Universidade Estadual de Campinas
Instituto de Computação

Mauricio Lisboa Perez

Video pornography detection through deep learning techniques and motion information

Detecção de pornografia em vídeos através de técnicas de aprendizado profundo e informações de movimento

CAMPINAS
2016

Mauricio Lisboa Perez

Video pornography detection through deep learning techniques and motion information

Detecção de pornografia em vídeos através de técnicas de aprendizado profundo e informações de movimento

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre 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 Master in Computer Science.

Supervisor/Orientador: Prof. Dr. Anderson de Rezende Rocha
Co-supervisor/Coorientadora: Dra. Vanessa Testoni

Este exemplar corresponde à versão final da Dissertação defendida por Mauricio Lisboa Perez e orientada pelo Prof. Dr. Anderson de Rezende Rocha.

CAMPINAS
2016

Agência(s) de fomento e nº(s) de processo(s): FUNCAMP; CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

P415v

Perez, Mauricio Lisboa, 1989-
Video pornography detection through deep learning techniques and motion information / Mauricio Lisboa Perez. – Campinas, SP : [s.n.], 2016.

Orientador: Anderson de Rezende Rocha.
Coorientador: Vanessa Testoni.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Reconhecimento de padrões. 2. Visão por computador. 3. Redes neurais (Computação). 4. Fluxo óptico. 5. MPEG (Padrão de codificação de vídeo). I. Rocha, Anderson de Rezende,1980-. II. Testoni, Vanessa. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Detecção de pornografia em vídeos através de técnicas de aprendizado profundo e informações de movimento
Palavras-chave em inglês:
Pattern recognition
Computer vision
Neural networks (Computer science)
Optical flows
MPEG (Video coding standard)
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Anderson de Rezende Rocha [Orientador]
Ricardo da Silva Torres
Fernando José Von Zuben
Data de defesa: 09-06-2016
Programa de Pós-Graduação: Ciência da Computação

Universidade Estadual de Campinas
Instituto de Computação

Mauricio Lisboa Perez

Video pornography detection through deep learning techniques and motion information

Detecção de pornografia em vídeos através de técnicas de aprendizado profundo e informações de movimento

Banca Examinadora:

- Prof. Dr. Anderson de Rezende Rocha
  Instituto de Computação - Unicamp
- Prof. Dr. Ricardo da Silva Torres
  Instituto de Computação - Unicamp
- Prof. Dr. Fernando José Von Zuben
  Faculdade de Engenharia Elétrica e de Computação - Unicamp
- Dra. Fernanda Andaló (Suplente)
  Instituto de Computação - Unicamp
- Profa. Dra. Letícia Rittner (Suplente)
  Faculdade de Engenharia Elétrica e de Computação - Unicamp

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

Campinas, 09 de junho de 2016

Acknowledgements

A huge acknowledgement to my parents, for everything they allowed me to live, not only literally speaking, but also figuratively. Thanks for their unconditional love, constant support and unquestionable confidence in me. I'm also grateful to my brothers, whom, although not always close to me physically, had a great and positive influence on my formation. I also thank my family, which with its diversity, teaches me many things, and with its constant good mood, makes family gatherings so fun.

I also thank all my friends, for all their different life stories and opinions, which enrich me as a person, and also, more importantly, for making life so much fun and joyful, not only at the clearly unforgettable moments, but also during those small moments together. A special thanks to my oldest friends, those from high school, and to the not-so-old friends, from college.

I am grateful to my supervisor, for encouraging me to apply for the master's degree, even at the last day of application. Also for his support and teaching during the master's and the opportunities he provided me with, both in the masters and in the pursuit of a doctorate abroad. I cannot forget also to thank him for giving me the opportunity to have this experience with Science, which gave me the certainty of what I aspire for my life – as a researcher.

I also thank my co-supervisor and the other professors present at my research group, for their contributions to my work with ideas, suggestions and also teachings. A big thanks to my colleagues at the laboratory, in special those from my research group, for all their help, knowledge exchanges, general tips and for making the day-to-day life lighter.

Finally, an acknowledgement to Unicamp and the Institute of Computing, for the high quality master's program, to Samsung, for the scholarship, equipments and infrastructure provided, and to CAPES for the scholarship at the very end of the program.

Resumo

Com o crescimento exponencial de gravações em vídeos disponíveis online, a moderação manual de conteúdos sensíveis, e.g, pornografia, violência e multidões, se tornou impraticável, aumentando a necessidade de uma filtragem automatizada. Nesta linha, muitos trabalhos exploraram o problema de detecção de pornografia, usando abordagens que vão desde a detecção de pele e nudez, até o uso de características locais e sacola de palavras visuais. Contudo, essas técnicas sofrem com casos ambíguos (e.g., cenas em praia, luta livre), produzindo muitos falsos positivos. Isto está possivelmente relacionado com o fato de que essas abordagens estão desatualizadas, e de que poucos autores usaram a informação de movimento presente nos vídeos, que pode ser crucial para a desambiguação visual dos casos mencionados. Indo adiante para superar estas questões, neste trabalho, nós exploramos soluções de aprendizado em profundidade para o problema de detecção de pornografia em vídeos, levando em consideração tanto a informação estática, quanto a informação de movimento disponível em cada vídeo em questão. Quando combinamos as características estáticas e de movimento, o método proposto supera as soluções existentes na literatura. Apesar de as abordagens de aprendizado em profundidade, mais especificamente as Redes Neurais Convolucionais (RNC), terem alcançado resultados impressionantes em outros problemas de visão computacional, este método tão promissor ainda não foi explorado suficientemente no problema detecção de pornografia, principalmente no que tange à incorporação de informações de movimento presente no vídeo. Adicionalmente, propomos novas formas de combinar as informações estáticas e de movimento usando RNCs, que ainda não foram exploradas para detecção de pornografia, nem em outras tarefas de reconhecimento de ações. Mais especificamente, nós exploramos duas fontes distintas de informação de movimento: Campos de deslocamento de Fluxo Óptico, que tem sido tradicionalmente usados para classificação de vídeos; e Vetores de Movimento MPEG. Embora Vetores de Movimento já tenham sido utilizados pela literatura na tarefa de detecção de pornografia, neste trabalho nós os adaptamos, criando uma representação visual apropriada, antes de passá-los a uma rede neural convolucional para aprendizado e extração de características. Embora os experimentos mostrem que as duas fontes são equivalentes em termos de eficácia quando complementando a informação estática, a técnica de Vetores de Movimento MPEG é mais vantajosa devido a sua eficiência e também pois estar presente, por construção, nos vídeos, enquanto se decodifica os frames.

Nossa melhor abordagem proposta supera os métodos existentes na literatura em diferentes datasets. Para o dataset Pornography 800, o método consegue uma acurácia de classificação de $\sim 97,9\%$, uma redução do erro de 64,4% quando comparado com o estado da arte ($\sim 94,1\%$ de acurácia neste dataset). Quando consideramos o dataset Pornography 2k, mais desafiador, nosso melhor método consegue um acurácia de $\sim 96,4\%$, reduzindo o erro de classificação em 14,3% em comparação ao estado da arte ($\sim 95,8\%$).

Abstract

With the exponential growth of video footage available online, human manual moderation of sensitive scenes, e.g., pornography, violence and crowd, became infeasible, increasing the necessity for automated filtering. In this vein, a great number of works has explored the pornographic detection problem, using approaches ranging from skin and nudity detection, to local features and bag of visual words. Yet, these techniques suffer from some ambiguous cases (e.g., beach scenes, wrestling), producing too much false positives. This is possibly related to the fact that these approaches are somewhat outdated, and that few authors have used the motion information present in videos, which could be crucial for the visual disambiguation of these cases. Setting forth to overcome these issues, in this work, we explore deep learning solutions to the problem of pornography detection in videos, taking into account both the static and the motion information available for each questioned video. When incorporating the static and motion complementary features, the proposed method outperforms the existing solutions in the literature. Although Deep Learning approaches, more specifically Convolutional Neural Networks (CNNs), have achieved striking results on other vision-related problems, such promising methods are still not sufficiently explored in pornography detection while incorporating motion information. We also propose novel ways for combining the static and the motion information using CNNs, that have not been explored in pornography detection, nor in other action recognition tasks before. More specifically, we explore two distinct sources of motion information herein: Optical Flow displacement fields, which have been traditionally used for video classification; and MPEG Motion Vectors. Although Motion Vectors have already been used for pornography detection tasks in the literature, in this work, we adapt them, by finding an appropriate visual representation, before feeding a convolution neural network for feature learning and extraction. Although the experiments show that both sources are equivalent in terms of effectiveness when complementing the static information, the MPEG Motion Vectors technique is more advantageous due to its efficiency and also because it is present, by construction, in the video while decoding the frames. Our best approach outperforms existing methods in the literature when considering different datasets. For the Pornography 800 dataset, it yields a classification accuracy of ∼97.9%, an error reduction of 64.4% when compared to the state of the art (∼94.1% in this dataset). Finally, considering the more challenging Pornography 2k dataset, our best method yields a classification accuracy of ∼96.4%, reducing the classification error in 14.3% when compared to the state of the art (∼95.8% in the same dataset).

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Classification of sensitive media content (e.g., pornography, violence, crowd) has received increased attention recently because of its applications: it can be used for detecting, via surveillance cameras, inappropriate behavior; blocking undesired content from being uploaded to general purpose websites (e.g., social networks, online learning platforms, forums), or from being viewed in some environments (e.g., schools, workplaces); preventing children from accessing adult content on personal computers, smart-phones or television; and avoiding that improper content of someone is distributed through the phone network by sexting, among others.

Amongst the different kinds of sensitive media filtering, one of particular interest is pornography detection and, consequently, child pornography detection. Given its particular importance, this work focuses primarily on this problem, but the proposed methodology is potentially adaptable to other types of sensitive content as well. A common aspect regarding sensitive content is the subjectivity of its definitions. In this work, we choose to consider as pornography content "any sexually explicit material with the aim of sexual arousal or fantasy" [63].

A natural approach for pornography detection in the literature consists of first trying to detect nudity [23, 24, 26, 77] and then defining some appropriate thresholds to further filter the content. This type of solution commonly uses human skin features, such as color and texture, and human geometry [39, 59, 47, 10]. These methods normally use this information for modelling which pixel values and spatial distribution characterize a nude person. Although these methods are intuitive, they are somewhat naïve. Scenes of people in bath suits or sports, such as wrestling, will probably be misclassified, leading to a high rate of false positives and low classification accuracy. Therefore, nudity by itself, is not a reliable source of information for pornography detection. In addition, this kind of approach is also hard to implement, because of the many ad-hoc thresholds involved in the process, what also hinders generalization, especially when considering the diversity of ethnicities and different aspects of illumination, which may occur during the capture of the video.

In more recent years, other types of approaches have been explored in adult content filtering [19, 36, 72, 68, 5]. Inspired by the bag of words model from text classification, some authors have explored bags of visual words (BoVW) methods for pornography detection. They insert an intermediary description between the low-level features extracted from

the images, and the classification component. These methods normally involve choosing some low-level feature representation, normally gradient-like information, and creating a representative codebook. The involved steps are referred to as generating the codebook, coding the features and pooling the codewords count. In the end, a classifier will learn, through examples, which representations belong to the pornography class. These methods are more robust than the skin-based, having more classification power, but still suffer from some ambiguous cases. Choosing the codewords, the size of the codebook and which of the many coding and pooling strategies to use are crucial tasks for the good performance of the solutions.

Although thus far relatively underestimated for this problem, motion information available in videos would likely help to disambiguate the most difficult cases in pornography detection. Unfortunately, only a few works have explored spatio-temporal features or motion information in this problem until now [36, 6]. In these cases, the spatio-temporal feature evaluated was Space-Time Interest Points (STIP) [43] and the motion information coming from a statistical analysis of the MPEG-4 Motion Vectors.

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

Among the many machine learning techniques available, Deep Neural Networks, more specifically Convolutional Neural Networks (CNN), are showing groundbreaking results for image and video classification tasks [41, 64, 69, 51], such as detection of objects (e.g. airplane, bird, tennis ball) and action recognition (e.g. archery, golf, sit). In the last years, these networks have achieved state-of-the-art performance across different datasets and challenges, such as ImageNet[1] and UCF101 [66], outperforming, by a great margin, the previous approaches, including BoVW-based solutions.

Of particular attention, some authors have been studying how to adapt CNNs for human action recognition in videos, whereby the spatio-temporal information can be explored to improve the extracted features [45, 37, 40, 64]. Different architectures are possible, each one combining the spatial and temporal information in different ways, leading to better or worse features for the classification task. Some authors sought to extract the motion information implicitly by feeding a sequence of frames to the CNN [45, 37, 40], while others opted for explicitly feeding this information to the network through a previously computed Optical Flow Displacement Fields image representation [64].

The literature on pornography detection using deep learning techniques is very scarce. Moustafa [51] classified videos as porn or normal based on the majority voting from a sample of frames classified with off-the-shelf CNNs architectures. No spatio-temporal or motion information was applied in his method.

When targeting at sensitive media filtering, some interesting challenges appear for deep learning-based solutions: how to define an appropriate architecture; the possibility of reusing already trained architectures for related image categorization problems, thus

[1] http://www.image-net.org/

avoiding the need for huge amounts of training data; and how to incorporate time/motion information, which complements the spatial/static information.

1.1 Objectives

Our main objective in this work is to design and develop deep learning-based approaches to automatically extracting discriminative spatio-temporal characteristics for filtering sensitive content in videos, e.g., pornography. As far as we know, this is the first time convolutional neural networks, along with motion information, is applied for pornography detection in videos. Primarily, we explore static and motion information independently, and afterwards, investigate different ways for combining them. Also, we analyze whether or not it is possible to transfer learning from Convolutional Neural Networks trained previously for another problem (e.g., classification of objects with the ImageNet dataset), to our problem. Furthermore, distinct sources of motion information are explored, including novel forms of incorporating them (e.g., image representation of MPEG Motion Vectors).

1.2 Research Questions

The main research questions we address in this work are:

I) Are convolutional neural networks suitable for pornography detection in videos?

II) Is the motion information complementary to static ones?

III) If the static and motion information are complementary,

   i) Which is the most promising method for extracting the motion information?

   ii) How the fusion of such information should occur?

1.3 Contributions

Our main contributions in this work are:

I) A novel method for detecting pornographic videos, using convolutional neural networks along with static and motion information;

II) A new technique for exploring the motion information contained in the MPEG motion vectors [58];

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

1.4 Outline

We organized the remaining of this work as follows. Chapter 2 discusses existing approaches for dealing with the pornography detection problem. Chapter 3 presents a short summary of the necessary concepts to understanding this research proposal: Deep Learning; Motion Information; and Local Descriptors. Chapter 4 introduces the proposed research methodology. Chapter 5 presents the experimental setup along with the experiments and validation of the proposed methods and existing counterparts in the literature. Finally, Chapter 6 concludes the work and points out to some possible future research directions.

Chapter 2

State of the Art - Pornography Detection

In a relatively recent work, Short et al. [63] wrote a review of 46 articles that approached, to some extent, internet pornography. In their work, the authors highlight the importance of the authors to make explicit the definition of pornography used by them, since it has direct influence on the results and issues that can be encountered further on. The definition is also relevant for comparisons between works. As an example, some works consider the presence of genitals as being enough for classifying the content as pornography, while other authors argue that explicit sexual acts are necessary. It is proposed that a well formalized definition should contain the type of pornography and the reason that it is apparently expecting to motivate the viewers. The definition we adopted in this research was "any sexually explicit material with the aim of sexual arousal or fantasy", and was assembled by Short et al. [63], being a mixture of the definitions in the reviewed articles.

2.1 Skin-based techniques

Nudity detection using skin information has been extensively explored in the literature [23, 24, 25, 26, 77]. Fleck et al. [23] proposed a content-based retrieval strategy for returning images with naked people. It consisted of two main steps. First, filtering the images that have large areas of skin regions. To identify skin pixels, it is used thresholds on the intensity, hue and saturation value of each pixel. These areas are grouped and analyzed geometrically, validating if they could represent human limbs. The authors point out that the first phase is fragile to scale and saturation, and returns false positives from scenes with many people, or from materials with colors that are similar to human skin. The geometrical analysis suffers from missing limbs because of occlusion, close-ups or even by failure of the skin detector, among other reasons. These aspects lead to low precision and recall measures, when in comparison to newer methods we shall see later, more robust to these aspects.

Jones and Rehg [39] focused exclusively on the color information from the pixels, building some skin-based statistical models. A histogram of 256 bins for each channel is computed from the skin images, and another for the non-skin. These histograms model

the probability of the color belonging to a skin region. With a standard likelihood ratio approach, an RGB value can be labeled skin if above a certain threshold. A feature vector is then created comprising features that include the number of pixels detected as skin and the average confidence of the detected skin. A C4.5 decision tree classifier is used for the decision-making process.

2.2 Bag-of-Visual-Words techniques

The next milestone for the pornography detection problem was reached with the Bag-of-Visual-Words (BoVW) models. Deselaers, Pimenidis and Ney [19], aware that this type of solution had showed good results in many image classification problems, built a classifier for adult images using visual codebooks. Patches around interest points, with scaling and dimensionality reduction via Principal Component Analysis (PCA), were used as features. Codewords were selected through Gaussian mixture models, generating the codebook. The authors employed a hard-assignment coding policy followed by sum pooling. Other types of coding and pooling were proposed later on. The decision making considered Support Vector Machines (SVM) and Log-linear classifiers. The reported results showed that their method clearly outperforms the previous methods, mainly based on color features. In addition, the authors show that little performance is gained by combining this solution with skin-based ones.

2.3 Classifying Videos

When it comes to video, the basic approach considers extracting the frames and applying an image-based description and classification approach. However, these methods disregard valuable information that videos provide, the concept of motion. Although not directly in the field of pornography detection, the importance of temporal information for action recognition has been assessed for many years now. Dollar et al. [21] proposed a corner detector algorithm similar to the Harris detector [29], that seeks for "corners" in time. The detected "motion" corners are then described with cuboids around them. With the help of a codebook from the cuboids, the histograms of features from the short scenes demonstrated much greater classification power than the spatial-based descriptors. Some recent works keep the trend of exploring motion information for action recognition as Laptev et al. [44], Wang et al. [73], and Simonyan and Zisserman [64].

Turning our attention to pornography detection, Jansohn et al. [36] were one of the first authors to explore the time information while detecting pornography. They used a statistical analysis of MPEG-4 motion vectors, with a bag-of-visual-words similar to the one proposed by Deselaers, Pimenidis and Ney [19]. Different ways of combining the motion vector information in overlapping windows of time were experimented. A description of the video was generated by pooling these windows, generating a motion histogram (MHIST). The decision making in the end considers an SVM classifier. The classifier using the time information alone gave effective results and was improved upon when combined with a BoVW-based (spatial characterization) approach.

Avila et al. [5] proposed an extension of the bag-of-visual-words approach for the pornographic video detection task. The improved design involved new pooling and coding formalisms for the local descriptors. Instead of simply summing up the activations in the pooling step, as in Deselaers, Pimenidis and Ney [19], an estimation on the distribution of the descriptors distance to the codewords is used. For the new coding, a semi-soft scheme was used, on which different softness parameters, based on the variation of each cluster, are applied. The decision-making process considers an SVM classifier. Different datasets were used to validate the extension including a pornographic benchmark available online[1].

Another supplementary information provided by video, that can be used for pornography detection, comes from audio. Rea et al. [57] proposed an audio feature extraction approach for this problem, that consists of analyzing the periodicity from the sound. The inspiration comes from the fact that this type of content usually has repetitive sounds. To capture and measure the periodicity, an autocorrelation of the energy filter is applied to the audio signal, and the area between the local maxima's and minima's curves is computed. If the area is above a certain threshold, that configures a repetitive sound, suggesting it comes from pornography. In an evaluation with diverse audio samples, not from pornographic content, a false alarm rate of 2% was reported. But the authors highlight that this approach alone is not robust to other periodic sounds, such as in a tennis match, so visual features should also be present to remove ambiguities.

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

2.4 Convolutional Neural Networks

Although Deep Learning has been responsible for most of the current breakthroughs in image classification tasks, few explorations were made for these techniques within the context of pornography detection in video. Moustafa [51] performed a superficial adaptation of well-known CNN architectures for image classification to the pornographic video classification task. He used AlexNet [41] and GoogLeNet [69] architectures directly on selected frames for classifying them in porn or non-porn. Afterwards, the author performs a majority voting with these frames, to classify the whole video. The author used the weights learned from Imagenet dataset, fine-tuning only the last layer, which corresponds to the classifier. Within this approach, no motion information was explored.

2.5 Third-party Solutions

The nudity and pornography detection problem has not been tackled only by the academia. There exists some software, mostly commercial, aiming at solving this problem. Some focus on blocking websites that contain this type of content (e.g., CyberPatrol, CYBERsitter, NetNanny, K9 Web Protection, Profil Parental Filter). Others scan the hard drive in search of pornography (e.g., SurfRecon, Porn Detection Stick, PornSeer Pro). There is

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

even a Brazilian software, called NuDetective, developed by the Brazilian Federal Police, that focus on detecting child pornography. These commercial solutions are mainly based on skin detection approaches, and none explored the space-time nature of videos for aiding the detection of pornography. Therefore, they are behind the current state of the art for this problem.

2.6 Summary Table

Table 2.1 presents an overview of the many related work on the pornography detection problem and its sub-problems, skin and nudity detection.

Table 2.1: Summary of approaches on skin, nudity or pornography detection. For the **Type** column, we have Nude Image (NI), Nude Video (NV), Porn Image (PI) and Porn Video (PV).

Reference	**Type**	**Method**	**Classifier**
Fleck et al. [23]	NI	Skin detection; geometrical analysis	Threshold
Lopes et al. [49]	NI	BoVW model; PCA on SIFT and HueSIFT descriptors	Linear SVM
Lopes et al. [48]	NV	BoVW model; PCA on SIFT and HueSIFT descriptors; voting scheme	Linear SVM
Jones and Rehg [39]	PI	Skin color histogram; color probabilities	C4.5 decision tree
Rowley et al. [59]	PI	Skin color histogram; skin texture histogram; face detection	RBF SVM
Zheng et al. [77]	PI	Skin color detection; skin region detection; shape descriptors	AdaBoost with C4.5 decision tree
Zuo et al. [79]	PI	Patch-based skin color detection; human body part detection	Random forest
Deselaers et al. [19]	PI	BoVW model; PCA on SIFT descriptors; GMM model	SVM; histogram intersection kernel
Ulges and Stahl [72]	PI	BoVW model; DCT in YUV color space	SVM; χ^2 kernel
Steel [68]	PI	Mask-SIFT; skin percentage	Cascade classifier of three stages
Zaidan et al. [75]	PI	Bayesian method with a grouping histogram; segmentation with back-propagation neural network	Artificial neural network
Zhuo et al. [78]	PI	ORB descriptors; BoVW model	SVM
Nian et al. [52]	PI	CNN architecture CaffeNet	CNN Softmax
Jansohn et al. [36]	PV	BoVW model; DCT in YUV color space; motion histograms	SVM; late fusion
Avila et al. [5]*	PV	BoVW-based model: BossaNova; HueSIFT descriptors	RBF SVM
Caetano et al. [13]*	PV	BoVW-based model: BossaNova; binary descriptors	RBF SVM
Caetano et al. [14]*	PV	BoVW-based model: BossaNova; binary descriptors; multiple aggregation functions	RBF SVM
Valle et al. [6]*	PV	BoVW model; STIP descriptors	Linear SVM
Rea et al. [57]	PV	Skin color estimation; MPEG motion information; periodic patterns detection	Threshold over periodicity measure
Ulges et al. [71]	PV	BoVW model; DCT in YUV color space; MFCC audio features; motion histograms; skin detection	SVM; RBF and χ^2 kernels; late fusion
Moustafa [51]*	PV	CNNs on raw frames	Majority Voting

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

Chapter 3

Related Concepts

In this chapter, we cover the main concepts necessary for understanding this manuscript. Since Convolutional Neural Networks are the core of our proposed methodology, Section 3.1 details Deep Learning concepts, such as basic notions of Convolutional Neural Networks (ConvNets), filter optimization, the GoogLeNet architecture, we rely upon this research, and the motion/temporal networks. Section 3.2 brings an explanation of the motion information sources explored herein, Optical Flow and MPEG Motion Vectors. Finally, Section 3.3 presents the techniques associated with local description methods that we used for comparison with our proposed methodology.

3.1 Deep Learning

The main reason for the success of the deep neural networks (DNNs) lies on their ability to generate features directly from raw data [9] (most notably the intensity values of the pixels). These extracted features help overcoming the gap between the bits representing an image to its semantic value, such as objects and scenes. With this new level of representation as input, supervised learning systems, e.g., another Neural Network or Support Vector Machines (SVM) [34], can train with this enhanced data, generating highly discriminative models.

In this section, we discuss the: basic notions of the deep architecture explored in our proposed methodology, Convolutional Neural Networks; optimization methods; design of GoogLeNet [69]; and temporal approaches with ConvNets.

3.1.1 Convolutional Neural Networks

ConvNets have been designed aiming to emulate a powerful "vision" system, which is the visual cortex [46]. This biological inspiration guided the architecture decisions that are explained here.

Spatial correlation, that is the association of nearby information, is an important step for vision processing [35]. ConvNets enforce this kind of correlation by sparsely connecting the neurons on adjacent layers. That means each neuron will be connected to just a sampling of the neurons of the layer below, where these neurons' output are related to regions of the signal that were somehow close to each other in the original input,

e.g., nearby pixels in an image. Figure 3.1 depicts a visual representation of this sparse connectivity of neurons. This way, at each layer, we aggregate information from smaller subregions, which allows the network to generate a more accurate information from a larger subregion than the previous layer. The sub-regions that a neuron has as input is often called its receptive field. Thus, the stack of convolutional layers go from local information, beginning with pixel data such as colors and brightness, to broader concepts, including borders and corners. The key-point of success is that, through learning, these concepts naturally evolve from corners, to high-level notions, such as eyes and noses [76].

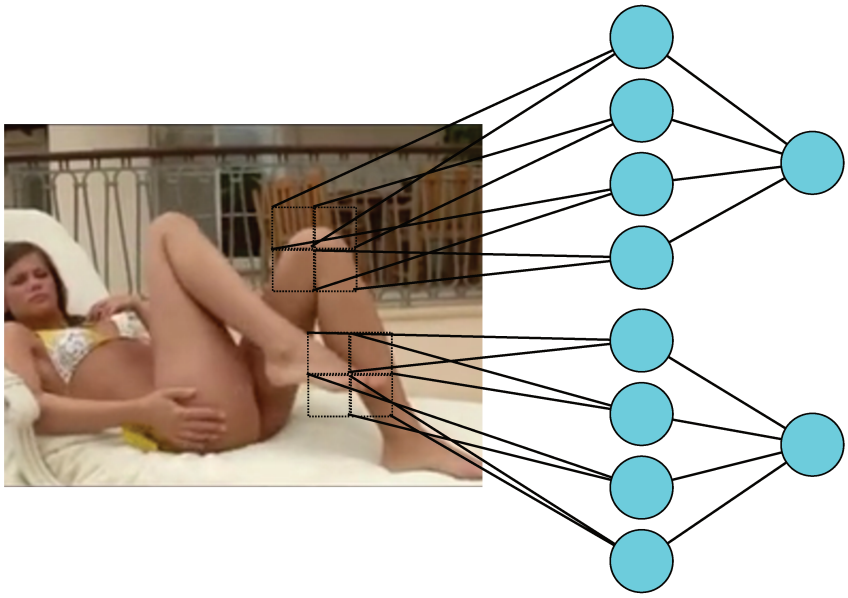


Figure 3.1: Sparse connectivity of the neurons present in convolutional neural network architectures, for enforcing spatial correlation.

These concepts may appear in different locations on an image, and possibly repetitively. Therefore, the same type of information should be filtered on these different locations. This necessity led to the next important architecture detail of ConvNets, which is parameter sharing among a subset of the neurons in the same layer. The collection of parameters shared by a subset of neurons define a feature map. Due to the sparse connectivity explained above, each of the neurons of this subset has a distinct combination of inputs at the receptive field, which means that they analyze different sub-regions of the signal, but in search of the same content because of the common parameters. Figure 3.2 depicts the weight sharing present at neurons with the same feature map.

The feature maps mentioned above are, in fact, a set of weights, composing a filter bank. Filter banks are applied to the input signal through a mathematical operation denominated convolution, hence the name of this network architecture. For learning the filter weights, we normally apply gradient descent [60]. Filter optimization is a very delicate and determinant step for the performance of the network, hence we will detail it in the next section.

Normally, after linear filtering the input with the filter bank through convolution, the output is added to a bias term and fed to a non-linear activation function, usually a *Rectified Linear Unit (ReLU)*. Subsequently to applying the feature map, some sub-

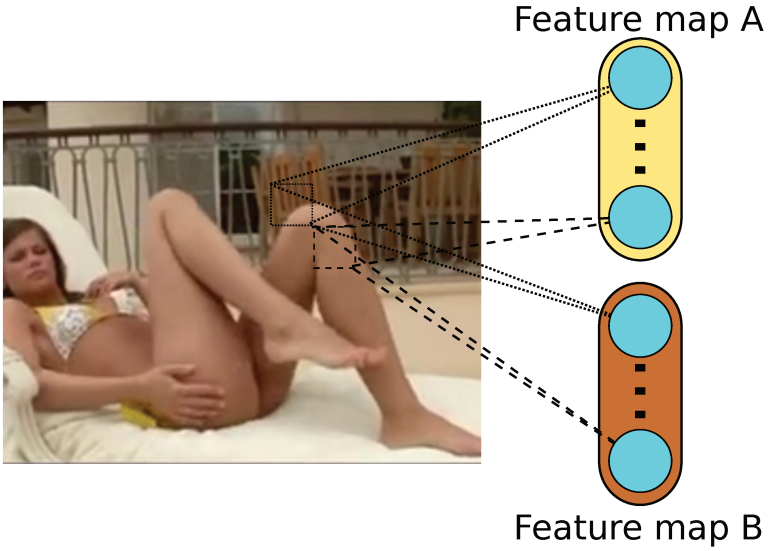


Figure 3.2: Sharing of weights among a subset of the neurons in the convolutional neural network architecture, enabling the repeated application of the same feature map in all regions of the input image.

regions may be grouped by max-pooling their outputs, before being fed to the upper layer. This downsampling step is optional, but it helps reducing the dimensionality and allows the composition of spatial invariance information.

Hence, deep convolutional neural networks consist of basically stacking up the previously described components, commonly with more than twenty layers, and some fully-connected layers plus a softmax layer for classification. In a nutshell, the basic convolutional network structural parameters can be summarized to: 1) number of layers; 2) number of filters in each layer; 3) size and stride of the receptive fields; 4) type of the non-linear function; 5) size and stride of the pooling.

3.1.2 Filter Optimization

An appropriate architecture from a convolutional neural net is fundamental for effectively extracting the best features from raw data. But if used with random weights for its filters, they could still be far from the optimal performance. For learning filters more suitable to a certain problem, we normally employ a Stochastic Gradient Descent technique, using backpropagation for computing the gradients [60]. In summary, the learning consists of optimizing an objective function, which is related to the network weights, by analyzing the derivatives (or gradient) of this function at each layer, backpropagating the gradients from the top layer (prediction output) to the bottom layer (data input). From its respective gradient information, the network weights undergo small adjustments, aiming to optimize the objective function.

Filter optimization, specially on deep networks (e.g., from twenty to more layers), may have an undesired side-effect, also common in other machine learning approaches, known as overfitting. Overfitting is associated with the parameters optimized during training becoming heavily specialized to the training data, but lacking generalization for dealing with the unseen samples while testing later on, leading to a high performance in

training, but poor performance during testing. This anomaly happens frequently when using DNNs, mainly because they contain a huge number of parameters for modeling the system to the samples fed during the learning process. Fortunately there are some countermeasures to avoid, or at least reduce, overfitting.

A natural approach for combating overfitting is to employ more data during training. Unfortunately, it is not always possible, or feasible, to acquire more samples to a given problem. As a workaround, what we commonly do is to synthetically generate more samples from the existing data. This method, named data augmentation, has been applied in many of the latest works [41, 69, 64]; and Chatfield et al. [17], for instance, evaluated different data augmentation techniques. These tricks consist of basic manipulations on the images to increase the number of input samples, such as resizing, cropping different regions, flipping and color jittering.

Another method, besides data augmentation, widely used with neural networks to overcome overfitting is Hinton et al.'s [30] technique called "Dropout". This technique consists of randomly dropping the output of some neurons on each training iteration, preventing them to contribute to the forward pass and, consequently, not participating in backpropagation. Thereby, the active neurons will have to compensate for the absence of the other neurons, and improve the robustness of their learned features. The co-adaptation of the neurons is therefore reduced. Neurons with more robust features to the presence or not of other neurons, also have more robust features for the data, increasing the generalization of the network.

Although data augmentation and dropout techniques have been developed for a scenario of insufficient training data, they can still be applied anyhow, for improving even more the performance of a given classification system.

As we have explained in this subsection, filter optimization is a tricky and important step for achieving a more satisfactory performance during designing and development of a deep convolutional neural network. Luckily, we can use the filters optimized from a previous training, as a starting point for learning with new training data, even if it is from a different, but related, problem. This technique is referred to as "fine-tuning" in the literature. Therefore, through fine-tuning, we can transfer learning from another dataset, or problem, that has many samples, to our task, diminishing the need for a high number of data samples on the latter. This normally leads to a better performance than the one achieved when training from scratch, using only the data of our problem.

In the methodology we propose herein, in Section 4, we employ directly and indirectly these recently explained filter optimization concepts. We directly apply fine-tuning and dropout by training the GoogLeNet [69] architecture from the weights learned from Imagenet 2014 on over one million images [61]. The GoogLeNet [69] model, in its turn, was trained using data augmentation and dropout techniques, thus their indirect use.

3.1.3 GoogLeNet

Different models of deep neural networks have been proposed and evaluated over the past years. But it was only in the 2012 edition of the Imagenet Challenge [61], with Krizhevsky et al.'s contribution [41], that the research community really turned its attention to Deep

Learning. The Convolutional Neural Network they built for the competition ranked first, with a top-5 error rate of 15.30%, against 26.20% of the second-best, an unprecedented improvement in image classification problems. Since then, there has been great improvements in the CNNs architecture and training techniques. In the 2014 edition of ImageNet, Szegedy et al. [69] designed GoogLeNet, a CNN which achieved 6.67% in the top-5 error for the competition, a classification accuracy improvement of over 55% over Krizhevsky et al.'s results

Architecture

The main components of the GoogLeNet [69] architecture are the inception modules. This module performs 1×1, 3×3 and 5×5 convolutions plus a 3×3 max pooling in the same convolutional layer, and on top of the same input, concatenating all the outputs into a single vector afterwards. Due to computational requirements, filter banks of 1×1 are applied before the 3×3 and 5×5 convolutions, and after the 3×3 max pooling, for dimensionality reduction of the input before these more expensive operations. Figure 3.3 depicts one inception module.

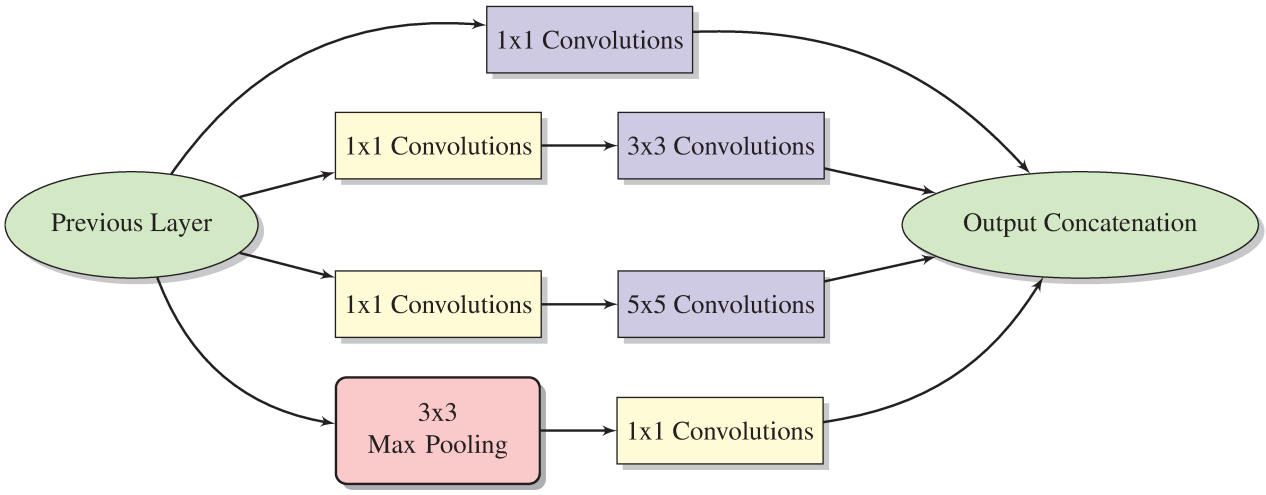


Figure 3.3: Representation of the Inception module present in the GoogLeNet architecture. Image based on Szegedy et al. [69].

Thus, GoogLeNet [69] is assembled as a stack of the inception modules, along with commonly found layers of convolutional neural networks, such as regular convolutions, max-pooling, fully-connected, dropout and softmax. Rectified linear activation is used on all convolutions, reduction and pooling projection layers. In total, it contains 27 layers (if counting the pooling layers). The input of the network is an RGB image of dimensions 224×224 pixels.

The network also contains two auxiliary classifiers besides the final one. These classifiers were inserted in intermediate layers as a countermeasure against the vanishing gradient problem [31], which occurred because of the elevated depth of the network. The extra classifiers are appended on top of selected inception modules, and take shape in the

form of smaller CNNs. They are structured with the following components and in this order: 1) Average pooling layer; 2) Reduction layer of 1×1 Convolution; 3) Fully-connected Layer; 4) Dropout layer; 5) Softmax as the classifier. Their losses are added to the final classifier loss, but with a discount weight of 0.3 each.

Figure 3.4 depicts the whole GoogLeNet architecture. For more details on why the authors have made these architecture decisions, we encourage the reader to refer to Szegedy et al.'s work [69].

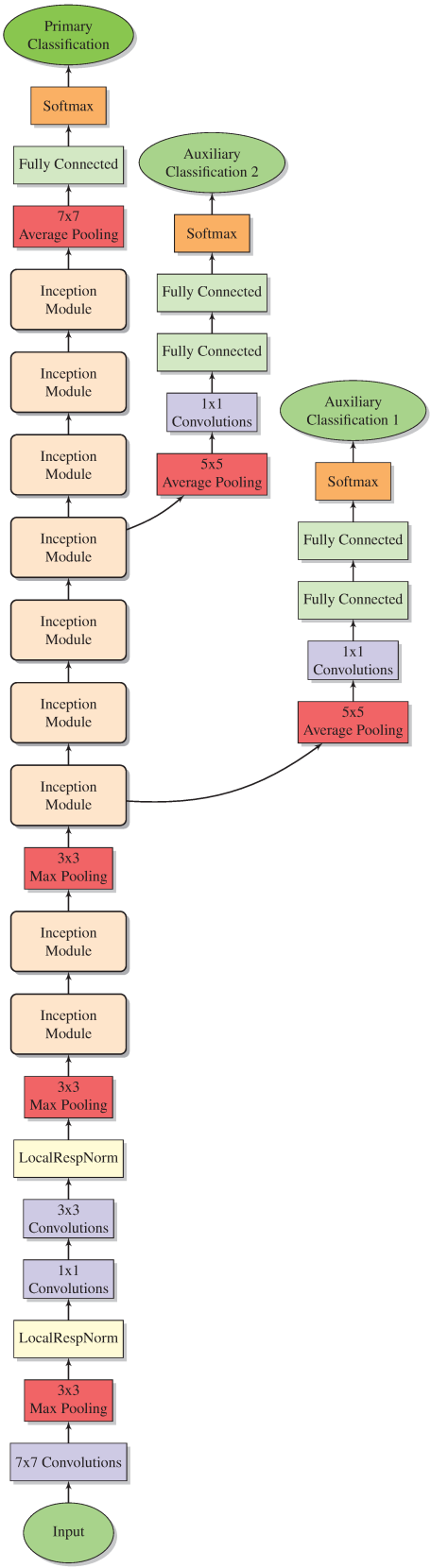


Figure 3.4: Representation of the GoogLeNet architecture. Image based on Szegedy et al. [69].

Training Methodology

For the Imagenet Challenge [61], the authors trained GoogLeNet [69] using an asynchronous stochastic gradient descent algorithm. A momentum of 0.9, along with a step learning decay of 4% at each 8-epoch interval, was used. Although it has been applied some image sampling methods, the authors explain that there were not a unified pipeline for all models generated for the Imagenet Challenge [61]. But they have later validated a prescription with good performance that contains: distinct image crops with size ranging from 8% to 100% of the original area, while maintaining an aspect ratio constrain of [3/4,4/3]. Also, photometric distortions [33] helped combating overfit.

Under Caffe[1] deep learning framework, there is a distribution for the GoogLeNet [69] architecture and an ImageNet model. The model present within this distribution has been trained with just some of the data-augmentation techniques, including cropping from different regions and flipping, but not others such as relighting, scaling and varying aspect-ratio. It used a polynomial learning rate decay policy, instead of the step policy, because it allowed a much faster training, about 4×. It would be infeasible to train this network from scratch using the original configurations within a viable time, given the available computer sources at our disposal. Also, given that these weights are fine-tuned later on for our problem of interest herein, this should not have a major impact on performance. Therefore we opted for using the Caffe model.

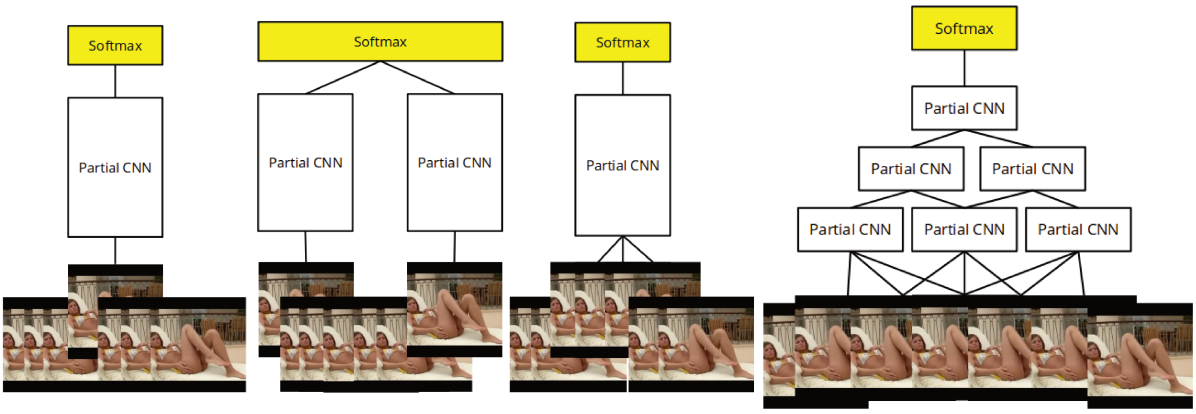
3.1.4 Motion/Temporal Networks

The previous architectures described here were designed targeting at image classification. Although they can be used for video classification in a frame-wise approach, the temporal information will be almost completely discarded. As stated before, this type of information should not only increase the performance, but it might also be indispensable for removing the ambiguity on the pornography classification problem, for instance. As a matter of fact, however, only a few authors have addressed video classification with Convolutional Networks [40, 64] thus far offering us a whole new venue for possible original contributions.

Karpathy et al. [40] explored a variety of architectures to implicitly capture temporal information between a sequence of frames. First of all, they set as baseline a single frame approach, where the network model is learned from pixels in the raw frames. The classification is done independently for each frame. The following architectures strive to add time information, with three different types of fusion: Late Fusion, Early Fusion and Slow Fusion. Figure 3.5 depicts Karpathy approaches. In the first, two frames, with fifteen frames in between, are fed to two single-frame networks, with outputs being combined only at the top layers. With the Early type, a set of frames contained in a temporal window is fed to the first convolutional layer, where modified filters have an extra dimension on its receptive fields, related to some temporal extent, which is essentially the number of frames. The Slow fusion is the more complex, mixing the previous two schemes. Small and overlapping temporal windows are defined inside a larger temporal window, whereby each sub-set is then fed to separated paths in the network, that will be progressively combined

[1] `http://caffe.berkeleyvision.org/`

in each of the layers above. The reported results exhibited small performance variance between the fusion approaches and also in comparison with the single-frame network, with the Slow architecture performing slightly better. These initial results indicate that Convolutional Networks have some troubles to implicitly capture the motion information from sequential frames.



(a) Single Frame (b) Late Fusion (c) Early Fusion (d) Slow Fusion

Figure 3.5: Karpathy et al. [40] designed architectures to incorporate motion information in a Convolutional Neural Network.

Simonyan and Zisserman [64] proposed a Two-Stream ConvNet, that uses optical flow to supply complementary information to the classification. Inspired by the biological aspect of human vision, they designed an architecture related to the two-stream hypothesis, in which the visual cortex separately recognizes objects and motion [27]. This is accomplished by having an architecture with two pathways, one for the frames and another for the motion information, that is later combined by score averaging (Figure 3.6). For the motion information, the authors used stacked optical flow displacement fields. This stacking comprises the image representations of the vertical and horizontal components, from the displacement vector field, of an arbitrary number of consecutive frames. This approach achieved great improvement on previous state-of-the-art deep neural nets on action recognition datasets such as the UCF-101 [66] and HMDB-51 [42]. In contrast to Karpathy et al. [40], Simonyan and Zisserman explicitly provided the motion information to the network, through optical flow image representation, and this is probably the reason for the performance improvement, leading to a more promising approach.

3.2 Motion Information

As stated earlier, the motion information contained in a sequence of images is very valuable for tasks of video classification. We now review two sources for this type of information that have been used over the past years: Optical Flow and MPEG Motion Vectors.

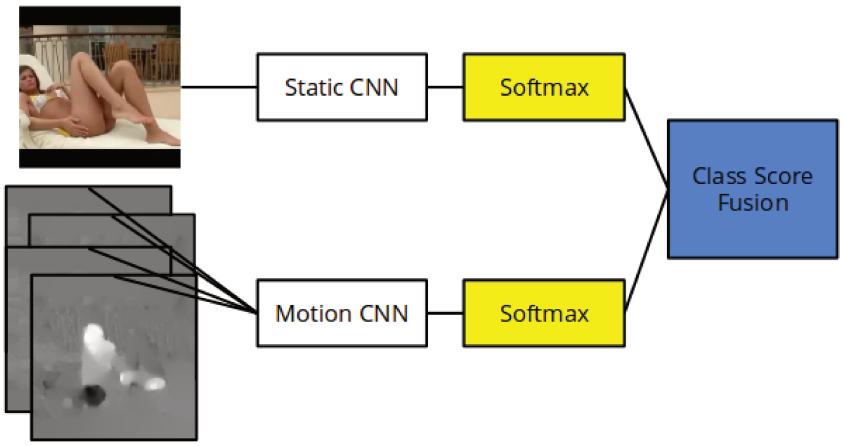


Figure 3.6: Simonyan and Zisserman [64] proposed a two-stream CNN architecture, designed to incorporate explicit motion information in a convolutional neural network, and combine it with static information through score fusion.

3.2.1 Optical Flow

Optical flow comes from the problem of estimating motion for each pixel between images from the same scene [32]. This estimation helps understanding the movement, from the perspective of the viewer, of the objects contained in the scene. Although this is an old problem in Computer Vision, there are recent works improving optical flow computing [11]. This task is intrinsically related to the image registration problem. As a result, most approaches intend to match pixels present in both images, by analyzing their neighborhood, and inserting some mechanisms to increase robustness to brightness changes and noise, such as gradient constancy assumption. With the pixels matched, an estimation on the gradient of the movement can be generated.

The final output for the different optical flow computation methods is a displacement field, which contains in each pixel, the relative movement for it between the given images. Each position in this field has a displacement vector indicating the estimation of which direction the respective pixel has moved to and the intensity (gradient) of this movement. Figure 3.7 depicts an example of the output. Altogether, these vectors provide us with a relevant estimative from the motion of the objects in the scene.

3.2.2 MPEG Motion Vectors

Another source of motion information frequently used for video classification [57, 36] is the motion vectors contained in the MPEG codification of the analyzed media. Differently from the optical flow, this motion estimation was not originally created to allow an understanding of the movement in the scene; instead, it was designed for aiding the compression of the video [58].

During video compressing, a frame can be decomposed into residual and predicted parts. Actually, what happens is that a predicted frame is generated from a reference frame, through a series of transformations and complementary information, so the difference from this prediction with the current frame results in the residual part. During

Figure 3.7: Sequential Raw Frames (left and middle) and the respective Optical Flow Displacement Field (right) computed from them. The regions with more movement in the raw frames (e.g., hand, knee and foot) are also the regions with the greatest displacement vectors in the field.

decompressing, for reconstructing the current frame, it is only needed the information about the reference frame, the necessary transformations and the residual part.

Among the transformations and complementary information used for retrieving the predict frame, there is a method named Motion Estimation and Compensation [58], which in one of its steps maps the pixels movements between the current and the reference frame. This information is what is called *Motion Vectors*. Motion estimation in video compression is performed in a block-based fashion, where pixels are grouped in macroblocks in order to reduce computational complexity. Macroblocks are rectangular regions of $M \times N$ pixels, usually with M and N as 8 or 16. Motion vectors are then computed per macroblock and contain the following information: the position (x,y) of the macroblock in the current frame; its position (x',y') in the reference frame; and the size of the macroblock $(M \times N)$. Figure 3.8 illustrates an example. Note that this mapping only occurs during compression, hence, when the video is decompressed for analysis, this information is readily available. The gathering of the motion vectors, from a frame being reconstructed, gives us useful information about the motion that has occurred at that time.

Although we referenciate here to the motion vectors from the MPEG codec, and this codec is one of the most used today, this source of information is commonly present in other codecs as well, such as Google's VP9 [28].

3.3 Local Descriptors

Most of the existing approaches to pornography detection uses local description methods, which depend on low-level extraction of features. These methods involve engineered methods for detection of points of interest in the image or video, and also for description of the raw data in that point and neighborhood. Some of these methods were designed only for images, so, although they can be applied framewise, they do not take advantage of any time information contained in the video, therefore they are considered *static*. In contrast to the *static* methods, some other approaches are *time-aware*, as they analyze the frames not only considering their spatial information, but also the information from

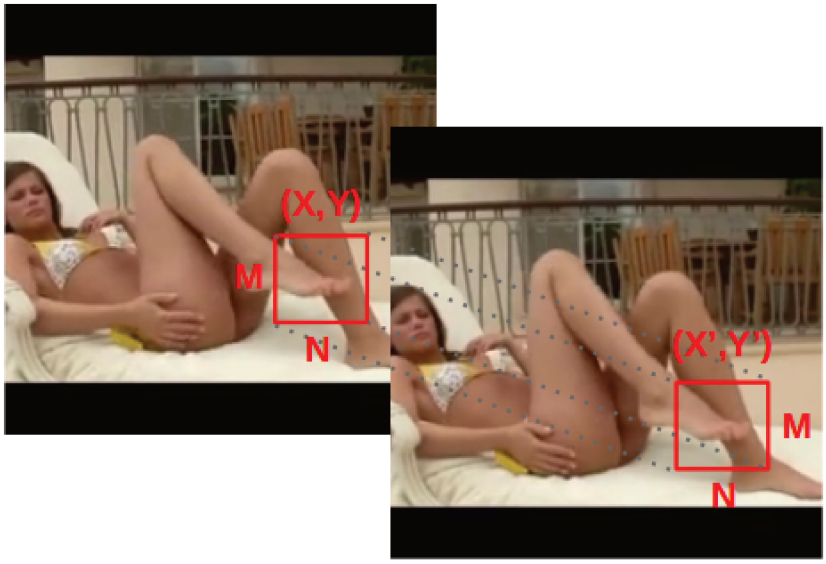


Figure 3.8: Example of a macroblock and its respective Motion Vector between the current frame (left) and the reference frame (right).

the frames that are temporally neighbors.

In most of the cases, these low-level descriptions are still semantically far from abstract concepts such as pornography, and an intermediary step before the decision making is indispensable for achieving a satisfactory performance. Hence, a mid-level representation is commonly used.

In this section, we cover the following traditional low-level local descriptors: Speeded Up Robust Features (SURF) [8], Space-Temporal Interest Points (STIP) [43] and Dense Trajectories (DTRACK) [73]. The first one is *static* and the latter two are *time-aware*. We also give details on one of the best mid-level representation methods, Fisher Vectors [55]. All of these techniques can be used for the pornography detection problem without difficulty.

3.3.1 Speeded Up Robust Features (SURF)

Bay et al. [8], while developing SURF, aimed at designing a fast and efficient interest point detector and descriptor, that was invariant to scale and rotation. For detection, we apply the Hessian matrix using its determinant as measure for choosing both the location and the scale. The descriptor is generated by employing Haar-wavelet filters on the neighborhood of the interest point and describing the distribution of responses.

SURF uses concepts similar to the Scale Invariant Feature Transformation (SIFT) [50], but reducing its complexity, achieving a much faster method. With the purpose of accelerating the technique, integral images and box-type convolution filters are employed, allowing a more efficient computation of approximate differential operators.

3.3.2 Space-Temporal Interest Points (STIP)

Laptev [43] furthered extended upon the Harris [29] corner detector to encompass a third dimension — the time — to the equations. The main idea behind this extension is to identify in a frame, regions with high spatial variance, that along the consecutive frames, have moved to other locations, indicating high temporal variance. With this modification, it was possible to detect spatio-temporal interest points, corresponding to interesting events in a video timeline that may be discriminative for identifying more abstract concepts which depend on motion information.

Besides the spatio-temporal interest point detector, it is also necessary to employ a description method that also incorporates the information from the time dimension. For this purpose, Laptev [44] suggests using Histograms of Oriented Gradients and Histograms of Optical Flow (a.k.a., HOG-HOF descriptions) for the description of three-dimensional video patches, distributed along the neighborhood of the detected interest points.

3.3.3 Dense Trajectories (DTRACK)

Wang et al. [73] explored a different approach than Laptev et al. [43] for the detection of spatio-temporal interest points. Instead of defining a corner detector with three dimensions, Wang et al. densely sample feature points from a starting frame, within different spatial scales, and track their trajectory through optical flow on the frames ahead, and their respective scales. These trajectories may be pruned if they are static, since the main interest is in motion points, or if they have large and sudden displacements, which may characterize an error during the tracking.

For each remaining feature point after pruning, a series of descriptions is generated along the spatio-temporal neighborhood of its trajectory. These descriptions are performed with HOG-HOF descriptors [44], combined to Motion Boundary Histograms (MBH) [18]. For description, the spatio-temporal neighborhood around the trajectory is subdivided by a grid, for maintaining structure information on the final representation.

3.3.4 Fisher Vectors

The previously explained low-level features extract useful information from the raw data contained in videos/frames. Still, they do not have enough semantic value for discriminating more abstract concepts, requiring a mid-level representation. Perronnin et al. [55] introduced the best mid-level representation currently available, the Fisher Vector.

This was achieved by extending the Bag-of-Words (BoW) method [65]. BoW methods involve selecting some of the low-level features as visual codewords and assembling a codebook. The remaining features are then encoded into this codebook, and their distribution into these codewords is used as a new description for the original data, by generating a histogram from it.

The extension present in the Fisher Vectors consists of using more information about the distribution of codewords. Besides using information from simply counting the codewords, related to zero-order statistics, the Fisher Vectors method derive information from

the mean and standard deviation of the distribution, respectively first- and second-order statistics, through the computation of the gradient of the log-likelihood from this distribution [54].

Chapter 4

Methodology

In this chapter, we present the details of our proposed methodology for exploring and developing Deep Learning techniques, jointly with motion information, targeting at video pornography detection.

The approaches we designed, had two major inspirations. The first one was by the work of Simonyan and Zisserman [64], in which the motion information is explicitly provided to the Convolutional Neural Network, and each type of information (static and motion) is independently processed by the network. The second inspiration came from Karpathy et al. [40], in which it was expected that the CNN extracted the motion information implicitly contained on the frames. However, in our adaptation, we feed the spatial information along with explicit motion information to the CNN.

Section 4.1 presents the details of our static stream of information. Afterwards, in Section 4.2, we explain the motion stream and also the two motion sources explored in this work: Optical Flows and MPEG Motion Vectors. Section 4.3 explains the distinct ways we have explored for fusing the static and motion information. Finally, Section 4.4 details the CNN architecture and training process we adopted in this work, the GoogLeNet [69].

4.1 Static Information

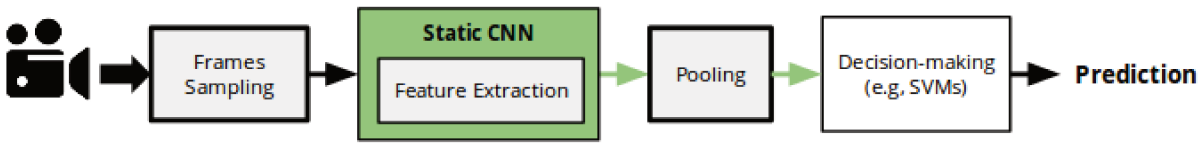


Figure 4.1: Pipeline for the static information. It comprises the feature extraction from a sampling of the frames, which are average pooled for feeding a decision-making classifier in the end.

The simplest adaptation for applying image-based techniques in videos, consists of extracting static information directly from some sampling of the video frames, and later combining this information somehow for classifying the whole video. In the static pipeline we propose, which is represented in Figure 4.1, this is performed in the following manner: a chosen sampling of the video frames (Step *Frames Sampling*) has its features extracted

with a convolutional network (Step *Feature Extraction*); these features are average pooled (Step *Pooling*) to form a single description of the whole video (there are some alternatives to the pooling, e.g., voting, and other types of pooling, such as Max and Sum, but throughout some prior experiments and our own experience, we opted for a standard average pooling procedure); finally, the video description is fed to a classifier for the final classification (Step *Decision-making*).

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

For the static ConvNet, we explored two models of the chosen architecture: the first one considers a trained model obtained with the ILSVRC dataset [61]; and a custom-tailored model fine-tuned to our problem, in which we take advantage of filter optimization techniques through back-propagation starting with the weights obtained in an ImageNet pre-training step.

4.2 Motion Information

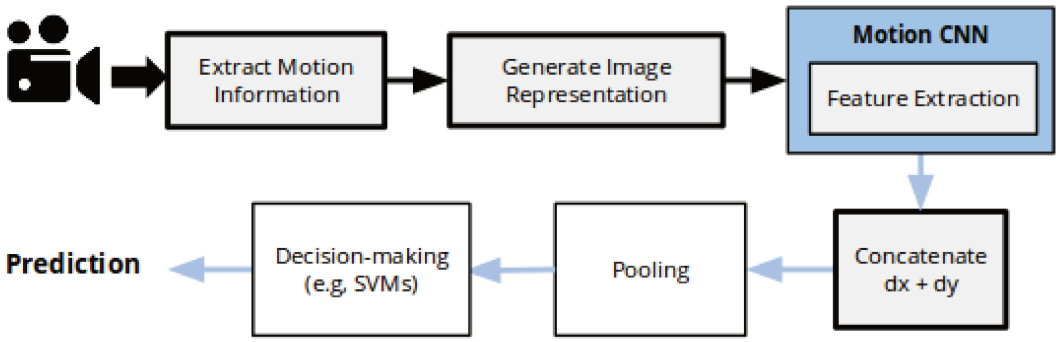


Figure 4.2: Pipeline for the motion information. It comprises: extraction of the motion information from the video; generation of an image representation to this extracted information; the feature extraction with the Motion CNN (each motion source has its own CNN model); concatenation of the horizontal (dx) and vertical (dy) descriptions; average pooling of the descriptions; and an SVM for the final classification.

As stated previously, the motion information, at first, is analyzed independently from the static information. The pipeline (Figure 4.2) for this type of information is somewhat similar to the static pipeline, with differences in the input and output of the convolutional network.

In our methodology, we evaluate two sources for the motion information: Optical Flow Displacement Fields [32] and MPEG Motion Vectors [58]. The motion sources follow the pipeline independently, therefore there is a specific Motion CNN model for each. Each source requires a unique form for extracting the motion information, whose details we shall give later on.

Differently than with the raw frames used for static analysis, the motion information does not come ready-to-use in a CNN. It is necessary to process the extracted information

(Step *Extract Motion Information*) and generate image representations (Step *Generate Image Representation*), which is the data format the CNN expects. The motion information, extracted with Optical Flows or MPEG Motion Vectors, is represented herein by two motion maps, one for the horizontal (dx) component of the motion and another for the vertical (dy), containing in each (x,y) position, a measure of motion in the respective direction. Negative and positive values in these maps indicate that the motion has happened respectively up and down for dx, or left and right for dy. When transforming these maps to images, their values are linearly rescaled to [0,255] and stored as gray-scale images, one image for each component of the motion. Fig. 4.3 depicts examples of the generated image representations.

With image representations at hand, the adopted convolutional network can process the motion information for training and for feature extraction (Step *Feature Extraction*). Each motion source will be processed by its own CNN model.

After feature extraction, the descriptions of the components (dx and dy) from the same motion are concatenated (Step *Concatenate dx + dy*) to form a single feature vector. The rest of the pipeline is similar to the static pipeline: the concatenated descriptions will be pooled (Step *Pooling*) and fed to a classifier for decision making (Step *Decision-making*). Again, each motion source will be processed by its own classifier model, giving rise to different classifications, one for each considered motion information.

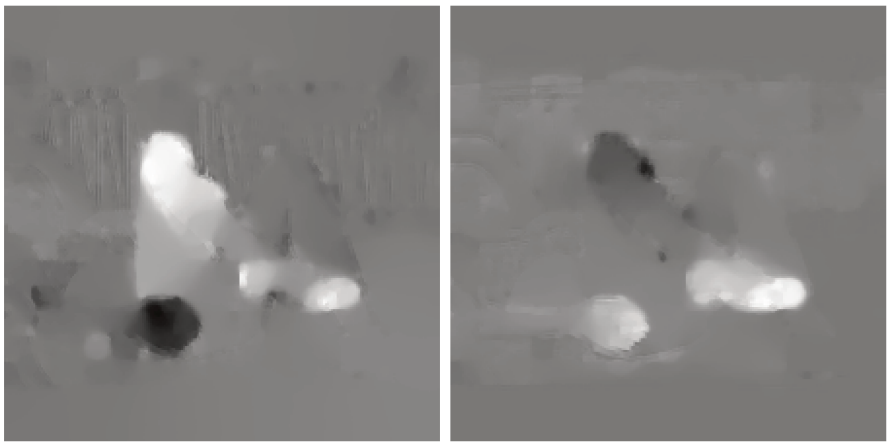
This pipeline is similar to Simonyan and Zisserman [64] temporal stream. However, here we have opted for each motion information and each component to be separately processed by the CNN, as in contrast to Simonyan and Zisserman [64], which have stacked both components of the motion information from a temporal neighborhood (e.g., displacement fields from an arbitrary number of consecutive frames) before feeding it to the network.

We have decided not to stack dx and dy components so we could have more samples for training the network, but still we bundle them, at a later stage, by concatenating the features extracted with the Motion CNN. Regarding the motion information from a temporal neighborhood, we found no improvement by inserting them, in distinct manners, on the pipeline of previous experiments. We tried stacking them before feeding it to the network, just like Simonyan and Zisserman [64]. We also tested feeding them separately to the Motion CNN and concatenating its generated descriptions, similarly to the concatenation of the components dx and dy. Given that these operations did not aid the detection of pornography, we opted not to use motion information from a temporal neighborhood.

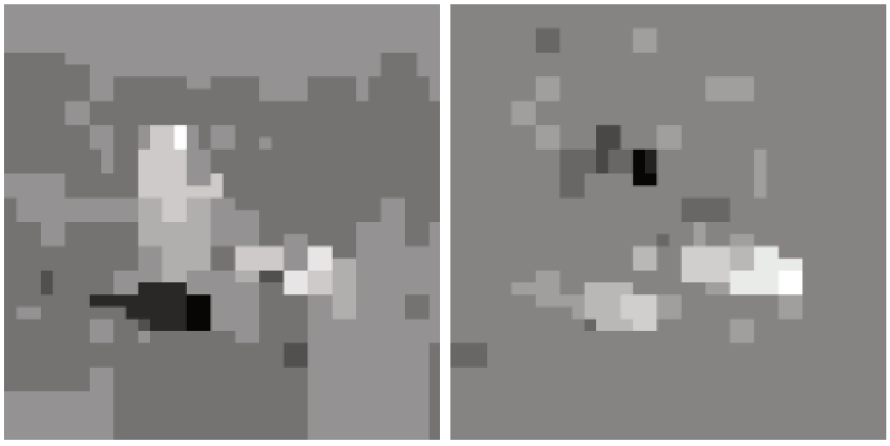
4.2.1 Optical Flow

Our first source of motion information explored is the Optical Flow Displacement Fields technique. As we explain in Section 3, there are methods that compute dense optical flow displacement fields, which characterize the visual motion of each position (pixel) between two images, in our case, two consecutive frames. Originally, the information at each position would be the magnitude and the direction of the motion. However, this information is decomposed in horizontal (dx) and vertical (dy) components, generating two motion

(a) Raw frames: Previous (left) and Next (right).



(b) Optical Flow image representation



(c) MPEG Motion Vectors image representation

Figure 4.3: Sequential Raw Frames (4.3a) and motion image representations from Optical Flows (4.3b) and MPEG Motion Vectors (4.3c). The Horizontal (dx) component is on the left, and Vertical (dy) on the right. The regions with more movement in the raw frames (e.g., hand, knee and foot) are also the regions highlighted by darker or lighter shades of gray in the motion representations of both sources.

maps with the magnitude values for each component separately. Figure 4.3b depicts an example of the optical flow representation, generated as explained in the previous section, from the generated motion maps.

The Optical Flow Displacement Fields are computed using Brox et al.'s method [11], whose GPU implementation is readily available at *OpenCV 2.4.10* toolbox. For reference, Simonyan and Zisserman [64] also used this method and toolbox, although for a completely different problem, the action recognition problem in datasets such as the UCF-101 [66] and HMDB-51 [42]. The frames, and their pairs, were preprocessed before extraction of the optical flows, just as the raw frames: resize preserving the aspect ratio, then center crop with the input dimensions of the chosen CNN.

4.2.2 MPEG Motion Vectors

Another explored source of the motion information is the Motion Vector data encoded within the MPEG codec. In each vector for a particular frame, it is encoded the position from a given macroblock of pixels in the current frame and its position at the reference, where the reference could be a frame in the past or in the future, with respect to the current frame.

Differently from previous works using these motion vectors, which performed statistical analysis on the MPEG-4 Motion Vectors [57, 36] or created images only representing how recently the motion occurred in that macroblock [7], in this work, we measure how much the block from each motion vector has moved by computing the distance, in pixel value, from the reference position to the current position in each direction separately, horizontal and vertical. Furthermore, these distances are analogous to the magnitude of the movement at the region contained in that macroblock, and generate two motion maps, one for each direction, just as in the Optical Flow motion extraction.

Motion vectors are extracted using *FFMPEG 2.7* API. They are extracted from the original videos and no resizing is performed. Therefore, differently than with Optical flow, for the motion vectors we apply the resizing operation later on, directly on the generated image representations. Figure 4.3c illustrates an example of the image generated.

4.3 Fusion

The spatial and temporal information by itself may already be very effective for the tackled problem and yield good results at pinpointing pornographic content. Notwithstanding, it is likely it will fail in some situations and probably the spatial and temporal information can lead to more effective results if their collected evidence (video telltales) are complementary in some sense. Therefore in this section, we explore different forms of combining the spatio-temporal collected information from a video and answer the third research question we posed in this work (c.f., Section 1).

4.3.1 Early Fusion

In the early fusion method, the static and the motion information are combined at the very beginning of the pipeline, being processed together by a special convolutional network. This way, the features benefit from both the static and the motion information. Hence, this approach will contain a single stream, in contrast to Simonyan and Zisserman [64], which

processed the static and the motion information separately using two distinct streams. Although using a single stream, as Karpathy et al. [40], there is a crucial difference: the motion information is explicitly provided along with the static one, meanwhile Karpathy et al. only explored static information. Figure 4.4 depicts a representation of the pipeline.

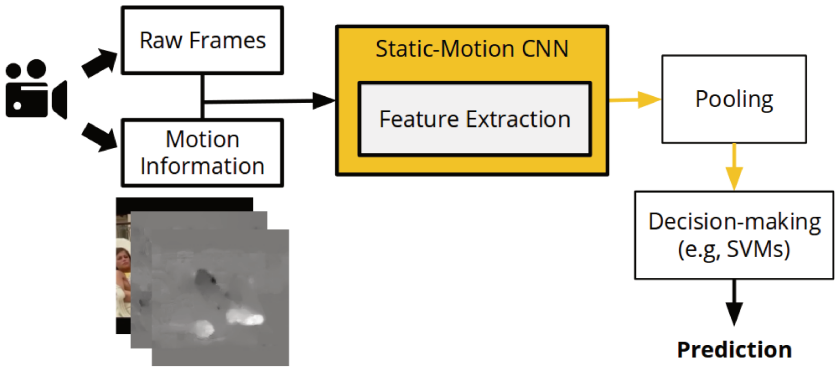


Figure 4.4: Pipeline for the early fusion. The static and motion information are combined before feature extraction, through a custom-tailored CNN trained for extracting features with both static and motion information.

The three color channels of the frame, along with its respective motion representations, dx and dy, are stacked together for input in the convolutional network, giving rise to a 5-channel input data. It is also possible to generate an image containing the raw frame information in gray scale on one of its channels and the motion information on the other two channels, one for the horizontal component and another for the vertical. The advantage of having a 3-channel input is custom-tailoring the network weights starting from ImageNet weights instead of starting the weights randomly from scratch. Our experience shows that starting with some pre-trained weights for a visual problem (image categorization) leads to better results than training from scratch if there is not enough training data as it is often common in forensic setups. In this work, we have explored both options.

4.3.2 Mid-level Fusion

Within this approach, the feature extracted from each type of information, and from each independent CNN, is concatenated on a single feature vector (Step *Concatenation*) before being fed to the classifier. Figure 4.5 shows a representation of the mid-level fusion pipeline.

4.3.3 Late Fusion

In this fusion scheme, each information is processed by a separate decision-making approach (e.g., SVM classifier), generating independent classification scores that can then be combined on a single score (Step *Average Scores*) for the final classification, similarly to Simonyan and Zisserman [64] and Jansohn et al. [36]. Figure 4.6 depicts a visual representation of this pipeline.

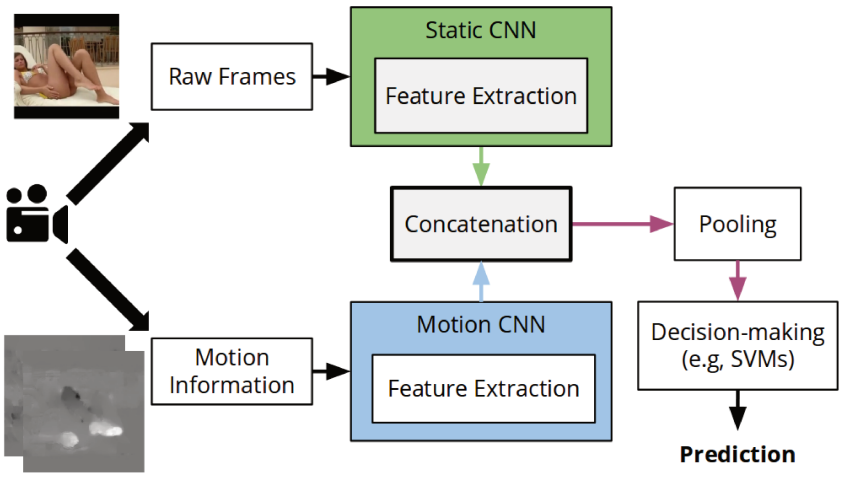


Figure 4.5: Pipeline for the mid-level fusion. The fusion of static and motion information happens after feature extraction, and before the decision making, by concatenating the feature vectors into a single description.

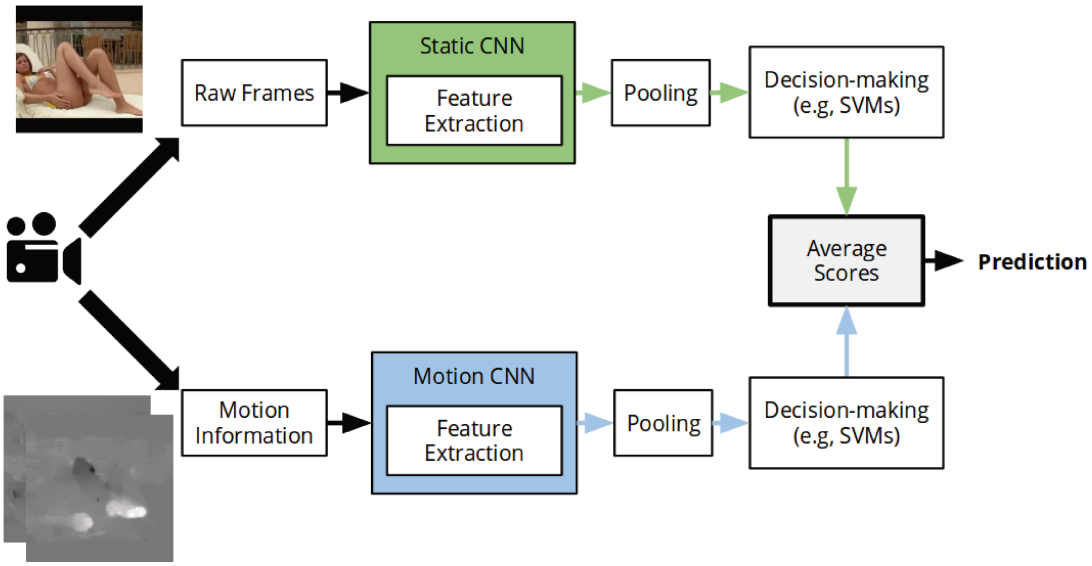


Figure 4.6: Pipeline for the late fusion scheme. The information is combined at the end, after each classifier (e.g., SVM) produces a prediction score, by averaging these scores for the final classification.

4.4 Architecture Specs - GoogLeNet

The convolutional neural network architecture we consider for the experiments was proposed in [69], and is referred to as GoogLeNet. The motivation for exploring this CNN model as a feature extractor, comes from the fact that GoogLeNet had been the winner of ImageNet 2014 Challenge [61]. This architecture managed to achieve 6.67% in the top-5 error for the competition. The ImageNet training dataset comprises about 1.2 million images, containing 1,000 classes with a wide range of subjects, from plants, animals and persons to scenes, sports and objects. Thus it is expected that GoogLeNet architecture has the capability of learning to extract highly discriminative visual features from input images. It is also expected that a model pre-trained with ImageNet 2014 dataset

should hold a very advanced state of optimization for image feature extraction, which may be very useful for application on pornography detection, by itself, or by generating a custom-tailored model with weights fine-tuned to our problem.

GoogLeNet architecture and weights are available online and ready to use off-the-shelf. We have used the distribution under Caffe [38][1] deep learning framework.

For feature extraction, we pick the output from the last layer before the final classification, which is the last average pool layer. The output from this layer has a dimensionality of 1,024-d. In order to adapt the weights for our particular 2-class detection problem, taking as input the static and motion data of interest, we tweak the classification layer filter bank from 1,000 (which is the number of classes in the ImageNet classification problem) to two (porn vs. non-porn). After that, all the network weights are fine-tuned via backpropagation.

[1] `http://caffe.berkeleyvision.org/`

Chapter 5

Experiments and Results

In this chapter, we cover the experiments performed for the evaluation of the proposed methodology. In Section 5.1, we discuss the experimental setup including: the dataset; evaluation metrics; training details; and details on the existing methods in the literature as well as third-party solutions. Then, in Section 5.2, we present the experiments and obtained results also comparing the proposed method with the existing ones in the literature, and third-party solutions.

5.1 Experimental Setup

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

5.1.1 Datasets

We have decided to use two datasets for our experiments: Pornography-800 [5] and Pornography-2k. As a matter of fact, Pornography-2k is an extension of Pornography-800, but since it is recent, there is no published work that uses it to evaluate its methods. Therefore, in this work, we report all the experiments with the proposed methods on Pornography-2k, along with the methods and solutions we choose for comparing. Meanwhile, we only evaluate our best proposed approaches on Pornography-800, for direct comparison with other works that have utilized it.

Pornography-800 dataset

This dataset[1] was originally proposed in [4] and is distributed under acceptance of an user agreement. It consists of approximately 80 hours, spanning 800 videos, 400 pornographic and 400 non-pornographic.

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

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

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

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

Pornography-2k dataset

As stated before, Pornography-2k dataset is an extended version of the Pornography-800 dataset. The new dataset comprises nearly 140 hours of 1,000 pornographic and 1,000 non-pornographic videos, which vary from six seconds to 33 minutes long.

The non-pornographic videos were acquired similarly to Pornography-800 [4], maintaining a concern for acquiring *easy* and *difficult* samples. Concerning the pornographic material, unlike Pornography-800 [4], it is not restricted to pornography-specialized websites. Instead, it was also explored general-public purpose video networks, in which it was surprisingly easy to find pornographic content. As a result, the new Pornography-2K dataset is very assorted, including both professional and amateur content. Moreover, it depicts several genres of pornography, from cartoon to live action, with diverse behavior and ethnicity. Figure 5.1 depicts some example frames from the Pornography-2K dataset.

Figure 5.1: Sample videos from the dataset. Image adapted from Avila et al. [5], with added samples.

To evaluate the results of our experiments, we consider a 5×2-fold cross-validation protocol [20]. It consists of randomly splitting the Pornography-2K dataset five times into two folds, balanced by class. In each time, training and testing sets are switched and consequently 10 analyzes for every model employed are conducted. We opted for this validation protocol as it is stricter than the traditional 5-fold cross-validation one [20].

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

5.1.2 Evaluation Metrics

The assessment of the proposed methodologies on pornography detection considers the classification of the whole video. Thus, we report the classification *accuracy* (ACC) and the F_2 *measure* (F_2), both averaged in all experimental folds.

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

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall}, \tag{5.1}$$

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

The mean over the 5×2 folds from the evaluation metrics we have chosen, ACC and F_2, may be insufficient to certify that a specific method is better than another, due to some large variations in the population of measures that may be hidden during averaging. To overcome this, we employ a Wilcoxon signed-rank test [74], which is a paired difference test that allows us to quantify how different two populations are; in this case, the populations are sampled from each fold measure from each method, without assuming a normal distribution of the population. Therefore, we can confirm more confidently whether or not they are statistically different from one another. Two methods are considered statistically different if their Wilcoxon's p-value returned is lower than 0.05 (95% confidence test).

For the Pornography-800 dataset, it will only be reported the mean video classification *accuracy* and *standard deviation*, since the works, to which we will compare ours, do not have reported the F_2 measure. Also, we do not have the result by fold from these related works, so we could not employ Wilcoxon signed-rank test [74].

5.1.3 Proposed Method's Setup

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

The same sampling was used for both the training and test phases. During training, they were utilized separately, for learning the Convolutional Network models, and pooled by movie after feature extraction, for learning the classification model. During testing, these frame/image representations pass directly to feature extraction by the trained CNN.

The training of the Convolutional Network model was performed with the Caffe frame-

work [38]. We picked the polynomial learning rate decay policy, because GoogLeNet ImageNet model we considered, from Caffe, was trained much faster using this policy. For each type or source of information, we picked a suitable value for the base learning rate, weight decay, polynomial power and the number of epochs to run. Table 5.1 shows the exact values for these hyperparameters. These learning hyperparameters were selected experimenting during training. Dropout values were maintained from original GoogLeNet Caffe architecture.

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

Table 5.1: Learning hyperparameters used for training the architecture used in this work.

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

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

We perform the final classification with a linear Support Vector Machine (SVM) classifier using *LIBSVM 3.18* [15]. We apply grid search to find the best C SVM parameter during training.

5.1.4 Comparison with Existing Methods

For a better interpretation of the performance of our proposed method, it is necessary to compare it with relevant works in the literature. We chose well-known state-of-the-art methods of low-level local image/video description, along with Fisher Vectors [55] for mid-level representation, because of their effectiveness and robustness among many distinct tasks in pattern recognition tasks, such as image classification and action recognition [70, 44, 53]. The low-level description methods we consider herein are the: Speeded Up Robust Features (SURF) [8], Space Temporal Interest Points (STIP) [43] and Dense Trajectories (DTRACK) [73]. Figure 5.2 depicts the framework applied with these existing methods.

As with the feature vector from the CNNs, the mid-level descriptions generated with the Fisher Vectors method can also be temporally pooled to form a single feature vector for the whole video. Finally, this information is fed to an SVM for label prediction. Classification is performed by Support Vector Machines classifiers using LIBLINEAR library [22]. We apply grid search to find the best C SVM parameter during training.

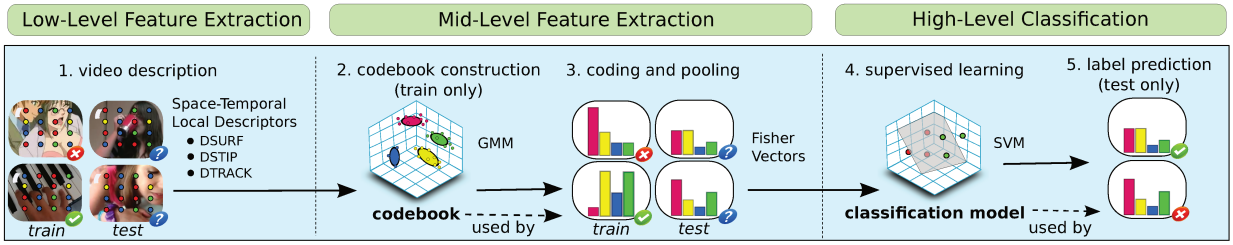


Figure 5.2: The three-layered BoVW-based framework for video pornography detection. The "training" and "test" phases are very similar, having the following two distinctions: Firstly, *Step 2. Codebook Construction* is performed only during "training", with the labeled videos, for generation of the codebook that is applied during *Step 3. Coding and Pooling* of both phases; Secondly, during the steps in *High-Level Classification*, the SVM learns a classification model from the labeled videos while in the "training" phase, later applying this model at the "test" phase for classifying the videos in the *Step 5. Label Prediction*.

Speeded Up Robust Features (SURF)

To provide a controlled baseline for the spatial techniques, we extract SURF descriptors [8], which operate over static images only. Therefore we need to use a sampling of the frames for extracting the features using this method. We use the same frame sampling of our proposed methodology, one frame per second (1fps), discarding 10% of the image borders to remove possible watermarks.

Chatfield et al. [16] demonstrated that a denser sampling of local descriptions yields a higher accuracy, so we decided not to use SURF [8] detection, instead we extract features from points in a dense grid, as it is represented in *Step 1. video description* of Figure 5.2. For increasing the scale invariance of this method, we need to extract features at different image scales, however, Bay et al. [8] actually perform this by employing distinct patch sizes for description around the points present in the grid. Thus, SURF descriptors are extracted from the sampled frames, with the borders cut, on a dense spatial grid at five scales. More specifically, we use patch sizes of 24, 32, 48, 68 and 96 pixels, with step sizes of 4, 6, 8, 11 and 16 pixels, respectively.

Space-Temporal Interest Points (STIP)

STIP [43] was the first local descriptor designed for analyzing the video space-time, therefore it can serve us as baseline of the space-temporal techniques. Although it has a specialized detector that incorporates the time dimension present in videos, we chose to employ it densely, as SURF, because of Chatfield et al.'s [16] recommendation for local descriptors.

We extract the dense STIP with the source-code provided by Laptev [43], using default values. Within these default values it also defined the density of frames sampling, since the description blob considers also the temporal dimension. The description blob extracts information from 11 consecutive frames, whose 6th one is the central frame. Default dense STIP uses as the central frame, one frame every six other frames (e.g., if we have 26 frames, descriptions are generated using as central frame the 6th, 11th, 16th and 21st

frame). This central frame sampling leads to approximately 4 frames in about 26 frames, therefore, since most of the videos have a 25fps ratio, this results in a frame sampling higher than the 1fps used by our proposed methods and SURF.

Dense Trajectories (DTRACK)

DTRACK [73] represents the current state of the art in the field of time-aware local descriptors, achieving the best results in many action recognition tasks [53], such as UCF-101 [66] and HMDB-51 [42]. This method gives us an upper bound on the previous available methods, being essential in any kind of video analysis comparison.

This method relies on a dense sampling of descriptors, not only spatially, at feature points in the starting frame, but also temporally, by tracking the feature points in the subsequent frames. We extract the dense trajectories from the video files using the source-code provided by Wang et al [73], with default values.

Fisher Vectors

Fisher Vectors are one of the best mid-level representations in the literature for computer vision problems [16, 62]. Hence, we employ it to the mid-level stage of the comparison framework.

We extract Fisher Vectors, using a Gaussian Mixture Model (GMM) to obtain the visual codebook, as suggested in [55]. The model parameters are trained over 10 million descriptions, randomly sampled (half of the descriptions sampled from positive videos, and half from negative ones in the training set), using an expectation maximization algorithm. By default, we use 256 Gaussians.

5.1.5 Comparison with Third-party Solutions

In addition to comparing the proposed method with the available scientific methods for pornography detection, we also perform comparisons with some third-party solutions readily accessible. We selected the most recent ones that rely on visual data: MediaDetective [1], Snitch Plus [3], PornSeer Pro [2], and NuDetective [56].

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 such probability is equal to or greater than 50%. NuDetective and PornSeer Pro, on the other hand, assigns binary labels to the video: positive (i.e., the video is pornographic) or negative (i.e., the video is non-pornographic).

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.

Since these solutions do not demand a training phase, they are executed directly at the dataset, without the need for training for each fold. Even so, the reported metrics are the average over all 5×2 folds, for fair comparison with the other methods.

5.1.6 Comparison using Pornography-800

After evaluating our proposed approaches with Pornography-2k and comparing it with existing methods in the literature and with third-party solutions, we employ our best approaches at Pornography-800. We do this for comparison with the reported results from other works that have also evaluated their methods on it: Avila et al. [4], Valle et al. [6], Souza et al. [67], Avila et al. [5], Caetano et al. [13] and Moustafa [51].

Avila et al. [4], which is the work that introduces the Pornography-800 dataset, employs: a HueSIFT descriptor at a regular grid of interest points for obtaining the low-level features; k-means, for construction of the codebook, with BOSSA – their proposed extension to BoVW – for the mid-level; and a non-linear SVM for the final classification. Valle et al. [6] evaluated the spatio-temporal descriptor STIP with a standard Bag of Visual Words, with random sampling for construction of the codebook, and a linear SVM. Souza et al. [67] also used a traditional BoVW, with random sampling, and a linear SVM, but with Color-STIP for low-level description. In Avila et al. [5], the authors improved their previous work [4] by proposing an extension upon BOSSA, named BossaNova, maintaining the use of HueSIFT, k-means, and a non-linear SVM. Caetano et al. [12] experimented with Binary Descriptors, of which BinBoost16 had the best performance, replacing the HueSIFT in the pipeline from Avila et al. [5] – BossaNova, k-means and a non-linear SVM.

Differently from previous approaches, Moustafa [51] did not used a BoVW-based method, instead, the author relied upon convolutional neural networks for the low- and mid-level representations and also for classification. The classification was given by a majority voting between the video frames classified using the CNN. Their best results were obtained with a max fusion of scores from AlexNet [41] and GoogLeNet [69] models, pre-trained in ImageNet and with fine-tuning of the last layer of the network using the Pornography-800 dataset.

5.2 Experimental Results

In this section, we present and discuss the obtained results from the outlined experiments. First, in Subsection 5.2.1, we assess the approaches we have proposed. Afterwards, in Subsection 5.2.2, we compare our best proposed approach to methods from the literature and third-party solutions.

5.2.1 Proposed Approaches

In Table 5.2, we show the obtained video classification *accuracy* (ACC) and F_2 *measure* (F_2) for each approach we have proposed in the static, motion and fusion modality.

In the Static stream, we show that the model relying on the GoogLeNet architecture trained with ImageNet data yields a very impressive performance of 94.56% ACC and 95.13% F_2. These results are further improved upon by fine-tuning the network weights with the pornographic data, reaching 96.00% ACC and 96.12% F_2, an improvement of 1.5 percentage point in ACC and 1 percentage point in F_2.

Table 5.2: Video classification *accuracy* (ACC) and the F_2 *measure* (F_2), averaged over the 5×2 experimental folds, from the proposed approaches on the Pornography-2K dataset. The methods are subdivided in Static, Motion and Fusion modality. Fusion is performed with fine-tuning for static information and with both motion sources, Optical Flow (O.F.) and MPEG Motion Vectors (M.V.), except Early fusion, which, due to its inferior performance with O.F, is not employed with M.V.

	Proposed Approach		ACC (%)	F_2 (%)
Static	ImageNet		94.56	95.13
	Fine-tuned*		**96.00**	**96.12**
Motion	Optical Flow		**94.40**	**95.25**
	MPEG Motion Vectors		91.00	91.96
Fusion	Early Fusion - Gray	O.F.	95.46	96.01
	Early Fusion - Color	O.F.	90.52	90.67
	Mid-level Fusion	O.F.	96.33	96.75
	Late Fusion*	O.F.	**96.39**	**96.70**
	Mid-level Fusion	M.V.	96.39	96.52
	Late Fusion	M.V.	96.37	96.59

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

When considering the motion information, Optical Flow (O.F.) by itself obtained a performance close to the static model. Meanwhile, the MPEG Motion Vectors (M.V.) had a lower performance, of only 91.00% ACC and 91.96% F_2. This difference in performance between these two sources of motion information may be explained by the fact that the M.V. represents the motion of a macroblock of pixels, which is a much lesser fine-grained description form than O.F., which takes into account the motion information for each pixel.

Despite the lower performance of motion information alone, when we combine it with the static information from the Fine-tuned network, by Mid-level Fusion and Late Fusion, we improve the ACC and F_2 results. Both Early Fusion variations, Gray and Color, yield a lower performance than using the fine-tuned static information by itself. Perhaps it is better to specialize the network to a single type of information, leaving the fusion to a higher level. Another reason may be related to the architecture considered in this work, GoogLeNet. Maybe it is not appropriate for processing five channels or combining static and motion right at the lowest level (e.g., raw data), demanding some customization such as increasing the number of filters or processing each information independently at the first layers. We believe that the better performance from the Gray variation over Color, comes from the fact that we could fine-tune its model using the ImageNet model and that the 3-channel input data is more appropriate for the GoogLeNet architecture. However, we expect that if these issues were overcome (e.g., by training an appropriate architecture with a large collection of samples), the full potential from using all color channels could be reached, outperforming the gray-only variation of this fusion, and perhaps the other

fusion approaches, Mid-level and Late.

Given the low performance of Early Fusion, and its costly requirements for training, we have opted for not fusing MPEG Motion Vectors this way.

Mid-level Fusion and Late Fusion, on the other hand, apparently could combine static and motion information better, surpassing the performance of the Fine-tuned network alone. Surprisingly, this happened even while fusing with M.V., showing that, although it had a worse performance when used alone, its complementarity to the static information is still advantageous. In addition, another advantage of using the M.V.s is that they are readily available during decoding of the video. In summary, the best method in isolation (no fusion) was *Static – Fine-tuned* while the best one considering the fusion was *Late Fusion*, which takes into account the information provided by the *static Raw Frames* and *Optical Flows* simultaneously.

With these results, we are now able to answer our Research Questions, presented in Section 1:

I) **Are Convolutional Neural Networks suitable for Pornography Detection on Videos?**
Yes, CNNs demonstrated to be appropriate for detection of pornographic content in videos, with error rates inferior to 4% when considering our best proposed approaches, Mid-level and Late Fusions taking into account Optical Flow as source of the Motion Information.

II) **Is the motion information complementary to static ones?**
Yes, by correctly combining the static information with the motion ones, we managed to outperform the methods using the static or the motion information alone.

III) **If the static and motion information are complementary,**

i) **Which is the most promising method for extracting the motion information?**
Both sources of motion information, Optical Flow and MPEG Motion Vectors, demonstrated a similar performance when combined with the static information. O.F. achieved a slightly better performance, but M.V. has the advantage of being ready to be consumed from the encoding of the video.

ii) **How the fusion of such information should occur?**
Mid-level and Late fusion had a close performance, being appropriate methods for combining the static and motion information.

5.2.2 Comparison to other solutions

For a better evaluation of the proposed approaches that obtained the best results in each modality, we compare them with the existing methods in the literature and third-party solutions. Table 5.3 contains the respective video classification *accuracy* (ACC) and F_2 *measure* (F_2) of the considered methods. As it is demonstrated, our best approaches outperform, or compare to, the other solutions.

Table 5.3: Results on the Pornography-2K dataset for the third-party solutions, other solutions in the literature and the best approaches we have proposed in each modality (Static – Fine-tuned; Motion – Optical Flow; Late Fusion with Optical Flow). We report the average performance on 5×2 folds.

	Solution	ACC (%)	F_2 (%)
Third-party	Snitch Plus [3]	66.58	46.35
	MediaDetective [1]	71.85	66.54
	NuDetective [56]	72.60	62.94
	PornSeer Pro [2]	79.10	75.61
BoVW-based	DSURF [8]	92.38	92.56
	DSTIP [43]§	94.52	94.51
	DTRACK [73]*†	95.76	95.60
Proposed Approaches	Static – Fine-tuned†	96.00	96.12
	Motion – Optical Flow§	94.40	95.25
	Late Fusion (O.F.)*	**96.39**	**96.70**

ACC: accuracy — F_2: F_2 measure

**DTRACK* and *Late Fusion (O.F.)* are statistically different (p-values: ACC ≈ 0.03; $F_2 \approx 0.002$).

†*DTRACK* and *Static – Fine-tuned* are not statistically different in ACC, but are in F_2 (p-values: ACC ≈ 0.24; $F_2 \approx 0.04$).

§*Motion – Optical Flow* and *DSTIP* are not statistically different in ACC, but are in F_2 (p-values: ACC ≈ 0.51; $F_2 \approx 0.04$).

The third-party solutions, which heavily depend on skin detection and do not take advantage of the space-time information, have demonstrated a poor performance. PornSeer Pro [2] obtained the best ACC and F_2 measures among them, with 79.10% and 75.61% respectively, far below the performance using the solutions in the literature and our proposed approaches, which have surpassed 90% in both metrics.

As stated, traditional and state-of-the-art solutions in the literature show a more reasonable performance. Dense SURF [8] (DSURF), a traditional, but not so recent and static method, manage to outperform PornSeer Pro [2] by more than 12 percentage points, achieving an ACC and F_2 measure around 92%. With the incorporation of temporal features using Dense STIP [43] (DSTIP) and Dense Trajectories [73] (DTRACK), there is a visible gain from DSURF, by two and three percentage points, respectively. As expected, since it is the state of the art in action recognition in videos, DTRACK [73] achieved the highest ACC (95.76%) and F_2 (95.60%) measures among the literature methods.

Even so, Dense Trajectories [73] (DTRACK) was outperformed by our best approaches. Not only by our spatio-temporal approach, Late Fusion (O.F.), with a gain over 0.5 percentage point in ACC (14.3% error reduction) and over 1.0 in F_2 measure, but also by our static stream, Fine-tuned, with a smaller margin, not being statistically different in ACC and being in F_2. Still, our best approach using motion information, with Optical Flow, manages to beat DSURF [8] and compares to DSTIP [43].

Table 5.4: Results on the Pornography-800 dataset for the other methods in the literature and the best approaches we have proposed in each modality (Static – Fine-tuned; Motion – Optical Flow; Mid-level and Late Fusion with Optical Flow). We report the average performance (and standard deviations) on 5 folds.

	Solution	ACC (%)
BoVW-based	Avila et al. [4]	87.1 ± 2
	Valle et al. [6]	$91.9 \pm -$
	Souza et al. [67]	$91.0 \pm -$
	Avila et al. [5]	89.5 ± 1
	Caetano et al. [12]	90.9 ± 1
	Caetano et al. [14]	92.4 ± 2
CNN	Moustafa [51]	94.1 ± 2
Proposed Approaches	Static – Fine-tuned	97.0 ± 2
	Motion – Optical Flow	95.8 ± 2
	Mid-level Fusion (O.F.)	$\mathbf{97.9 \pm 0.7}$
	Late Fusion (O.F.)	$\mathbf{97.9 \pm 1.5}$

5.2.3 Comparison using Pornography-800

In Table 5.4, we compare our best proposed approaches with the reported results from other methods in the literature that used the Pornography-800 dataset.

The proposed approaches significantly outperform the BoVW-based works in the literature [4, 6, 67, 5, 12], by 5–11 percentage points. The proposed methods also outperform, by almost 4 percentage points the results reported in Moustafa [51], which use Deep Learning as well. If we measure the classification error for comparison with Moustafa [51], we managed to reduce the error from 5.9% to 2.1%, a reduction of 64.4%. Even though we could not apply Wilcoxon's test, given the large perceptual difference in accuracy between the related works and our best approaches, with smaller standard deviation in some cases, we believe that the results would probably be statistically significant.

Though Moustafa [51] employs the same architecture used herein, GoogLeNet, there were critical differences on how he employed it, thus leading to the important difference in performance we report herein: he only fine-tuned the network last layer, while in our work we fine-tuned all layers; the network output, for each frame, was used in a majority voting scheme for classifying the video, while, in turn, we have opted for using GoogLeNet as a feature extractor, pooling the frame descriptions, then feeding them to an SVM for the video classification; finally, Moustafa only used static information, meanwhile we used static and motion information, and effective methods for combining them.

Chapter 6

Conclusions and Future Work

The designed and developed approaches we have proposed for video pornography detection in this work confirm the superiority of Deep Learning with Motion Information over third-party solutions [3, 1, 56, 2] and over previous state-of-the-art methods, not only for this specific task [6, 5, 12, 51], but also the state-of-the-art methods for related tasks, such as action recognition, when we employ it to our problem [73].

Our solution based only on the static information already yielded a competitive result with the state-of-the-art method in action recognition, Dense Trajectories [73], reaching an error rate of 4%, which is a considerably low error rate for a problem with a subjective concept as this. This leads us to believe that, for further lowering the error on this problem, it would be better to focus more on the motion information, such as adjusting the CNN – by adapting the architecture or boosting the model with more samples for training – or improving its fusion with the static information – by creating an architecture more suitable for processing the three channels, from the static frame, and the two gray channels, one from each component (dx and dy), of the motion information.

Independently of improving the classification of the whole video as pornography or not, which may be a problem close to be saturated, given the current performance reported here, we can employ our methods at a harder task, which is related to localization, in the temporal dimension, of the pornographic content in the video. The main motivation for this task is the possibility of filtering pornography in real time, which is an important matter for situations such as streaming of videos, surveillance systems and in devices that do not contain the whole video before reproducing it to the user.

Initially, our approaches can be used for tackling the localization problem, but with some caveats. For instance, with the introduction of the real-time requirement, it is necessary to take into consideration aspects such as the efficiency of the proposed approaches (e.g., test execution time and memory footprint), for filtering the content in a reasonable time and without interfering with other applications on the same system, specially if we target at mobile devices. However, we believe that the proposed approach holds potential to outperform its counterparts in terms of efficiency, specially when we think of Dense Trajectories [73] and STIP [43], widely known for their huge execution computational time. With the efficiency requirement on the table, MPEG Motion Vectors should be further explored, as it compares to Optical Flow in terms of effectiveness for complementing the static information, and it is readily available during video decoding,

being much faster to extract.

In addition to adapting our current method for the localization problem, additional neural network-based approaches could be explored as well, e,g., Long Short Term Memory (LSTM). LSTMs are a model of Recurrent Neural Network (RNN) that captures the sequential information of the input data, a highly desirable feature for classification of videos, given that they are a collection of sequential frames. The LSTM architecture could be used to process the extracted features, using the CNN models previously described in this work, from a fixed length number of frames, improving the real-time classification.

Bibliography

[1] Media Detective. `http://mediadetective.com/`.

[2] PornSeer Pro. `http://www.yangsky.com/products/dshowseer/porndetection/PornSeePro`.

[3] Snitch Plus. `http://www.hyperdynesoftware.com/`.

[4] Sadra 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.

[5] 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.

[6] Sandra Avila, Eduardo Valle, and A Luz Jr. Content-based filtering for video sharing social networks. In *Brazilian Symposium on Information and Computer System Security (SBSeg)*, 2012.

[7] Ram V. Babu and Kalpathi R. Ramakrishnan. Recognition of human actions using motion history information extracted from the compressed video. *Elsevier Image and Vision Computing*, 22(8):597–607, 2004.

[8] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *Springer Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.

[9] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798–1828, June 2013.

[10] Hajar Bouirouga, Sanaa El Fkihi, Abdeilah Jilbab, and Driss Aboutajdine. Skin detection in pornographic videos using threshold technique. *Journal of Theoretical and Applied Information Technology*, 35(1):7–19, 2012.

[11] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *Springer European Conference on Computer Vision (ECCV)*, volume 3024, pages 25–36. 2004.

[12] Carlos Caetano, Sadra Avila, Silvio Guimarães, and Arnaldo Araújo. Pornography detection using bossanova video descriptor. In *European Signal Processing Conference (EUSIPCO)*, pages 1681–1685, 2014.

[13] Carlos Caetano, Sadra 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.

[14] Carlos Caetano, Sandra Avila, William R. Schwartz, Silvio J. F. Guimarães, and Arnaldo Araújo. A mid-level video representation based on binary descriptors: A case study for pornography detection. *Elsevier Neurocomputing*, pages 1–13, 2016 in press.

[15] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:1–27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[16] Ken 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.

[17] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *arXiv preprint arXiv:1405.3531*, abs/1405.3:1–11, 2014.

[18] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Springer European Conference on Computer Vision (ECCV)*, volume 3952 LNCS, pages 428–441, 2006.

[19] Thomas Deselaers, Lexi Pimenidis, and Hermann Ney. Bag-of-visual-words models for adult image classification and filtering. In *Intl. Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, December 2008.

[20] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

[21] Piotr Dollar, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. In *IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.

[22] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *ACM Journal of Machine Learning Research*, 9:1871–1874, 2008.

[23] Margaret M. Fleck, David A. Forsyth, and Chris Bregler. Finding naked people. volume 1065, pages 593–602, 1996.

[24] David A. Forsyth and Margaret M. Fleck. Identifying nude pictures. *IEEE Workshop on Applications of Computer Vision*, pages 103–108, 1996.

[25] David A. Forsyth and Margaret M. Fleck. Body plans. *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 678–683, 1997.

[26] David A. Forsyth and Margaret M. Fleck. Automatic detection of human nudes. *International Journal on Computer Vision (IJCV)*, 32(1):63–77, 1999.

[27] Melvyn A. Goodale and David Milner. Separate visual pathways for perception and action. *Elsevier Trends in Neurosciences*, 15(1):20–25, 1992.

[28] Adrian Grange, Peter de Rivaz, and Jonathan Hunt. VP9 Bitstream & Decoding Process Specification. `http://www.webmproject.org/vp9/#draft-vp9-bitstream-and-decoding-process-specification`, 2016.

[29] Chris Harris and Mike Stephens. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, pages 189–192, 1988.

[30] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[31] Sepp Hochreiter, Fakultat F Informatik, Yoshua Bengio, Paolo Frasconi, and Jurgen Schmidhuber. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. *IEEE Press Field Guide to Dynamical Recurrent Networks*, pages 237–243, 2000.

[32] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. In *Intl. Society for Optics and Photonics Technical symposium east.*, pages 319–331, 1981.

[33] Andrew G. Howard. Some Improvements on Deep Convolutional Neural Network Based Image Classification. *arXiv preprint arXiv:1312.5402*, pages 1–6, 2013.

[34] Fu J. Huang and Yann LeCun. Large-scale learning with svm and convolutional for generic object categorization. *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:284–291, 2006.

[35] David H. Hubel and Torsten N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[36] Christian Jansohn, Adrian Ulges, and Thomas M. Breuel. Detecting pornographic video content by combining image features with motion information. pages 601–604, New York, New York, USA, 2009. ACM Press.

[37] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):221–231, 2013.

[38] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *ACM Intl. Conference on Multimedia*, pages 675–678, 2014.

[39] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *Springer Intl. Journal of Computer Vision (IJCV)*, 46(1):81–96, 2002.

[40] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[42] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *International Conference on Computer Vision*, pages 2556–2563, 2011.

[43] Ivan Laptev. On Space-Time Interest Points. *Springer Intl. Journal of Computer Vision (IJCV)*, 64(2-3):107–123, 2005.

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

[45] Quoc V. Le, Will Y. Zou, Serena Y. Yeung, and Andrew Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3361–3368, June 2011.

[46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[47] Seungmin Lee, Woochul Shim, and Sehun Kim. Hierarchical system for objectionable video detection. *IEEE Transactions on Consumer Electronics*, 55(2):677–684, 2009.

[48] Ana Lopes, Sandra Avila, Anderson Peixoto, Rodrigo S. Oliveira, Marcelo Coelho, and Arnaldo Araújo. A bag-of-features approach based on Hue-SIFT descriptor for nude detection. In *IEEE European Signal Processing Conference (EUSIPCO)*, pages 1552–1556, 2009.

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

[50] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Springer Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[51] Mohamed Moustafa. Applying deep learning to classify pornographic images and videos. *arXiv preprint arXiv:1511.08899*, pages 1–10, 2015.

[52] Fudong Nian, Teng Li, Yan Wang, Mingliang Xu, and Jun Wu. Pornographic Image Detection Utilizing Deep Convolutional Neural Networks. *Elsevier Neurocomputing*, pages 1–30, 2016.

[53] Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. *Action Recognition with Stacked Fisher Vectors*. Springer International Publishing, 2014.

[54] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[55] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Springer European Conference on Computer Vision (ECCV)*, pages 143–156, 2010.

[56] 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.

[57] Niall Rea, Gerard Lacey, Canice Lambe, and Rozenn Dahyot. Multimodal periodicity analysis for illicit content detection in videos. pages 106–114, 2006.

[58] Iain E. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. 2004.

[59] Henry A. Rowley, Yushi J. Jing, and Shumeet Baluja. Large scale image-based adult-content filtering. In *Citeseer International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 290–296, 2006.

[60] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[61] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Springer Intl. Journal of Computer Vision (IJCV)*, 115(3):211–252, sep 2015.

[62] 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.

[63] Mary B. Short, Lora Black, Angela H. Smith, Chad T. Wetterneck, and Daryl E. Wells. A review of internet pornography use research: Methodology and content from the past 10 years. *Cyberpsychology, Behavior, and Social Networking*, 15(1):13–23, 2012.

[64] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576, 2014.

[65] Josef Sivic and Andrew Zisserman. Video Google: a text retrieval approach to object matching in videos. In *IEEE Intl. Conference on Computer Vision (ICCV)*, pages 1470–1477, 2003.

[66] Khurram Soomro, Amir R. Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in the Wild. Technical report, 2012.

[67] Fillipe Souza, Eduardo Valle, Guillermo Cámara-Chávez, and Arnaldo Araújo. An evaluation on color invariant based local spatiotemporal features for action recognition. pages 31–36, 2012.

[68] Chad M.S. Steel. The Mask-SIFT cascading classifier for pornography detection. In *World Congress on Internet Security (WorldCIS)*, pages 139–142, 2012.

[69] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[70] Jasper R. R. Uijlings, Arnold W. M. Smeulders, and Remko J. H. Scha. Real-Time Bag of Words, Approximately. In *ACM Intl. Conference on Image and Video Retrieval (CIVR)*, pages 1–6, 2009.

[71] Adrian Ulges, Christian Schulze, Damian Borth, and Armin Stahl. Pornography detection in video benefits (a lot) from a multi-modal approach. In *ACM International Workshop on Audio and Multimedia Methods for Large-Scale Video Analysis*, pages 21–26, New York, New York, USA, November 2012. ACM Press.

[72] Adrian Ulges and Armin Stahl. Automatic detection of child pornography using color visual words. *IEEE Intl. Conference on Multimedia and Expo*, pages 1–6, 2011.

[73] Heng Wang and Cordelia Schmid. Action Recognition with Improved Trajectories. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3551–3558, 2013.

[74] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

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

[76] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional neural networks. *Springer European Conference on Computer Vision (ECCV)*, pages 818–833, 2014.

[77] Huicheng Zheng, Mohamed Daoudi, and Bruno Jedynak. Blocking Adult Images Based on Statistical Skin Detection. *Electronic Letters on Computer Vision and Image Analysis (ELCVIA)*, 4:1–14, 2004.

[78] Li Zhuo, Zhen Geng, Jing Zhang, and Xiao guang Li. ORB feature based web pornographic image recognition. *Elsevier Neurocomputing*, 173:511–517, 2016.

[79] Haiqiang Zuo, Weiming Hu, and Ou Wu. Patch-Based Skin Color Detection and Its Application to Pornography Image Filtering. *International Conference on World Wide Web (WWW)*, pages 1227–1228, 2010.