TRoF

(e) ALL (THR)

(f) ALL (NBC)

(g) ALL (SVM)

Figure 7.3: Localization quality over a 1.5-minute long Pornography-2k video sample. Red and white areas depict the localization groundtruth: red for positive, and white for negative. Black dots represent the mislocalization of each technique: the lesser the quantity of black dots, the better the result.

level annotation of the entire 140 hours of the Pornography-2k dataset — we evaluate the use of the sensitive-content localization pipeline that was proposed in Chapter 6, for tackling the problem at hand.

From the preliminary results, reported herein, we verify that we are in the direction of solving the problem although many contributions are still possible. With the best solutions (THR-HOG+TROF, NBC-HOG+TROF, and SVM-HOG+TROF, which present very close results), we fail at identifying around five minutes in every hour of pornographic content, indicating a reasonable content filter. In addition, out of every minute we identify as being pornographic, five seconds are benign, therefore being mistakenly classified as inappropriate. That also indicates a reasonable content analyzer.

Finally, we find the isolated or combined use of auditory features such as MFCC and prosodic less effective than combining visual features, for locating pornographic content. Reasons for that may be related to the poor sound edition of the frequent amateur content, or even the low discriminative ability of the captured speech-related features (since MFCC and prosody are originally aimed at speech recognition).

The best solutions rely upon visual snippet classifiers only (HOG- and TRoF-based), which individually present low-memory footprint, and small processing time. In this vein, with the correct configuration of snippet overlap — which is yet subject to investigations, considering particular mobile platforms and requirements, as we properly indicate as future work, in Chapter 9 — these solutions are suitable for deployment on mobile devices.

Chapter 8

Violence Localization: Experiments

In this chapter, we validate the sensitive-content localization pipeline that was introduced in Chapter 6, for the particular case of violence detection. Similar to the case of pornography detection, we evaluate the combination of diverse snippet classifiers, which individually rely upon the description of different data modalities (from auditory to visual features). Additionally, for the sake of investigation, we compare the pipeline with three works of the literature [29, 99, 55], which ranked best in the MediaEval 2014 subjective violence localization competition [80].

For that, in Section 8.1, we explain the adopted experimental setup, in terms of dataset, experimental protocol, and metrics (which are inherited from the MediaEval 2014 VSD task [80]), in addition to selected snippet classifiers, parametrization, and implementation details. Afterwards, in Section 8.2, we report the experimental results, while in Section 8.3, we present some final remarks.

8.1 Experimental Setup

We have discussed, throughout this work, the importance of the MediaEval VSD task [80, 34, 33], for the problem of violence detection. Once more, to benefit from the provided common groundtruth and standard evaluation protocols, we use the MediaEval benchmark for conducting the experiments. Hence, in Section 8.1.1, we present some details of the MediaEval 2014 dataset [80], which refers to the VSD edition that had evaluated violence localization. Next, in Section 8.1.2, we explain the competition experimental protocol, and the metrics used to evaluate the results, while in Section 8.1.3, we present the multimodal snippet classifiers we have designed and selected for combination. Next, in Section 8.1.4, we define the experimented high-level fusion solutions, focusing on the *Fusion Meta-Learning* task alternatives (please refer to Section 6.1, for details about such task).

8.1.1 MediaEval 2014 Violent Scenes Detection Dataset

The MediaEval 2014 VSD dataset [80] is built on top of the 2013 edition (please refer to Section 5.1.1, for details). All the movies used in 2013, but *Kill Bill*, are adopted as the new training set (thus comprising 24 titles). The new test set, in turn, comprises seven

Table 8.1: MediaEval 2014 VSD dataset summary. The dataset is divided into three parts, namely a 24-title *Hollywood Training* set, a seven-title *Hollywood Test* set, and a set with 86 clips from *YouTube*. For all samples, the competition provides annotations that indicate the violent video segments, with starting and ending frame numbers.

	Title	violence (%)
Hollywood Training	01. Armageddon	7.78
	02. Billy Elliot	2.46
	03. Dead Poets Society	0.58
	04. Eragon	13.26
	05. Fight Club	15.83
	06. Harry Potter V	5.44
	07. I am Legend	15.64
	08. Independence Day	13.13
	09. Leon	16.36
	10. Midnight Express	7.12
	11. Pirates of the Caribbean I	18.15
	12. Reservoir Dogs	30.41
	13. Saving Private Ryan	33.95
	14. The Bourne Identity	7.18
	15. The Sixth Sense	2.00
	16. The Wickerman	6.44
	17. The Wizard of Oz	1.02
	18. Fantastic Four I*	20.53
	19. Fargo*	15.04
	20. Forrest Gump*	8.29
	21. Legally Blond*	0.00
	22. Pulp Fiction*	25.05
	23. The Godfather*	5.73
	24. The Pianist*	15.44
	Total — 50h02m	**12.35**

	Title	violence (%)
Hollywood Test	01. 8 Mile	4.70
	02. Braveheart	21.45
	03. Desperado	31.94
	04. Ghost in the Shell	9.85
	05. Jumanji	6.75
	06. Terminator II	24.89
	07. V for Vendetta	14.27
	Total — 13h53m	**17.18**
YouTube — 2h03m — 86 clips		**44.47**

* 2013 test dataset

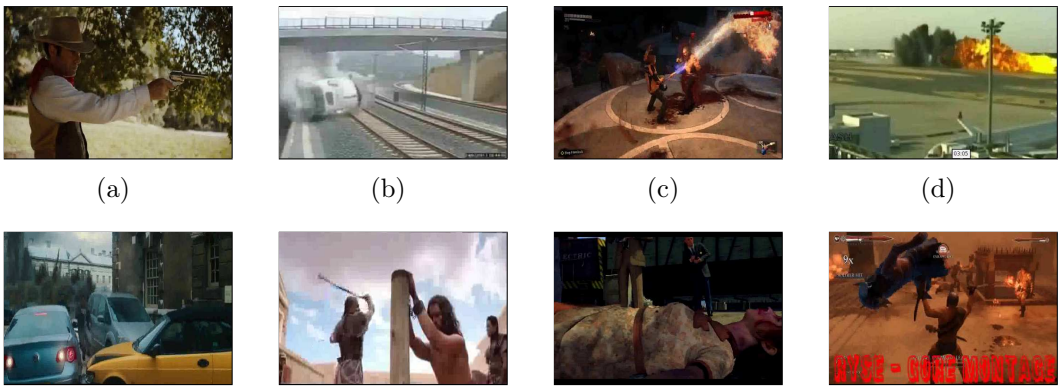
additional Hollywood titles, which must also be purchased, in the same manner as the previous movies, due to copyright issues. In addition to these 31 titles, the MediaEval 2014 VSD dataset also comprises a second minor dataset, composed of 86 *YouTube* clips, which may be from six seconds to six minutes long. In this particular case, these clips are provided within the dataset for free, since they are licensed under Creative Commons regulation.

The provided annotations, in turn, do not support shot segmentation anymore. With the intent of challenging participants to perform violent content localization, the 2014 edition counts on frame-level annotations of all violent video segments, which are individually identified by their starting and ending frame numbers. These annotations had been carried out by several human assessors, in a hierarchical bottom-up revision manner [80]. For the annotators, violent segments were the ones that a person would not let an eight-year-old child watch, due to physical violence [80].

In Table 8.1.1, we summarize the content of the MediaEval 2014 VSD dataset, with the percentages of violent segments per title. The values were collected by [80]. For the sake of illustration, Figure 8.1 depicts some violent frames from the *YouTube* dataset.

8.1.2 Experimental Protocol and Metrics

As already mentioned, the VSD task motivation is the development of systems that may help users choose suitable titles for their children, by retrieving the most violent movie parts, for parental preview [33]. As a consequence, competitors' solutions are compared



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

(e) (f) (g) (h)

Figure 8.1: Violent frames sampled from the MediaEval 2014 VSD dataset, concerning the *YouTube* clips. In (c), (g), and (h), we have clips that were obtained from video games. All images are licensed under Creative Commons regulation.

from the perspective of retrieval, with the support of the Mean Average Precision (MAP) metric, which was properly explained in Section 5.1.2.

In the particular case of the 2014 edition, participants can provide any segmentation of the target video stream (in terms of segment sizes and positions), for attributing labels and confidence scores to each segment. As a consequence, competitors' segments may coincide only partially with the groudtruth segments, and the tested systems may also provide various small segments that fit and hit an eventual larger one, from the groundtruth. For dealing with these situations, MediaEval organizers propose a variation on the calculation of the hits (and thus of the precision), on the occasion of measuring MAP. For the sake of being fair, they only consider a segment prediction as a hit, if it overlaps with the corresponding groundtruth segment by more than 50%. In addition, to deal with the situation of evaluating many small segments, several hits on the same groundtruth segment only count as one true positive. The other hits are just ignored, for not raising the value of MAP inappropriately. For such variation of MAP calculation, they refer to as MAP2014.

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

Finally, given the nature of our approach — in which we have two moments of data learning, (i) one related to the snippet classification learning, and (ii) the other related to the fusion meta-learning — we separate the seven movies that belong to the 2013 test set[1], and 26 clips from the *YouTube* set, for exclusively using in the fusion meta-learning step.

[1]These seven movies are highlighted with asterisks (*), in Table 8.1.1.

8.1.3 Multimodal Snippet Classifiers

For evaluating violent content localization, we employ the exact same four snippet classifiers that were used for pornography content localization. Therefore, we evaluate diverse combinations of the following strategies:

- Two audio-based classifiers, namely MFCC (based upon MFCC [32] features), and PROS (based upon prosodic features [40]).
- Two image-based classifiers, namely HOG (based upon a dense application of the HOG descriptor [30]), and TRoF (based upon the proposed TRoF descriptor).

Each classifier is an expert in classifying five-second snippets, and returns confidence classification scores in the real interval [0..1]. For more details about these classifiers, please refer to Section 7.1.3.

8.1.4 Fusion Meta-Learning Solutions

For implementing the *Fusion Meta-Learning* task (please refer to Section 6.1), we employ the exact same three strategies that were used for pornography content localization. Therefore, we explore the following solutions:

- Score Thresholding (THR), a representative of the overall idea of the winner-takes-all.
- Naïve Bayes Classifier (NBC), a representative of generative strategies.
- Support Vector Machine (SVM), a representative of discriminative strategies.

For details concerning these solution, please refer to Section 7.1.4.

Finally, similar to the pornographic case, regardless of the used fusion meta-learning method, in the test system operation, we always convolve a Gaussian window with standard deviation $\sigma = 5s$ over the temporal sequence of obtained fusion scores, for smoothing. Missing values within the score fusion vector — which are related to the ϵ value, in Figure 6.3 — are filled with a linear interpolation of the present values. Afterward, we assume a fusion score threshold of $t = 0.5$, and we replace all the time-adjacent scores equal to, or greater than $t = 0.5$, by their average value. Complementarily, all the time-adjacent scores smaller than $t = 0.5$ are replaced by their own average value.

8.2 Results

In this section, we present the results of the proposed fusion pipeline, on the occasion of combining the snippet classifiers that were enlisted in Section 8.1.3, in varied manners. First, in Section 8.2.1, we present the individual results of the snippet classifiers, in face of the problem of violence localization, without combinations. Thereafter, in Section 8.2.2, we present the results of combining snippet classifiers that rely upon the same type of

Table 8.2: Results of violence localization over the MediaEval 2014 VSD dataset, without fusion of snippet classifiers. We report the MAP2014 official competition metric.

		Solution	MAP2014
No-fusion	audio	PROS	0.402
		MFCC	0.288
	image	HOG	0.299
		TRoF	0.401

low-level feature (i.e., we provide the results of fusions of audio-based classifiers, and of fusions of image-based classifiers, separately). Next, in Section 8.2.3, we present the results of multimodal combinations, while in Section 8.2.4, we present some graphs that depict the quality of the violence localization, over one selected title from the MediaEval 2014 VSD test set.

8.2.1 Single Solutions

In Table 8.2, we present the individual results of the snippet classifiers, without combinations, in face of the problem of violent content localization. We report the MAP2014, which is the official MediaEval VSD competition metric (see Section 8.1.2).

As one might observe, in the particular case of violence localization, and in opposition to pornography localization (see Section 7.2.1), auditory and visual features are equally suitable for the task, with the PROS-based alternative presenting the highest MAP2014, indeed. That may be related to the high sound edition quality of the Hollywood movies, which also follow a well-established grammar for affecting spectators. Moreover, we also verify that motion is an important feature for violence detection. While in the pornographic case, the HOG- and TRoF-based solutions are equivalently good, in the present situation, the still-image HOG-based solution presents a much inferior result, when compared to the space-temporal TRoF-based one.

In the following two sections, we verify how much effectiveness improvement a fusion of features leads to, when compared to these single solutions. Particularly, in the next section, we combine the auditory, and the visual features, according to their nature, for having a better understanding on how these modalities contribute to the localization of violent content. Later on, in Section 8.2.3, we take a step further, and combine features of different modalities.

8.2.2 Fusion of Solutions with Similar Nature

In Table 8.3, we present the results of combining same-nature solutions (i.e., PROS with MFCC, for being auditory, and HOG with TRoF, for being visual). Contrary to the cases of pornography localization, in violence detection, the THR fusion of same-nature features does not work quite well. It therefore leads to worse results of MAP2014, when compared to any single solution (see Table 8.2). Notwithstanding, the NBC- and SVM-based fusions of features lead to better results, specially in the NBC case.

Table 8.3: Results of violence localization, with fusion of snippet classifiers that rely upon features of the same nature (auditory or visual). We report the MAP2014 official competition metric.

	Solution	MAP2014
THR	PROS + MFCC	0.374
	HOG + TRoF	0.324
NBC	PROS + MFCC	0.453
	HOG + TRoF	0.473
SVM	PROS + MFCC	0.419
	HOG + TRoF	0.406

MJV: majority voting — NBC: Naïve Bayes Classifier — SVM: Support Vector Machine

For instance, in the case of auditory features (PROS and MFCC), the NBC fusion leads to an improvement in MAP2014 of nearly 12%, when compared to the solely PROS-based alternative, and of nearly 57%, when compared to the solely MFCC-based one. Similarly, in the case of visual features (HOG and TRoF), the NBC fusion yields an improvement in MAP2014 of around 58% and 18%, when compared to the solely HOG- and TRoF-based solutions, respectively.

In the following section, we report the results of multimodal fusion, and verify whether or not the auditory and visual features are complementary, for this particular problem.

8.2.3 Multimodal Fusion Solutions

In Table 8.4, we present the results of combining snippet classifiers that rely upon features of different nature (e.g., auditory and visual, a.k.a., multimodal solutions). We explore the combination of the best auditory feature, with the best visual one (PROS+TRoF), and alternatively, we explore a complete fusion, with all the four available snippet classifiers (referred to as ALL, therefore combining PROS, MFCC, HOG and TRoF).

In the case of combining all the features (ALL solutions), the fusions are not clearly better than exclusively combining only auditory (PROS+MFCC), or only visual features (HOG+TRoF solutions), with NBC or with SVM. It indicates that some auditory features are mistakenly canceling the hits of the visual ones — and vice-versa — revealing an absence of complementarity.

More important than that, though, is the fact that the multimodal combination of PROS and TRoF (the former auditory, and the later visual) leads to the best solution, thus far. The SVM-PROS+TRoF combination provides a MAP2014 of 0.502. It indicates that the auditory PROS-based snippet classifier produces hits on the occasions in which the visual TRoF-based one misses, and vice-versa. Hence, they are complementary, and we are able to take benefits from that. In the following section, we provide a qualitative analysis of the violence localization that is provided by such solution.

Table 8.4: Results of violence localization over the MediaEval 2014 VSD dataset, with multimodal fusion of snippet classifiers. We report the MAP2014 official competition metric. In bold, we highlight the best result.

	Solution	MAP2014
THR	PROS + TRoF	0.460
	ALL	0.406
NBC	PROS + TRoF	0.488
	ALL	0.476
SVM	**PROS + TRoF**	**0.502**
	ALL	0.397

MJV: majority voting — NBC: Naïve Bayes Classifier — SVM: Support Vector Machine

8.2.4 Qualitative Evaluation

For the sake of illustration, in this section, we present a qualitative evaluation of violence localization, over ten minutes that were selected from the *Jumanji* movie title. The localization is provided by the best multimodal solution (SVM-PROS+TRoF), which — during the experiments — obtained the highest MAP2014 over the MediaEval 2014 VSD test set.

Figure 8.2 depicts the ten-minute timeline, with groundtruth and system answer. Red and white areas refer to the localization groundtruth: red regards violent scenes, while white regards non-violent scenes. Black dots, in turn, represent mislocalization. In addition, segments of interest are properly identified by capital letters (from A to C).

As one might observe, along the observed ten minutes of video footage, we have many occurrences of false positives (which are related to the black dots that lie within the white regions), and of false negatives (which are related to the black dots that lie inside the red regions). Moreover, except for the first quarter of the footage at hand — which presents arbitrary false positives — the mislocalizations are concentrated around the regions of label transition (i.e., the instants when the scene changes from positive to negative, or vice-versa.

In order to understand the eventual difficulties faced by the proposed solution over the regions of transition, we focus on a particular sequence of the footage, which is related to the segments A, B, and C, and the transitions among them. Figure 8.3 depicts some frames that comprise such segments. In Figure 8.3 (a–d), we have the frames related to segment A, which is non-violent, although such frames are mistakenly labeled as positive. In Figure 8.3 (e–h), in turn, we have the frames related to segment B, which is violent, and whose frames are correctly identified as such. Finally, in Figure 8.3 (i–l), we have the frames related to segment C, which is non-violent, in spite of such frames being labeled as positive.

As one might observe, the violent scene — which is correctly detected and is related to Figure 8.3 (e–h) and to segment B — depicts a scene with panicked people, who are being attacked by an alligator.

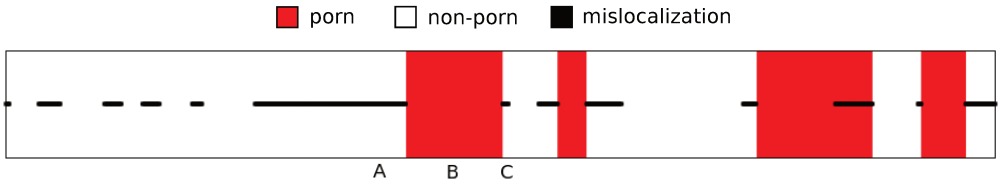


Figure 8.2: Localization quality over a ten-minute long footage that was sampled from the *Jumanji* movie title. The localization was provided by the multimodal SVM-PROS+TRoF solution. Red and white areas depict the localization groundtruth: red for positive, and white for negative. Black dots represent the mislocalization.

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

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

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

Figure 8.3: Frames sampled from the *Jumanji* movie title. In (a–d), we have a prior sequence of false positive frames that were sampled from segment *A*, within Figure 8.2. In (e–h), we have a middle sequence of true positive frames that were sampled from segment *B*. In (i–l), we have a posterior sequence of false positive frames that were sampled from segment *C*. All images are copyrighted, and therefore belong to Sony/Columbia.

Prior to that, segment *A* — represented by Figure 8.3 (a–b) — depicts a scene with the same studio setup of segment *B*. Although the groundtruth tells the opposite, the action already regards a flooded room, with apprehensive players and motion on water. The alligator, though, is not present yet. In such context, one might argue that the scene is already tense, indeed indicating a difficult transition.

Posterior to the violent scene, the studio setup changes completely, becoming outdoor (see Figure 8.3 (i–l), which regards segment *C*. However, we point out some elements that may turn that transition also difficult to cope with. First, the players are clearly tense, what might be captured by the prosody descriptor. Second, the police officer is holding a gun — see Figure 8.3 (l) — which is an action that is present in many positive scenes throughout the dataset, and the motion-aware TRoF descriptor might capture as well.

Table 8.5: Best result of violence localization in contrast with the literature. All works report the MAP2014 official competition metric.

Solution	MAP2014
Dai et al. [29]	0.630
Zhang et al. [99]	0.566
Lam et al. [55]	0.564
SVM-PROS+TRoF (proposed)	0.502

Although such example is a small one, considering the size of the MediaEval 2014 VSD dataset, it gives a hint about how difficult the localization task is. We therefore conclude the qualitative experiments.

8.3 Final Remarks

Thanks to the MediaEval initiative, we can compare the proposed solutions with the current state of the art of violence localization. In Table 8.5, we put the best multimodal approach herein proposed in perspective with three works from the literature, whose authors usually attend the MediaEval VSD task competition and that are worth highlighting.

As one might observe, we report a modest and inferior value for the official competition metric, although it is not so far from the mentioned publications. Nevertheless, in face of such numbers, there are some considerations that we find important to take into account, when analyzing such performances.

First and foremost, all three works make use of more than one combination of over three content classifiers, that rely upon diverse auditory and visual features[2]. Within those features, the use of time consuming space-temporal approaches is prime for obtaining a high effectiveness, specially regarding the Dense Trajectories [93], which represent the current state of the art of space-temporal video description for action recognition. Dense Trajectories, however, are computationally expensive, and present a high-memory footprint, as we have verified in Section 4.2.2. Additionally, the works of Lam et al. [55], and of Dai et al. [29] also rely upon deep neural networks, for obtaining the reported results. These solutions, therefore, will certainly not meet the restrictions of low-memory footprint and processing time that strongly characterize the current use of mobile devices, such as tablets and smartphones, at least under their current configurations (Southern Hemisphere Winter, 2016).

The SVM-PROS+TRoF solution, on the contrary, relies upon the use of only two classifiers, which individually present low-memory footprint and small processing time. While TRoF was conceived aiming at efficient video description, prosody is an auditory feature that presents the impressive characteristic of delivering only three values for each low-level feature vector. To the best of our knowledge, no other low-level descriptor presents such a low-memory footprint.

[2]For more details regarding these works, please refer to Section 2.3, and to Table 2.3.

In this vein, with the correct configuration of snippet overlap — which is yet subject to investigation, as we point out in Chapter 9 — the proposed solution is suitable for deployment on mobile devices.

Finally, when solving the problem of violence localization, we perceived many differences with respect to the problem of pornography localization. Although we have used the same high-level fusion pipeline for both situations, the explored setups led to distinct conclusions. For instance, audio is a negligible feature, in the particular case of pornographic localization, at least, for the particular datasets considered in this work. In the case of violence, though, it is paramount. Moreover, motion-aware and still-image descriptors present close performance, when employed in pornographic setups. However, that is not true for violence localization, whereby the TRoF-based solution significantly outperformed the HOG-based one.

In any case, the fusion pipeline could be nicely adapted for each situation, while still relying upon the classification and fusion of time-overlapping video snippets. That is related to hypothesis *H2* (please refer to Section 1.1), which states that it is possible to localize sensitive content within the video timeline by means of the classification and fusion of time-overlapping video snippets. In face of the presented and discussed results, we found strong evidence that such hypothesis is true.

Chapter 9

Conclusions and Future Work

In this chapter, we conclude this work. For that, in Section 9.1, we present the conclusions we have drawn, as a consequence of the conduction of the research, and of tackling the problem of sensitive-video analysis, while in Section 9.2, we elaborate on possible future work.

9.1 Conclusions

The analysis of sensitive video is a relevant task, due to the potential harm of sensitive content. The present high pervasiveness and big-data nature of digital video demands the use of automatic solutions, for performing such task. Notwithstanding, on the occasion of annotating the Pornography-2k dataset, we could realize how difficult it is to detect sensitive content, given the subjectivity and context dependency of the target concepts. Such characteristic hinders the design of fully automatic and computer-aided solutions, in face of the discrete and deterministic nature of computers. To make things worse, it is sometimes necessary to perform the task swiftly, because of urgent situations, such as saving violently injured people, or catching red-handed criminals.

From the gained experience, we could find that pornography detection is easier to perform than violence detection. That is related to the different levels of subjectivity that are inherent to each concept. Pornography is more explicit, as one might observe through the adopted definition, from Short et al. [81]: "any *explicit* sexual matter with the purpose of eliciting arousal". The challenges of detecting it rely mainly upon the necessity of reducing false positives, for instance in benign situations of nudity, such as baby breastfeeding, breast exam, body anatomy classes, etc. Violence, on the contrary, has a more subjective concept, which is borrowed from the MediaEval initiative [33, 80]: situations "one would not let an eight-year old child see". The challenges of performing violence detection do not rest only on reducing false negatives, but also on improving the number of true positives, given the variety of positive samples, which may range from people fighting, to car crashes, to injuries, to felony, and to gun threatening. That is indeed reflected throughout the results we have obtained, which are always better in the pornographic cases. Of course, if we intend to refine the pornography detection case to child pornography, the difficulty level is certainly increased and we envision that as a

possible future work worth pursuing.

Moreover, regarding the particular case of content localization, we could verify important differences between pornography and violence. For pornography localization, audio is negligible, and space-temporal features perform as well as still-image features. As already discussed, the audio aspect might be related to the abundance of pornographic amateur content, whose audio streams have nothing to do with the visual content, due to poor edition, compression, or stealth purposes. Concerning the space-temporal vs. still-image features, the best approach actually regards a combined use of both, since they seem to be very complementary, in the pornographic case. For violence localization, audio is a differential for improving effectiveness, and space-temporal approaches strongly outperform still-image solutions. In this case, it is worth mention that the violent dataset is mostly composed of Hollywood titles, which present professional special sound effects, and controlled camera pace rates. The datasets for pornographic and for violent content localization are thus really distinct, not only in content, but also in film grammar (studio vs. amateur).

In face of the proposed solutions, a natural question regards defining when one should use them, and when one should not. In the particular case of sensitive-content classification, the BoVW- and TRoF-based pipeline is indicated for generalization problems, in which the system is expected to learn, from a training dataset, the characteristics that better represent a class, or that better separate the elements from different classes. It thus may be considered for quickly separating video footage in two-class scenarios, with low-memory footprint requirements, as long as one have a properly annotated training dataset at hand. Concerning inappropriate applications, we do not expect such pipeline to be useful for solving specialization problems, in which one searches for an instance in particular (e.g., detection of near duplicates, landmark recognition, person identification, etc.), although we have not experimented with any of these problems.

With respect to the proposed high-level fusion pipeline, we find it useful for content retrieval setups, in which one may want to take a look at the top k more *relevant* scenes of a given stream, in order to decide if it is safe for disclosure, or for inclusion in a particular category of interest. Again, the *relevance* of content must be related to a generalization problem, instead of a specialization one. It might be used, for example, for localizing frames with fallen people, but not for tracking a specific person along frames.

The problem of sensitive-video analysis is far from being solved, and we have systematically tackled only a portion of its facets: classification and localization of pornographic and violent content. For doing so, however, we have found strong evidences that the hypotheses of the present research are true:

- It is possible to efficiently use video temporal information for effective sensitive-content classification, regarding low-memory footprint and small processing time, by combining simplified space-temporal video interest-point detection and description, with entire-footage representation through a single feature vector.

- It is possible to localize sensitive content within the video timeline by means of the classification and fusion of time-overlapping video snippets.

In practical terms, we designed solutions efficient enough for deployment on mobile devices with limited hardware, such as tablets and smartphones. We indeed have delivered two functional prototypes to Samsung Electronics (one for violence and pornography classification, and the other for violence and pornography localization), as a part of a major project that was coordinated by professor Anderson Rocha, and to which this research was linked.

Finally, we may admit that we have reached the goal of designing and developing effective and efficient methods for sensitive-content classification and localization. Nonetheless, there is yet room for further work, as we point out in the next section.

9.2 Future Work

In this section, we discuss some issues that we could not address in the present research, due to limitations of time and scope, and that we leave as future work.

We have tackled the problem of sensitive-video analysis in two fronts: (i) as a decision problem, which is related to the task of sensitive-video classification, and (ii) as a search problem, which is related to the task of sensitive-content localization. In this vein, a third front is to treat the problem as an optimization one, whereby somebody might want to localize not any occurrence of sensitive content, within a target video stream, but instead the occurrence of a particular one, which minimizes the cost, or maximizes the gain of a problem-dependent objective function. That is useful, for instance, in Forensic scenarios, in which one might want to track the behavior of a particular person, which had been previously identified as a criminal. Or for movie industry purposes, whereby an enthusiastic might want to see only the scenes from a target stream where a specific actor or actress appears. Or for fine-grained video categorization, regarding, for instance, only retrieving the titles that depict a particular landmark, etc. As we have already mentioned, the solutions herein proposed are not directly appropriate for solving such type of problems, revealing that some investigation is needed, in order to point out if there is any manner to adequate them.

Still in the direction of sensitive-content classification and localization, we have explored pornographic and violent content. Nevertheless, the representatives of sensitive content are untold, including, only to name a few, child abuse, elder abuse, child pornography, cruelty to animals, humiliation, murder, etc. With only pornography and violence, we were able to identify relevant differences, which directly impact on how the proposed pipelines shall be instantiated for each concept. For instance, we have already mentioned that audio is negligible for pornography, but not for violence, and so on, at least for the datasets and cases considered in this work. Therefore, all the remaining sensitive concepts are out there to be analyzed, for helping us draw even more interesting and striking conclusions.

Concerning the design of efficient solutions, that are suitable for deployment on mobile devices, we have considered only memory footprint and processing time, for reporting the efficiency of the solutions. However, mobile devices have another feature that is equally important, which is power consumption, since they depend upon batteries for operating.

Hence, understanding how much battery each strategy saves, when running on mobile devices, is an additional and interesting research direction, which demands knowledge about mobile architectures, and device benchmarking.

Regarding TRoF, which is a new space-temporal interest point detector, and video content descriptor, we have tested it only for sensitive-video analysis. Notwithstanding, there is a vast literature of video description for action recognition, to which the proposed detector and descriptor can also be put in perspective.

With respect to the proposed high-level fusion pipeline, there are some particular issues the are left for investigation, and that may provide improvements on the reported results:

- The combination of classifiers that are experts in detecting snippets with different sizes, other than the five seconds herein explored.
- The establishment of the optimal snippet overlapping rate, when describing the target video stream, considering the effectiveness vs. efficiency tradeoff.
- The use of other machine learning techniques in the fusion meta-learning step, such as random forests, or SVMs with other kernels different than RBF.

Taking into consideration the current popularization and impressive results of deep neural networks, it is worth considering putting them in perspective with the solutions proposed herein, as well as investigating appropriate forms of combining them, and exploring their complementarity, if existent.

Finally, all the reported results reveal that there is still room for improvements, specially in the case of violence detection, which by itself reveals to be a problem far more complex than one might foresee at first.

Bibliography

[1] Esra Acar, Frank Hopfgartner, and Sahin Albayrak. Violence Detection in Hollywood Movies by the Fusion of Visual and Mid-level Audio Cues. In *ACM Intl. Conference on Multimedia (MM)*, pages 717–720, 2013. 37, 40, 107

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Good Practice in Large-Scale Learning for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3):507–520, 2014. 24, 46

[3] Pradeep Atrey, Anwar Hossain, Abdulmotaleb El Saddik, and Mohan Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Springer Multimedia Systems*, 16(6):345–379, 2010. 88

[4] Sandra Avila, Daniel Moreira, Mauricio Perez, Daniel Moraes, Isabela Cota, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. RECOD at MediaEval 2014: Violent Scenes Detection Task. In *MediaEval*, pages 1–2, 2014. 20, 87

[5] Sandra Avila, Daniel Moreira, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Método Multimodal e em Tempo Real para Filtragem de Conteúdo Sensível, 4 2016. BR 10 2016 007265 4. 20, 87

[6] Sandra Avila, Daniel Moreira, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Multimodal and Real-Time Method for Filtering Sensitive Media, 6 2016. US 15/198,626. 20, 87

[7] Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo Araújo. BOSSA: Extended BOW Formalism for Image Classification. In *IEEE Intl. Conference on Image Processing (ICIP)*, pages 2909–2912, 2011. 26, 27, 30, 33, 62

[8] Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo Araújo. Pooling in Image Representation: the Visual Codeword Point of View. *Elsevier Computer Vision and Image Understanding (CVIU)*, 117(5):453–465, 2013. 25, 26, 27, 30, 31, 33, 61, 62

[9] Emma Barnett and Iain Hollingshead. The dark side of Facebook. `http://www.telegraph.co.uk/technology/facebook/9118778/The-dark-side-of-Facebook.html` (accessed May 3, 2016). 15

[10] Mark Bartsch and Gregory Wakefield. Audio Thumbnailing of Popular Music Using Chroma-Based Representations. *IEEE Transactions on Multimedia (TMM)*, 1(1):96–104, 2005. 35

[11] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Elsevier Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008. 25, 29, 50, 51, 53, 55, 57, 61, 65, 68

[12] Alierza Behrad, Mehdi Salehpour, Meraj Ghaderian, Mahmoud Saiedi, and Mahdi Barati. Content-based obscene video recognition by combining 3D spatiotemporal and motion-based features. *Springer EURASIP Journal on Image and Video Processing (JIVR)*, 2012(23):1–17, 2012. 31

[13] Fabrice Bellard. FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. `http://www.ffmpeg.org` (accessed May 22, 2016), 2016. 65, 79

[14] Enrique Bermejo, Oscar Deniz, Gloria Bueno, and Rahul Sukthankar. Violence Detection in Video Using Computer Vision Techniques. In *Springer Computer Analysis of Images and Patterns*, pages 332–339, 2011. 25, 26, 36, 37, 40, 59

[15] Blue Coat Systems. K9 Web Protection. `http://www.k9webprotection.com` (accessed March 4, 2016). 31

[16] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2559–2566, 2010. 24, 26

[17] Gary Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 1(1):1–6, 2000. 65, 67, 78, 79, 104, 107, 108

[18] Carlos Caetano, Sandra Avila, Silvio Guimarães, and Arnaldo Araújo. Pornography Detection using BossaNova Video Descriptor. In *IEEE European Signal Processing Conference (EUSIPCO)*, pages 1681–1685, 2014. 25, 27, 30, 31, 33, 61

[19] Carlos Caetano, Sandra Avila, Silvio Guimarães, and Arnaldo Araújo. Representing local binary descriptors with BossaNova for visual recognition. In *ACM Symposium On Applied Computing (SAC)*, pages 49–54, 2014. 27, 30, 31, 33, 61

[20] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011. 110

[21] Kevin Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference (BMVC)*, pages 1–12, 2011. 24, 48, 106

[22] Liang-Hua Chen, Hsi-Wen Hsu, Li-Yun Wang, and Chih-Wen Su. Violence Detection in Movies. In *IEEE Intl. Conference on Computer Graphics, Imaging and Visualization (CGIV)*, pages 119–124, 2011. 35

[23] Ming-yu Chen and Alexander Hauptmann. MoSIFT: Recognizing Human Actions in Surveillance Videos. Technical report, Carnegie Mellon University, School of Computer Science, 2009. 36

[24] Wen-Huang Cheng, Wei-Ta Chu, and Ja-Ling Wu. Semantic Context Detection based on Hierarchical Audio Models. In *ACM Intl. Workshop on Multimedia Information Retrieval (SIGMM)*, pages 109–115, 2003. 36

[25] Content Watch Holdings Inc. Net Nanny. `http://www.netnanny.com/` (accessed March 4, 2016). 31

[26] Council on Communications and Media. Policy Statement – Media Violence. *AAP Pediatrics*, 124(5):1495–1503, 2009. 15, 16, 34

[27] Franklin Crow. Summed-area tables for texture mapping. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 207–212, 1984. 53

[28] CyberPatrol Inc. CyberPatrol Parental Controls. `http://www.cyberpatrol.com/` (accessed March 4, 2016). 31

[29] Qi Dai, Zuxuan Wu, Yu-Gang Jiang, Xiangyang Xue, and Jinhui Tang. Fudan-NJUST at MediaEval 2014: Violent Scenes Detection Using Deep Neural Networks. In *MediaEval*, pages 1–2, 2014. 27, 39, 40, 97, 98, 107, 119, 127

[30] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 37, 55, 57, 75, 79, 106, 107, 122

[31] Ankur Datta, Mubarak Shah, and Niels Lobo. Person-on-Person Violence Detection in Video Data. In *IEEE Intl. Conference on Pattern Recognition (ICPR)*, pages 433–438, 2002. 34, 36

[32] Steven Davis and Paul Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing (TSP)*, 1(1):357–366, 1980. 35, 106, 122

[33] Claire-Hélène Demarty, Bogdan Ionescu, Yu-Gang Jiang, Vu Lam, Markus Schedl, and Cédric Penet. Benchmarking Violent Scenes Detection in Movies. In *IEEE Intl. Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2014. 37, 75, 76, 77, 80, 82, 119, 120, 129

[34] Claire-Hélène Demarty, Cédric Penet, Mohammad Soleymani, and Guillaume Gravier. VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation. *Springer Multimedia Tools and Applications*, 74(17):7379–7404, 2015. 37, 75, 82, 119

[35] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *ACM Journal of Machine Learning Research (JMLR)*, 7:1–30, 2006. 63, 105

[36] Nadia Derbas and Georges Quénot. Joint Audio-Visual Words for Violent Scenes Detection in Movies. In *ACM Intl. Conference on Multimedia Retrieval (ICMR)*, pages 1–4, 2014. 37, 40, 75, 80, 81, 107

[37] Tomas Deselaers, Lexi Pimenidis, and Hermann Ney. Bag-of-visual-words models for adult image classification and filtering. In *IEEE Intl. Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. 25, 26, 28, 29, 33

[38] Thomas Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *ACM Neural Computation*, 10(7):1895–1923, 1998. 63, 105

[39] Tadilo Endeshaw, Johan Garcia, and Andreas Jakobsson. Classification of indecent videos by low complexity repetitive motion detection. In *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7, 2008. 31

[40] Florian Eyben, Martin Wöllmer, and Björn Schuller. openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor. In *ACM Intl. Conference on Multimedia (MM)*, pages 1459–1462, 2010. 107, 122

[41] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research (JMLR)*, 9(1):1871–1874, 2008. 65, 78, 107

[42] Margareth Fleck, David Forsyth, and Chris Bregler. Finding Naked People. In *Springer European Conference on Computer Vision (ECCV)*, pages 593–602, 1996. 28

[43] David Forsyth and Margareth Fleck. Automatic Detection of Human Nudes. *Springer Intl. Journal on Computer Vision (IJCV)*, 32(1):63–77, 1999. 28

[44] Theodoros Giannakopoulos, Alexandros Makris, Dimitrios Kosmopoulos, Stavros Perantonis, and Sergios Theodoridis. Audio-visual fusion for detecting violent scenes in videos. In *Springer Artificial Intelligence: Theories, Models and Applications*, pages 91–100, 2010. 35

[45] Yu Gong, Weiqiang Wang, Shuqiang Jiang, Qingming Huang, and Wen Gao. Detecting Violent Scenes in Movies by Auditory and Visual cues. In *Springer Advances in Multimedia Information Processing (PCM)*, pages 317–326, 2008. 35

[46] Kentaro Hayashi, Makito Seki, Takahide Hirai, Takeuchi Koichi, and Sasakawa Koichi. Real-Time Violent Action Detector for Elevator. In *SPIE Optomechatronic Technologies*, pages 60510–60510, 2005. 34

[47] Tin Kam Ho. Random Decision Forests. In *IEEE Intl. Conference on Document Analysis and Recognition (CDAR)*, pages 278–282, 1995. 26

[48] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification. `https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf` (accessed June 6, 2016), 2003. 95, 110

[49] Hyperdyne Software. Snitch Plus. `http://www.hyperdynesoftware.com/` (accessed March 4, 2016). 31, 64, 68

[50] International Centre for Missing & Exploited Children. Child Pornography: Model Legislation & Global Review. `www.icmec.org/en_X1/pdf/Child-Pornography-Model-Law-English-7th-Edition-2012.pdf` (accessed March 4, 2016). 15, 16, 28

[51] Christian Jansohn, Adrian Ulges, and Thomas Breuel. Detecting Pornographic Video Content by Combining Image Features with Motion Information. In *ACM Intl. Conference on Multimedia (MM)*, pages 601–604, 2009. 25, 27, 31, 33

[52] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez". Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010. 26

[53] Wayne Kelly, Andrew Donnellan, and Derek Molloy. Screening for Objectionable Images: A Review of Skin Detection Techniques. In *IEEE Intl. Machine Vision and Image Processing Conference (IMVIP)*, pages 151–158, 2008. 28

[54] Vu Lam, Duy-Dinh Le, Sang Phan, Shin'ichi Satoh, and Duc Duong. NII-UIT at MediaEval 2014: Violent Scenes Detection Affect Task. In *MediaEval*, pages 1–2, 2014. 39, 107

[55] Vu Lam, Sang Phan, Duy-Dinh Le, Duc Duong, and Shin'ichi Satoh. Evaluation of multiple features for violent scenes detection. *Springer Multimedia Tools and Applications*, 1(1):1–25, 2016. 27, 38, 39, 40, 98, 107, 119, 127

[56] Ivan Laptev. On Space-Time Interest Points. *Springer Intl. Journal of Computer Vision (IJCV)*, 64(2):107–123, 2005. 25, 30, 46, 49, 66, 68, 72

[57] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 37, 61, 66, 75, 79

[58] Jiann-Shu Lee, Yung-Ming Kuo, Pau-Choo Chung, and E-Liang Chen. Naked image detection based on adaptive and extensible skin color model. *Elsevier Pattern Recognition (PR)*, 40(8):2261–2270, 2007. 28

[59] Ana Lopes, Sandra Avila, Anderson Peixoto, Rodrigo Oliveira, and Arnaldo Araújo. A Bag-of-Features Approach Based on Hue-SIFT Descriptor for Nude Detection. In *Springer European Signal Processing Conference (EUSIPCO)*, pages 1152–1156, 2009. 29, 33

[60] Ana Lopes, Sandra Avila, Anderson Peixoto, Rodrigo Oliveira, Marcelo Coelho, and Arnaldo Araújo. Nude detection in video using bag-of-visual-features. In *IEEE Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 224–231, 2009. 25, 27, 29, 33

[61] David Lowe. Object Recognition from Local Scale-Invariant Features. In *IEEE Intl. Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999. 25, 29, 143

[62] Alessandro Mecocci and Francesco Micheli. Real-time automatic detection of violent-acts by low-level colour visual cues. In *IEEE Intl. Conference on Image Processing (ICIP)*, pages I–345, 2007. 34, 36

[63] Ionuţ Mironică, Ionuţ Duţă, Bogdan Ionescu, and Nicu Sebe. A modified vector of locally aggregated descriptors approach for fast video classification. *Springer Multimedia Tools and Applications*, 1(1):1–28, 2015. 26, 37, 38, 40, 75, 80, 81, 107

[64] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. RECOD at MediaEval 2015: Affective Impact of Movies Task. In *MediaEval*, pages 1–2, 2015. 20

[65] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. From Many to One: Exploring Different Modalities for Sensitive Media Analysis. *In preparation*, 2016. 20, 87

[66] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Pornography Classification: The Hidden Clues in Video Space-Time. *Elsevier Forensic Science International (FSI), to appear*, 2016. 19, 20

[67] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Temporal Robust Features for Violence Detection. In *IEEE Intl. Workshop on Information Forensics and Security (WIFS), under revision*, 2016. 19, 20

[68] Mohamd Moustafa. Applying deep learning to classify pornographic images and videos. In *Pacific Rim Symposium on Image and Video Technology (PSIVT)*, pages 1–10, 2015. 32, 33

[69] Jeho Nam, Masoud Alghoniemy, and Ahmed Tewfik. Audio-visual content-based violent scene characterization. In *IEEE Intl. Conference on Image Processing (ICIP)*, pages 353–357, 1998. 35

[70] Eric Nowak, Frédéric Jurie, and Bill Triggs. Sampling Strategies for Bag-of-Features Image Classification. In *ACM European Conference on Computer Vision (ECCV)*, pages 490–503, 2006. 24, 26

[71] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher Kernel for Large-scale Image Classification. In *ACM European Conference on Computer Vision (ECCV)*, pages 143–156, 2010. 24, 26, 48, 65, 78, 106, 107, 143

[72] John Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *MIT Advances in Large Margin Classifiers*, 10(3):61–74, 1999. 95, 110

[73] Christian Platzer, Martin Stuetz, and Martina Lindorfer. Skin Sheriff: A Machine Learning Solution for Detecting Explicit Images. In *ACM Intl. Workshop on Security and Forensics in Communication Systems*, pages 45–56, 2014. 28

[74] Mateus Polastro and Pedro Eleuterio. NuDetective: A Forensic Tool to Help Combat Child Pornography Through Automatic Nudity Detection. In *IEEE Database and Expert Systems Applications (DEXA)*, pages 349–353, 2010. 31, 64, 68

[75] Rajat Raina, Yirong Shen, Andrew Mccallum, and Andrew Ng. Classification with Hybrid Generative/Discriminative Models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1–8, 2003. 94, 95

[76] Niall Rea, Gerard Lacey, Canice Lambe, and Rozenn Dahyot. Multimodal Periodicity Analysis for Illicit Content Detection in Videos. In *IEEE European Conference on Visual Media Production (CVMP)*, pages 106–114, 2006. 31

[77] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *IEEE Intl. Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011. 29

[78] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*, volume 1. Prentice Hall, 3 edition, 2010. 93, 94, 97

[79] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *Springer Intl. Journal of Computer Vision (IJCV)*, 105(3):222–245, 2013. 48, 65, 143

[80] Markus Schedl, Mats Sjöberg, Ionuţ Mironică, Bogdan Ionescu, Vu Lam, Yu-Gang Jiang, and Claire-Hélène Demarty. VSD2014: A Dataset for Violent Scenes Detection in Hollywood Movies and Web Videos. In *IEEE Intl. Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2015. 37, 75, 119, 120, 129

[81] Mary Short, Lora Black, Angela Smith, Chad Wetterneck, and Daryl Wells. A Review of Internet Pornography Use Research: Methodology and Content from the Past 10 Years. *Reuters Cyberpsychology, Behavior, and Social Networking*, 15(1):13–23, 2012. 16, 28, 30, 104, 129

[82] Josef Sivic and Andrew Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Springer Intl. Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477, 2003. 26

[83] Fillipe Souza, Guillermo Cámara-Chávez, Eduardo Valle, and Arnaldo Araújo. Violence Detection in Video Using Spatio-Temporal Features. In *IEEE Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 224–230, 2010. 25, 26, 36, 37, 40, 59

[84] Fillipe Souza, Eduardo Valle, Guillermo Cámara-Chávez, and Arnaldo Araújo. An Evaluation on Color Invariant Based Local Spatiotemporal Features for Action Recognition. In *IEEE Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 31–36, 2012. 25, 26, 30, 31, 33, 59, 61

[85] Chad Steel. The Mask-SIFT Cascading Classifier for Pornography Detection. In *IEEE World Congress on Internet Security (WorldCIS)*, pages 139–142, 2012. 26, 29, 33

[86] TapTap Software. Media Detective. `http://mediadetective.com/` (accessed March 4, 2016). 31, 64, 68

[87] Tinne Tuytelaars and Krystian Mikolajczyk. Local Invariant Feature Detectors: A Survey. *ACM Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008. 24

[88] Adrian Ulges, Christian Schulze, Damian Borth, and Armin Stahl. Pornography Detection in Video Benefits (a lot) from a Multi-modal Approach. In *ACM Intl. Workshop on Audio and Multimedia Methods for Large-scale Video Analysis (AMVA)*, pages 21–26, 2012. 31, 33

[89] Adrian Ulges and Armin Stahl. Automatic detection of child pornography using color visual words. In *IEEE Intl. Conference on Multimedia and Expo (ICME)*, pages 1–6, 2011. 25, 29, 33

[90] Eduardo Valle, Sandra Avila, Fillipe de Souza, Marcelo Coelho, and Arnaldo Araújo. Content-Based Filtering for Video Sharing Social Networks. In *Brazilian Symposium on Information and Computer System Security (SBSeg)*, pages 625–638, 2012. 25, 26, 27, 30, 31, 33, 59, 61

[91] Vladimir Vapnik. *Statistical Learning Theory*, volume 1. Wiley, 1998. 27, 93, 95, 97

[92] Andrea Vedaldi and Brian Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms. `http://www.vlfeat.org` (accessed May 22, 2016), 2008. 65, 78, 107

[93] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *Springer Intl.*

Journal of Computer Vision (IJCV), 103(1):60–79, 2013. 25, 46, 49, 61, 65, 66, 68, 72, 79, 127, 143

[94] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In *Springer European Conference on Computer Vision (ECCV)*, pages 650–663, 2008. 50, 51, 53, 54

[95] World Health Organization. WHA49.25 – Prevention of violence: a public health priority. `http://www.who.int/violence_injury_prevention/resources/publications/en/WHA4925_eng.pdf` (accessed March 4, 2016). 16, 34

[96] Chenggang Clarence Yan, Yizhi Liu, Hongtao Xie, Zhuhua Liao, and Jian Yin. Extracting Salient Region for Pornographic Image Detection. *Elsevier Journal of Visual Communication and Image Representation (JVCIR)*, 25(5):1130–1135, 2014. 26, 29, 33

[97] YangSky. PornSeer Pro. `http://www.yangsky.com/products/dshowseer/porndetection/PornSeePro` (accessed July 7, 2014). 31, 64, 68

[98] Wojtek Zajdel, Johannes Krijnders, Tjeerd Andringa, and Dariu Gavrila. Cassandra: audio-video sensor fusion for aggression detection. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 200–205, 2007. 34

[99] Bowen Zhang, Yun Yi, Hanli Wang, and Jian Yu. MIC-TJU at MediaEval Violent Scenes Detection (VSD) 2014. In *MediaEval*, pages 1–2, 2014. 27, 38, 39, 40, 98, 107, 119, 127

[100] Jing Zhang, Lei Sui, Li Zhuo, Zhenwei Li, and Yucong Yang. An approach of bag-of-words based on visual attention model for pornographic images recognition in compressed domain. *Elsevier Neurocomputing*, 110(1):145–152, 2013. 26, 29, 30, 33

[101] Qu Zhiyi, Liu Yanmin, Liu Ying, and Jiu Kang. A method for reciprocating motion detection in porn video based on motion features. In *IEEE Intl. Conference on Broadband Network & Multimedia Technology (ICBNMT)*, pages 183–187, 2009. 31

[102] Li Zhuo, Zhen Geng, Jing Zhang, and Xiaoguang Li. ORB feature based web pornographic image recognition. *Elsevier Neurocomputing*, 173(3):511–517, 2016. 29, 33

Appendix A

Effects of PCA Dimensionality Reduction upon TRoF-based Pornography Classification

As pointed out by Sánchez et al. [79], PCA dimensionality reduction is key to make the Fisher Vectors work as mid-level representation. Besides that, they report that classification accuracy does not seem to be overly sensitive to the exact number of PCA components, if enough dimensions (namely 64 to 128, in the case of 128-D SIFT [61] descriptions) are used.

In this vein, in this appendix, we verify if such behavior applies to the proposed TRoF-based video pornography classifier.

To help us choose candidate amounts of dimensionality reduction for a proper analysis, we express in Figure A.1 the normalized cumulative variance that is associated to each possible PCA dimensionality reduction — from one to 192 — over 192-D TRoF descriptions that were obtained from a set of 50 pornographic and 50 non-pornographic randomly chosen webvideos.

As one might observe, Figure A.1 depicts four options of PCA dimensionality reduction that we judge as of interest. The first one (a) is a reduction to the 96 TRoF dimensions that present highest variance (comprising 97% of the total TRoF data variance). This is the option that we use in all the experiments of the thesis; it represents a reduction of the low-level video description dimensionality by a factor of two, as recommended in [93, 71]. The second one (b) is a reduction to the 48 most variant TRoF dimensions, which comprise 90% of data variance. The third one (c), in turn, is a reduction to the 31 most variant dimensions (comprising 75% of variance), while the fourth (d) is a reduction to 17 dimensions (which comprises 50% of variance).

Table A.1 contains the classification accuracy that is obtained with the use of each chosen PCA dimensionality reduction within a TRoF-based pornography classifier. The same data fold was used in each configuration. As one might observe, the results are close to each other, suggesting that the findings of Sánchez et al. [79] are also applicable to TRoF: accuracy does not seem to be overly responsive to the exact number of PCA components, even if we maintain only 50% of description data variance (related to the largest reduction to only 17 dimensions).

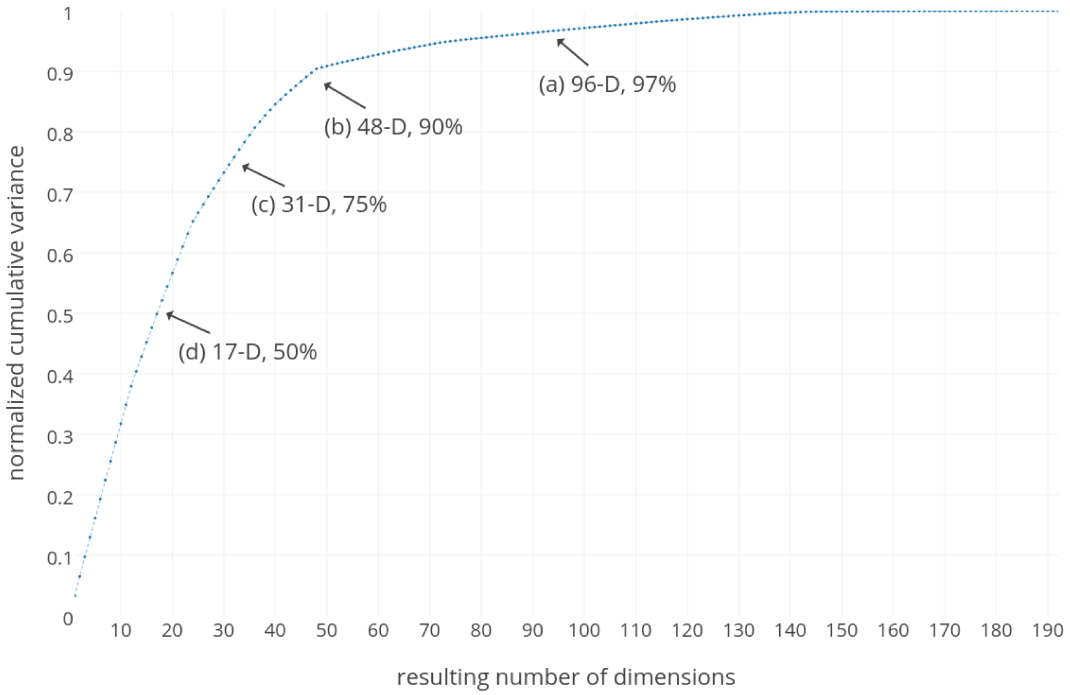


Figure A.1: Normalized cumulative variance of each possible PCA dimensionanlity reduction of TRoF descriptions. In (a), we have a reduction to 96 dimensions, which is the one that is employed in all the experiments of the thesis, and that comprises 97% of TRoF data variance. In (b), we have a reduction to 48 dimensions, which comprises 90% of TRoF data variance. In (c), we have a reduction to 31 dimensions, which comprises 75% of TRoF data variance. In (d), we have a reduction to 17 dimensions, which comprises 50% of TRoF data variance.

Table A.1: Normalized classification accuracy (ACC, cf. Equation 4.4) associated to each chosen PCA dimensionality reduction. The values were collected over the same data fold.

	dimensions (#)	ACC (%)
(a)	96	92.5
(b)	48	91.6
(c)	31	92.1
(d)	17	92.3

Last but not least, the present findings reveal an interesting performance issue: we can use smaller PCA-transformed TRoF descriptions within the proposed pipeline, if we want. That will probably lead to a faster classifier, with little impact on the classification accuracy. Nevertheless, in the main experiments of the thesis, we follow the literature and keep reducing the dimensionality of all low-level descriptions by half.