UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

João José de Macedo Neto

CSAM Detection based on Age Estimation From Faces

Belo Horizonte
2019

João José de Macedo Neto

CSAM Detection based on Age Estimation From Faces

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Jefersson Alex dos Santos
Co-Advisor: Filipe de Oliveira Costa

Belo Horizonte
2019

Macedo Neto, João José de.

M141C CSAM detection based on age estimation from faces [recurso eletrônico] / João José de Macedo Neto - 2019.

1 recurso online (75 f. il., color.) : pdf.

Orientador: Jefersson Alex dos Santos.
Coorientador: Filipe de Oliveira Costa.

Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 68-75

1. Computação – Teses. 2. Visão por computador – Teses. 3. Processamento de imagens – Teses. 4. Reconhecimento de padrões - Teses. 5. Aprendizado profundo – Teses. 6. Crime sexual contra as crianças. I. Santos, Jefersson Alex dos. II. Costa, Filipe de Oliveira. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. IV. Título.

CDU 519.6*84(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

CSAM Detection based on Age Estimation From Faces

JOÃO JOSÉ DE MACEDO NETO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. JEFERSSON ALEX DOS SANTOS - Orientador
Departamento de Ciência da Computação - UFMG

Dr. Filipe de Oliveira Costa - Coorientador
Centro de Pesquisa e Desenvolvimento em Telecomunicações - CPqD

PROF. FABRÍCIO BENEVENUTO DE SOUZA
Departamento de Ciência da Computação - UFMG

PROFA. SANDRA ELIZA FONTES DE ÁVILA
Instituto de Computação - UNICAMP

Belo Horizonte, 12 de março de 2019.

To my lovely wife Luciana, for the support and companionship.

Resumo

O combate à distribuição e aquisição de material de abuso sexual infantil é uma questão central para a maioria dos países e envolve agências policiais, organizações não-governamentais e empresas de todo o mundo. A detecção desse conteúdo é trabalhosa e há uma grande demanda para automatização devido à grande quantidade de dados que podem ser armazenados nos celulares e mídias atuais. Um outro fator determinante para a automatização é o impacto psicológico e o estresse causado pela exposição continuada a esse material. A maioria dos métodos para detecção automática de pornografia infantil propostos na literatura realiza uma classificação binária das imagens ou vídeos, indicando se são relacionados a pornografia infantil. Esses métodos empregam técnicas de processamento de imagens, aprendizado de máquina e reconhecimento de padrões visuais. Na área forense, uma funcionalidade desejável para tais métodos é a estimativa da idade das vítimas, mas poucos métodos a implementam. Uma das formas de atender esse requisito é combinando um detector de pornografia com um método de estimativa de idade das faces, que é uma das tarefas mais desafiadoras no campo de análise facial. Como não existem bases de dados de material de abuso sexual infantil, cuja posse é ilegal em vários países, muitos métodos são desenvolvidos usando bases de pornografia (não infantil) e métodos auxiliares. Ainda assim, a inexistência de datasets dificulta a comparação de métodos existentes. Neste trabalho, propomos uma técnica de detecção de material de abuso sexual infantil incorporando métodos baseados em técnicas de aprendizado profundo para a) estimativa de idade, que foi desenvolvido nesse trabalho; b) detecção facial, e c) classificação de pornografia. Nós também montamos um dataset anotado de material de abuso sexual infantil contendo imagens de acesso restrito à Polícia Federal do Brasil, visando avaliar e comparar métodos de detecção deste tipo de conteúdo. A abordagem proposta foi avaliada neste dataset e obteve 79.84% de acurácia na tarefa de detecção de pornografia infantil, superando duas ferramentas utilizadas na Polícia Federal do Brasil.

Palavras-chave: detecção de material de abuso sexual infantil; estimativa de idade automática; aprendizado profundo.

Abstract

The fight against the acquisition and distribution of child sexual abuse material (CSAM) is a major concern for the legal systems of most countries. It involves law enforcement agencies, non-governmental organizations, and companies around the world. Detecting such content is very labor-intensive and there is a great demand for automatic methods to support this task because of the large amount of data that can be stored in current media and mobile devices. Another determining factor for the use of automated tools is the psychological impact and stress caused by continued exposure to this type of material. Most of the proposed methods to detect child pornography automatically in literature are designed to perform binary classification of images or videos, indicating whether they are related to child pornography. These methods employ techniques of image processing, machine learning, and visual pattern recognition. In forensics, a desirable feature for such methods is the estimation of the age of the victims, but few methods implement it. One way to accomplish this is by combining a method of pornography classification with an approach for age estimation through face images, which is one of the most challenging tasks in the field of facial analysis. As there are no available CSAM datasets, since their possession is illegal in many countries, many methods are developed using datasets of pornography (without children) and auxiliary methods. Still, the lack of datasets makes it difficult to compare existing methods. In this work, we propose a CSAM detection technique incorporating deep learning-based methods for (a) age estimation, which was developed in this work, (b) face detection, and (c) pornography classification.

We also set up an annotated CSAM dataset containing images of restricted access to the Brazilian Federal Police, aiming at evaluating and comparing child sexual abuse detection methods. The proposed approach was evaluated using this benchmark dataset and achieved 79.84% of accuracy for the child sexual abuse detection task, which overcomes two tools currently used by the Brazilian Federal Police.

Keywords: CSAM detection; automatic age estimation, deep learning.

List of Figures

List of Tables

List of Acronyms

BoVW	*Bag-of-Visual-Words*
CNN	*Convolutional Neural Network*
CSA	*Child Sexual Abuse*
CSAM	*Child Sexual Abuse Material*
DNN	*Deep Neural Network*
FC	*Fully Connected*
FN	*False Negative*
FP	*False Positive*
HCI	*Human-Computer Interaction*
HOG	*Histograms of Oriented Gradients*
ICSE	*International Child Sexual Exploitation database*
LBP	*Local Binary Pattern*
LSTM	*Long Short-Term Memory*
MAE	*Mean Absolute Error*
MLP	*Multilayer Perceptron*
MSE	*Mean Squared Error*
NCMEC	*National Center for Missing and Exploited Children*
NGO	*Non-Governmental Agency*
NSFW	*Not Suitable/Safe for Work*
PCA	*Principal Component Analysis*
RBF	*Radial Basis Function*
RCPD	*Region-based Annotated Child Pornography Dataset*
ReLU	*Rectified Linear Unit*
RoR	*Residual Networks of Residual Networks*
SEIC	*Sexually Exploitative Imagery of Children*
SGD	*Stochastic Gradient Descent*

SFW	*Suitable/Safe for Work*
SIFT	*Scale Invariant Feature Transform*
SVM	*Support Vector Machines*
TN	*True Negative*
TP	*True Positive*
VPR	*Visual Pattern Recognition*
YGA	*Yamaha Gender and Age Dataset*

Contents

Chapter 1

Introduction

The ever-growing Internet access around the world has been an enabling factor for the production and dissemination of all types of data, revamping communication systems and facilitating access to books and information. Within this new reality, users can keep in touch with partners worldwide and commercialize, publish or disseminate multimedia documents. Unfortunately, this easiness is also used to share materials with illegal or abusive content, including files related to child sexual abuse (CSA).

The term child pornography is a legal term that has different meanings in distinct countries but generally refers to any content that depicts explicit sexual activity involving a child, and also includes other offenses, such as producing, consuming, sharing or possessing such material [Greijer and Doek, 2016]. Although some studies in this field initially adopted such terminology, the term child sexual abuse material (CSAM) is preferred to reference to this type of crime. The term pornography may imply the idea of consent, which a child can never give. On the other hand, the term CSAM better represents the grooming, coercion, and exploitation carried out by abusers. In this work, we use the term CSAM to refer to images and videos related to child pornography.

The task of preventing, repressing and identifying victims of child sexual abuse involves multiple actors on a national and global scale, including law enforcement agencies and non-governmental agencies (NGOs), such as The National Center for Missing and Exploited Children (NCMEC) [1], Thorn [2], ECPAT [3] and Safernet [4].

The work of analysis performed in this process involves a large volume of data. Since its inception in 2002, NCMEC has already analyzed 261 million videos and images. The International Child Sexual Exploitation database (ICSE), maintained by Interpol, contains 1.5 million distinct child sexual abuse images and videos. Big technology companies such as Google, Microsoft, Facebook, Twitter also have created initiatives to help fighting CSAM distribution, given the extent and urgency for solving this problem.

The traditional approach employed to detect CSAM is to conduct visual inspection, which is inadequate due to the large amount of data and to the stressful and exhausting

[1] http://www.missingkids.com. (Last accessed: 08.12.2018)
[2] http://www.wearethorn.org. (Last accessed: 09.12.2018)
[3] http://www.ecpat.org. (Last accessed: 09.12.2018)
[4] http://new.safernet.org.br. (Last accessed: 09.12.2018)

nature of the work. It has been reported that professionals with continuous exposure to CSAM are at risk and may suffer negative psychological impacts [Krause, 2009; Edelmann, 2010].

In recent years, automatic techniques have been developed to support CSAM analysis [Thakor, 2017]. These techniques employ distinct strategies to perform the classification task, including file name analysis, hash comparison and visual pattern recognition (VPR) methods. A special category of the last methods combines a pornography classifier with a face detector and an age estimator to classify CSA images [Yiallourou et al., 2017; Sae-Bae et al., 2014].

An advantage of the combined approach is that, besides telling certain image is related to child pornography, it can provide more information, such as gender and age or age range of both the victims and offenders who appear in the scene. This information is important because a) it allows to justify the classification procedure, indicating a criterion with legal relevance; b) it allows the separation of the content analyzed by age ranges, and c) it can be used to analyze aspects of child pornography crimes, allowing the identification of risk categories and prioritizing strategies. In this respect, a recent report, made in cooperation with NCMEC, shows that most of the victims found in child sex abuse material are female and most of the perpetrators are male [Seto et al., 2018].

The age information in child ponographic images can be of interest in a forensic scenario. It can help identify risk situations and categories with greater vulnerability to this type of crime. Also, this approach can be easily extended to videos using a frame sampling strategy, avoiding the analysis of all frames.

In addition, the development of accurate and efficient methods to detect CSAM can greatly impact other applications and areas. Some of the potential applications of these methods are described below.

- Detection tools for social networks, email providers and cloud storage services, that can be used to prevent improper use of the platform or to alert competent authorities [Shupo et al., 2006];

- Parental control applications for cameras and mobile devices [Ganguly et al., 2017; Wehrmann et al., 2018];

- Peer-to-peer network monitoring applications [Chopra et al., 2006].

In this work, we explore the CSAM detection task through the analysis of pornography level of images and age estimation of the faces. The approach designed to detect CSAM combines three methods based on convolutional neural networks: a pornography classifier, a facial detector and an age estimation method.

One problem in this research area is the lack of datasets for comparison of CSAM classification methods. This is due to the fact that the publication or possession of this

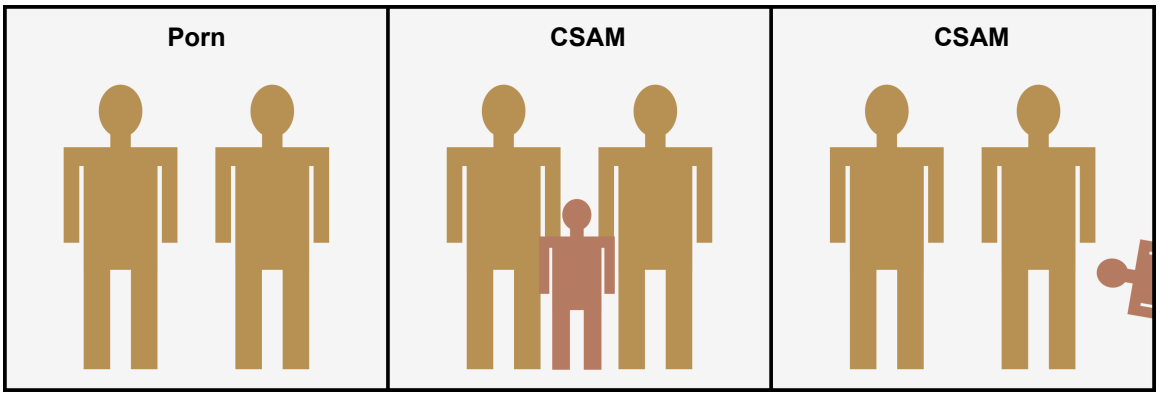

Figure 1.1: Child sexual abuse images can be very similar to pornography images. Comparing the leftmost image with the rightmost image, it is possible to verify that a small change can turn a pornography image into a child sexual abusee image.

material is illegal in many countries. We worked with the Brazilian Federal Police to introduce a region-based annotated dataset of CSAM images to evaluate and compare classification methods. Our CSAM classifier achieved an accuracy of 79.84% in the proposed dataset, surpassing two existing methods with which it was compared.

1.1 Objectives

This work aims to design, develop and evaluate solutions that allow us to detect child sexual abuse images by combining a pornography classification method with a face detector and an age estimation technique. We will exploit the fact that the presence of a child's face can turn a pornographic image into a child pornographic image. Thus, if a pornographic image is found to have at least one face classified as child, it will be considered as related to CSAM, as illustrated in the two rightmost pictures of Figure 1.1.

Comparing with an approach that simply classifies an image as related or not to child pornography, this work can provide more information, such as the amount of people found in the image in addition to the age range and gender distribution of victims and offenders. The benefits of this approach compensate for the greater weakness of this method, which are the imagens without children's faces, as shown in figure 1.2.

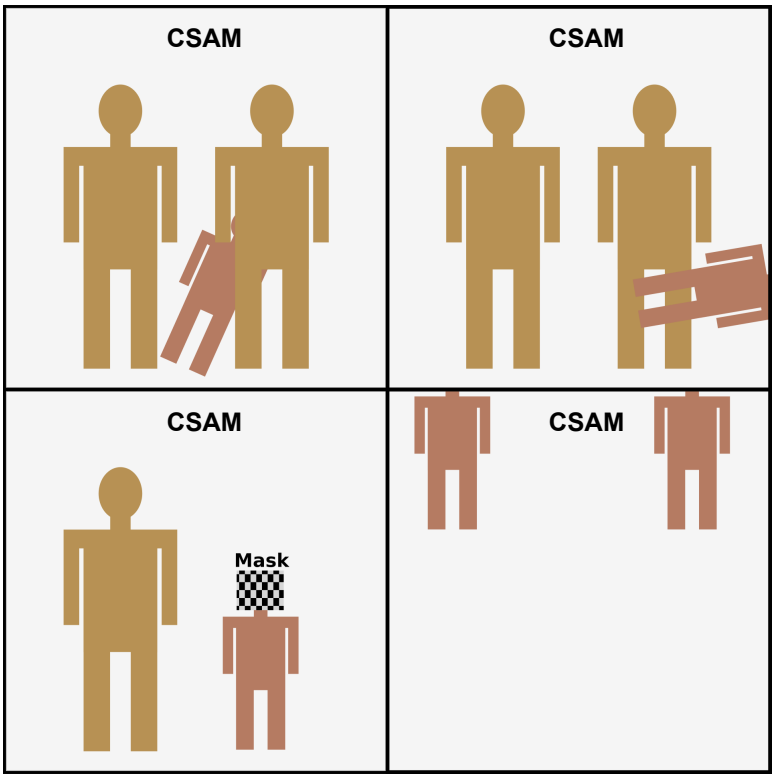

Figure 1.2: Images without children's faces. CSAM detection methods that rely on the presence of children's faces may not correctly classify images that do not display them.

1.2 Research Challenges

The CSAM classification task is challenging for several reasons. First, it is very correlated to the pornography classification problem, as shown in figures 1.1 and 1.2. In fact, CSAM classification can be seen as a special case of pornography classification where images contain one or more children and zero or more adults, resulting in a large number of possible configurations, as shown in figure 1.3. Thus, in some cases, it is very difficult to distinguish regular pornography from CSAM. Second, the images usually have poor illumination, with many occlusions, being sometimes of low resolution. Finally, there is no public available dataset for training purposes, since the distribution or possession of such data is considered a crime in many countries. Besides that, researchers have no common base to evaluate or compare their methods with the research community results.

The proposed strategy to classify CSAM assumes that a pornographic image containing a child's face must be classified as related to child sexual abuse. To achieve this goal, we combine a pornography classifier with a facial age estimator, that brings additional challenges to our work.

The task of automatic age estimation from faces is a branch of automatic face recognition and they share many processes and problems, such as face detection, face

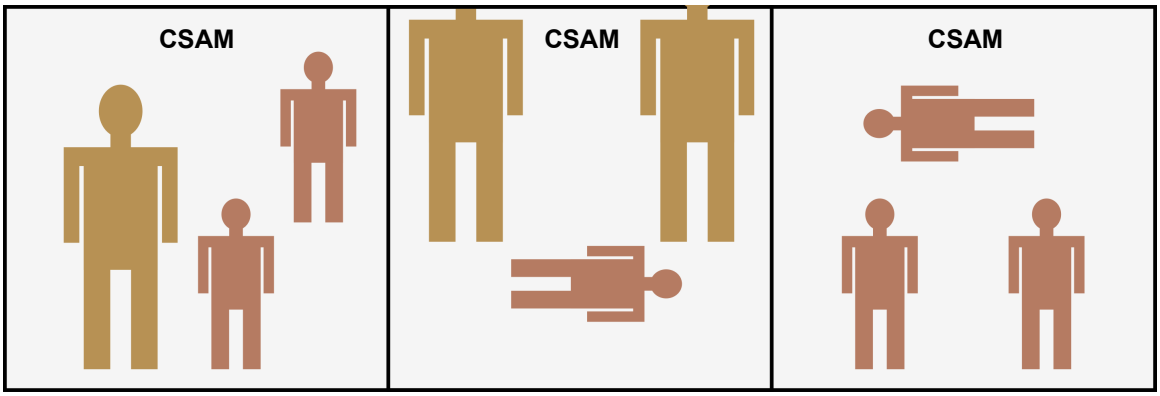

Figure 1.3: Child sexual abuse images can have multiple configurations.

alignment, feature extraction and datasets [Dong et al., 2016].

The age estimation task poses many difficulties that are related to the uncontrolled and personalized nature of the human aging process, which is not completely understood. Each person has its own aging pattern, rate and characteristics, that are different from everyone's else and are determined by genetic factors and by extrinsic and environmental factors, such as exposure to extreme temperatures or climates, sunlight, ultraviolet radiation, wind and arid air and also smoking, diseases, living style, sociality, among others [Albert et al., 2007; Fu et al., 2010].

Besides that, the images can present poor illumination and low resolution and the faces in the image can appear in several poses or with partial occlusions. The influence of all these factors shows that the visual aspects among faces present many variations not only across gender, ages and ethnic groups, not to mention the effects caused by the usage of accessories, hats, glasses, mustaches, beards and make-ups. Figure 1.4 shows some examples of such variations, illustrating the use of hats, glasses, beards and some images with different poses, with partial occlusions and low resolution.

1.3 Scientific Contributions

The main contributions of this work are:

- The evaluation of an approach to CSAM classification based on the combination of auxiliary classifiers and its usefulness in a forensic context;
- The development of an age estimation technique that is used as one of the auxiliary classifiers in the proposed CSAM classification method;

Figure 1.4: Faces from adience dataset [Eidinger et al., 2014].

- The introduction of a region-based annotated dataset belonging to the Brazilian Federal Police. This dataset forms the basis for a benchmark methodology for CSAM classification methods, that may be used for testing new techniques and can stimulate further research in this area.

1.4 Publications

During the development of this work, we published a paper entitled "A Benchmark Methodology for Child Pornography Detection" [Macedo et al., 2018] on the 31st conference on graphics, patterns and images (SIBGRAPI'18), held in Foz do Iguaçu, Brazil, from October 29 to November 1.

1.5 Dissertaton Roadmap

The remainder of this work is organized in five chapters.

Chapter 2: Background: Presents the main background concepts related to this work, including machine learning basics and convolutional neural networks.

Chapter 3: Related Work: Presents reference works to the proposed method.

Chapter 4: Methodology: Describes the proposed method for CSAM classification and the Region-based Annotated Child Pornography Dataset (RCPD).

Chapter 5: Experiments: Describes the setup of the auxiliary methods and the experiments that evaluated the proposed method and two forensic tools on the RCPD dataset.

Chapter 6: Conclusion and Future Work.

Chapter 2

Background

The field of visual pattern recognition involves knowledge of several disciplines, such as probability, statistics, signal processing, digital image processing, machine learning, neurobiology, etc. The explanation of all the areas involved goes beyond the objectives of this work, so we will focus on machine learning topics that are more relevant to the comprehension of the proposed method.

2.1 Machine Learning

The recognition of visual patterns by computer programs presents special challenges which make it difficult the employment of traditional approaches based on algorithms that explicitly transform an input into an output. When working with images, for example, a definition of a sequence of commands to detect an object can become overly complex and with an excessive number of variations and special cases that makes its generic use an impracticable strategy. In some cases, an exact algorithm may not exist or we may not know how to express it [Alpaydin, 2010]. One of the main difficulties in this area is the high dimensionality of data [Sebe et al., 2005]. A relatively small image with a resolution of 640 by 480 pixels has more than 300 thousand pixels, each one associated with three numbers, referring to the red, green and blue color channels. This results in almost one million numbers that should be analyzed for a pattern recognition task in this image.

A more effective approach to performing visual pattern recognition tasks is the use of machine learning techniques. These techniques rely on the analysis of a large number of input data samples and *learn* to determine the output based on these examples [Alpaydin, 2010]. Thus, instead of explicitly programming the computer to recognize a face, for example, this strategy suggests that a large number of faces are obtained, preferably with variation of gender, age and ethnicity. A desirable characteristic of the data is that they cover as many special cases as possible, such as wearing hats, glasses, beard,

accessories, makeup, as well as variations of pose, lighting and image quality. From these data, machine learning techniques can be employed to perform the face recognition task, without explicitly defining how to recognize a face or what a face should contain.

2.1.1 Definition

Machine learning can be defined as a form of applied statistics, allowing computer programs to statistically estimate complex functions [Goodfellow et al., 2016]. It can also be defined as a set of automatic methods for data analysis and pattern detection that are used to predict future data or to aid in decision making [Murphy, 2012].

Machine learning is an area of artificial intelligence that focuses on algorithms that learn from data, as humans do. It is concerned about algorithms that can improve its performance through experience. The concept of learning was formally delimited by [Mitchell, 1997], who stated that: *a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.* There can be several tasks T, experiments E and performance measures P, depending on the application.

The tasks are the problems that are being solved, like classification of images, estimation of price or clustering customers by their purchasing profile. The experiences are the inputs x or the dataset that the algorithm has access. Individual elements x_i of the dataset are sometimes referred to as *datapoints* or $features$. Each data point x_i is usually a vector of values related to one sample. In some cases, each sample also has an associated *label*, $target$ or output y_i, which is also stored in the dataset [Murphy, 2012]. For instance, the features in a dataset to train a predictor of risk of heart attack could be the gender, height, weight and race of an individual. The target could be the age at which the individual had a heart attack. A machine learning algorithm could be trained with this dataset to predict the risk of heart attack of unseen samples. The performance measures or evaluation metrics depend on the nature of the task. Some common choices are accuracy and mean average error (MAE), which will be detailed later.

Within this framework, for a task T of age estimation from face images, for example, the performance measure P can be accuracy and the experience E can be the face images used for training, each one labeled with their respective age.

2.1.2 Types of Learning

There are two main types of learning problems: supervised and unsupervised. This division is based on the configuration of the data to which the algorithms have access. If the data points x_i have an associated label or target y_i (available at training time), the learning is supervised. If the data is not labeled or the training has no access to it, the learning is unsupervised. Supervised learning problems are usually related to predictions and the objective is to learn a mapping from the input to the output. The most common supervised learning tasks are classification, where data is categorized into two or more classes, and regression, which assigns a quantitative value to the data. In unsupervised learning problems, the goal is to find regularities or understand the structure of the input data, such as in clustering tasks [Alpaydin, 2010].

There are other categories of learning, such as reinforcement learning, in which the algorithm also receives inputs from the environment. These inputs are considered in the learning problems and the learning is based on actions and rewards. Usual applications of reinforcement learning include game playing and robotics [Alpaydin, 2010].

2.1.3 Evaluation Metrics

The evaluation metrics used to measure the performance of machine learning algorithms depend on the nature of the task. Some examples are:

Accuracy: is the ratio of correct predictions and the total number of predictions N [Bruce and Bruce, 2017]:

$$Accuracy = \frac{CorrectPredictions}{N} \tag{2.1}$$

In a binary classification task, where the output assume a value of Yes or No, each prediction can be classified as True Positive (TP), when prediction is Yes and the correct output is Yes, True Negative (TN), when prediction is No and the correct output is No, False Positive (FP), when prediction is Yes and the correct output is No, and False Negative (FN), when prediction is No and the correct output is Yes. With these definitions, the accuracy can also be defined as:

$$Accuracy = \frac{\sum TP + \sum TN}{N} \tag{2.2}$$

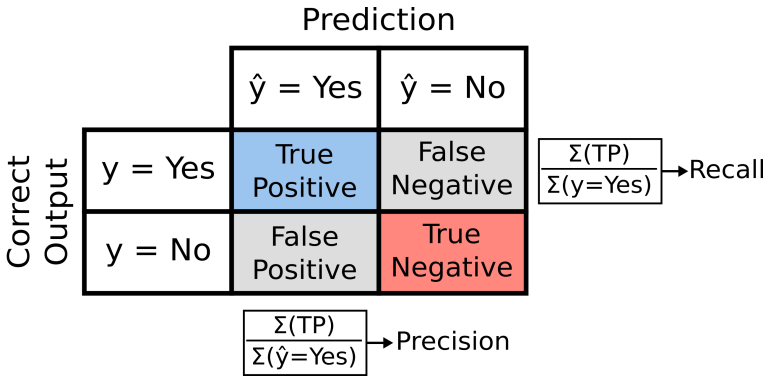

Figure 2.1: Confusion Matrix: relation between predictions and correct ouptups.

The relation between the correct and incorrect predictions can be illustrated in a confusion matrix, shown in Figure 2.1, which is a table that correlates the predictions $\hat{y}$ and the correct outputs y. The values in the confusion matrix are also used to calculate two other measures, called *precision* and *recall*. The precision is the relation of the true positive predictions with all positive predictions ($\hat{y} = Yes$) and the recall is the relation of the true positive predictions with all true outputs ($y = Yes$) [Bruce and Bruce, 2017].

$$Precision = \frac{\sum TP}{\sum TP + \sum FP} \tag{2.3}$$

$$Recall = \frac{\sum TP}{\sum TP + \sum FN} \tag{2.4}$$

F1-score: is a measure that correlates precision and recall, where:

$$F1 - score = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{2.5}$$

Mean absolute error (MAE): is the average of the difference between the predicted outputs $\hat{y}_i$ and the correct values y_i, considering the total number of predictions N:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{2.6}$$

Mean squared error (MSE): is the average of the squared difference between the predicted outputs $\hat{y}_i$ and the correct values y_i:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i|^2 \tag{2.7}$$

When a performance metric is used during the training of a machine learning algorithm, it is referred to as *loss function* or *cost function*. In this context, it is usually associated with an optimization procedure that aims to minimize it.

2.1.4 Generalization

In a typical machine learning configuration, the learning step is performed with one part of the dataset that is called *training set*. The evaluation of the algorithm is performed in a separate part of the dataset, called *test set*, which cannot be accessed by the algorithm during the training phase. One of the major goals of the learning algorithm is to achieve good *generalization*, which refers to the ability of the algorithm to have good performance on *unseen* data (on the test set), achieving small training error and a small gap between training error and test error [Goodfellow et al., 2016].

In the machine learning context, to perform a specific task, a representation of the problem (also called *model*) must be chosen. A model can be viewed as a general structure to represent the problem, or a hypothesis that can learn certain functions [Alpaydin, 2010]. This structure usually can be refined with the choice of parameters that affect the algorithm's operation, which are called *hyperparameters*.

For a given model, it is also possible to express some preferences about the behavior of the algorithm using a technique called *regularization*. The main objective of regularization is to reduce the gap between training error and test error, achieving better generalization. Most regularization techniques also involve the choice of hyperparameters.

The main strategy to choose the hyperparameter values is separating a part of the training set to validate them. This part is called *validation set*. One other strategy, called *cross-validation*, consists in partitioning the data in k folds, performing the training with $k-1$ folds and evaluating the model in the remaining fold. The process is repeated k times, always evaluating the model with a different fold. In the end, the performances of the models are averaged [Bishop, 2009].

In a learning task to fit a polynomial curve, for example, the choice of the order m of the polynomial is a hyperparameter that affects the generalization of the model. It is possible to express a preference over the size of the polynomial coefficients c using a regularization technique called *weight decay*, which penalizes high values of these parameters [Bishop, 2009]. Assuming that the learning algorithm uses an MSE loss function, this function can be modified to include the regularization penalty as follows:

$$MSE' = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|^2 + \lambda\sum_{j=1}^{m}c_i \tag{2.8}$$

The weight decay penalty in the modified MSE loss function includes the term λ that is also a hyperparameter.

Sometimes the *capacity* of the model is not sufficient to represent the complexities of the training data, leading to a situation called *underfitting*, in which the learning algorithm does not have good performance on the training set. In these cases, the model

probably will not have good generalization and the solution is to choose a model with greater capacity. An example of this is choosing a line to represent a set of data whose generating function is quadratic. The chosen model in this case is the function $\hat{y} = a*x+b$, with just two parameters. No matter how good a learning algorithm chooses the best values for a and b, the performance of the model in training will not be satisfactory if the training data points form a sharp curve. The model also will not generalize to the test data.

An opposite situation can occur when the model has sufficient capacity to represent the training set, in which it achieves good performance, but when the model is evaluated on the test set, it achieves a performance much lower than that presented in the training set. This situation is called *overfitting*. In other words, the model fits the training data but does not generalize to unseen data (test set). In this case, a careful choice of the hyperparameters and the use of regularization techniques can help to achieve better generalization.

2.1.5 Visual Pattern Recognition

Visual pattern recognition combines machine learning and digital image processing techniques to extract information from digital images and videos. Given the high dimensionality of the data and the difficulties to grasp semantic information from a huge grid of pixels, traditional approaches do not apply the machine learning algorithms to the images' data directly. Instead, some representative data are firstly extracted from the images, which are called *features*, and then used in the machine learning algorithms to perform specific tasks, such as classification or detection. The task of generating features, which is called feature engineering, is one of the most laborious and important tasks of the learning process and, for a long time, this was an active research area.

This approach was prevalent until the beginning of this decade. But with the availability of bigger datasets, the improvement of computing resources with faster CPUs and powerful GPUs and the advances of software infrastructure, the deep learning techniques started to have great success and making a great impact in many machine learning tasks. In this approach, the features are learned instead of *engineered*. An important milestone of this revolution happened when a deep convolutional neural network (CNN) [Krizhevsky et al., 2012] won the ImageNet Large Scale Visual Recognition Challenge, the most important contest in object recognition, improving the previous error rate of 26.1 percent to 15.3 percent [Goodfellow et al., 2016].

2.2 Deep Neural Networks

In this section, we describe the core concepts of deep neural networks (DNNs), including the network architecture and specific details such as the learning process and regularization techniques. DNNs can be defined as feedforward networks with multiple layers between the input layer and the output layer. These concepts are important because they are the framework upon which the convolutional neural networks (CNNs) are built. We will start defining the most basic units of a neural network, the perceptrons, sometimes referred to as *neurons*.

2.2.1 Perceptrons

Feedforward networks, also known as multilayer perceptrons (MLP), are an arrangement of basic elements called perceptrons, which are very simple units, with limited representation capacity.

The processing of a perceptron can be described in two operations. First, it performs a linear combination of N inputs x_i and coefficients or weights w_i, which are added to a *bias* or intercept term b:

$$a(x) = \sum_{i=1}^{N} (w_i * x_i + b) \tag{2.9}$$

To obtain a more concise equation, the bias term b can be placed at the index 0 of the weights w (w_0) and a constant 1 is placed at the index 0 of the input x (x_0). The equivalent expression is:

$$a(x) = w^T x \tag{2.10}$$

Then, a step function is applied to the linear combination. The step function is a nonlinear function that yields 0 if the input is equal or lower than 0 and 1 otherwise. The resulting output of a perceptron is:

$$h = \begin{cases} 0 & (w^T x) \leq 0 \\ 1 & (w^T x) > 0 \end{cases} \tag{2.11}$$

Figure 2.2 illustrates three representations of a perceptron, with distinct degrees of abstraction. Concise representations are usually preferred for convenience in these graphical representations.

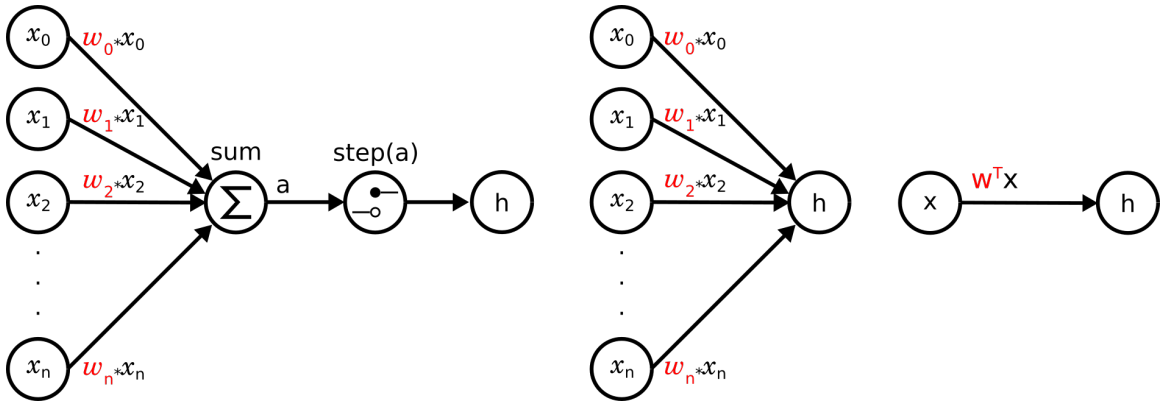

Figure 2.2: Three equivalent representations of a perceptron. (left) An analytical representation. (middle) The sum and activation operations are abstracted in the h node. (right) A more concise representation.

2.2.2 Network Architecture

In a feedforward network, multiple perceptrons, also called *units*, are combined in *layers*, forming an acyclic network with no feedback connections, where the output units $h_i^{(j)}$ of layer j work as the input of the units $h_i^{(m)}$ of subsequent layers ($m > j$). The intermediate layers are referred to as *hidden layers* and the last layer is referred to as the *output layer*.

One important difference between the units of a feedforward network and the traditional perceptrons is in the nonlinear function. There are many possible nonlinear operations, such as the classical logistic sigmoid and the *tanh* functions. In modern feedforward networks, the most recommended nonlinear operation to use is the rectified linear unit (ReLU) function [Nair and Hinton, 2010], that is the function $f(x) = max(0, x)$. With this configuration, the units vector $h^{(1)}$ of the second layer of a multilayer perceptron is given by:

$$h^{(1)} = max(0, W^{(1)T}x) \tag{2.12}$$

In the following layers, units vectors $h^{(j)}$ are computed by:

$$h^{(j)} = max(0, W^{(j)T}h^{(j-1)}) \tag{2.13}$$

Figure 2.3 illustrates an analytical and a synthetic representation of a feedforward network with two hidden layers. The input layer consists of three values: an x_0 term equal to 1 for the bias term and two data values, x_1 and x_2. The second layer contains 4 units (perceptrons) and the third layer contains 2 units. The final layer contains the output of the network, with two values, y_0 and y_1. The set of all weights $W^{(i)}$ are simply designated by W, and they represent the parameters of the network.

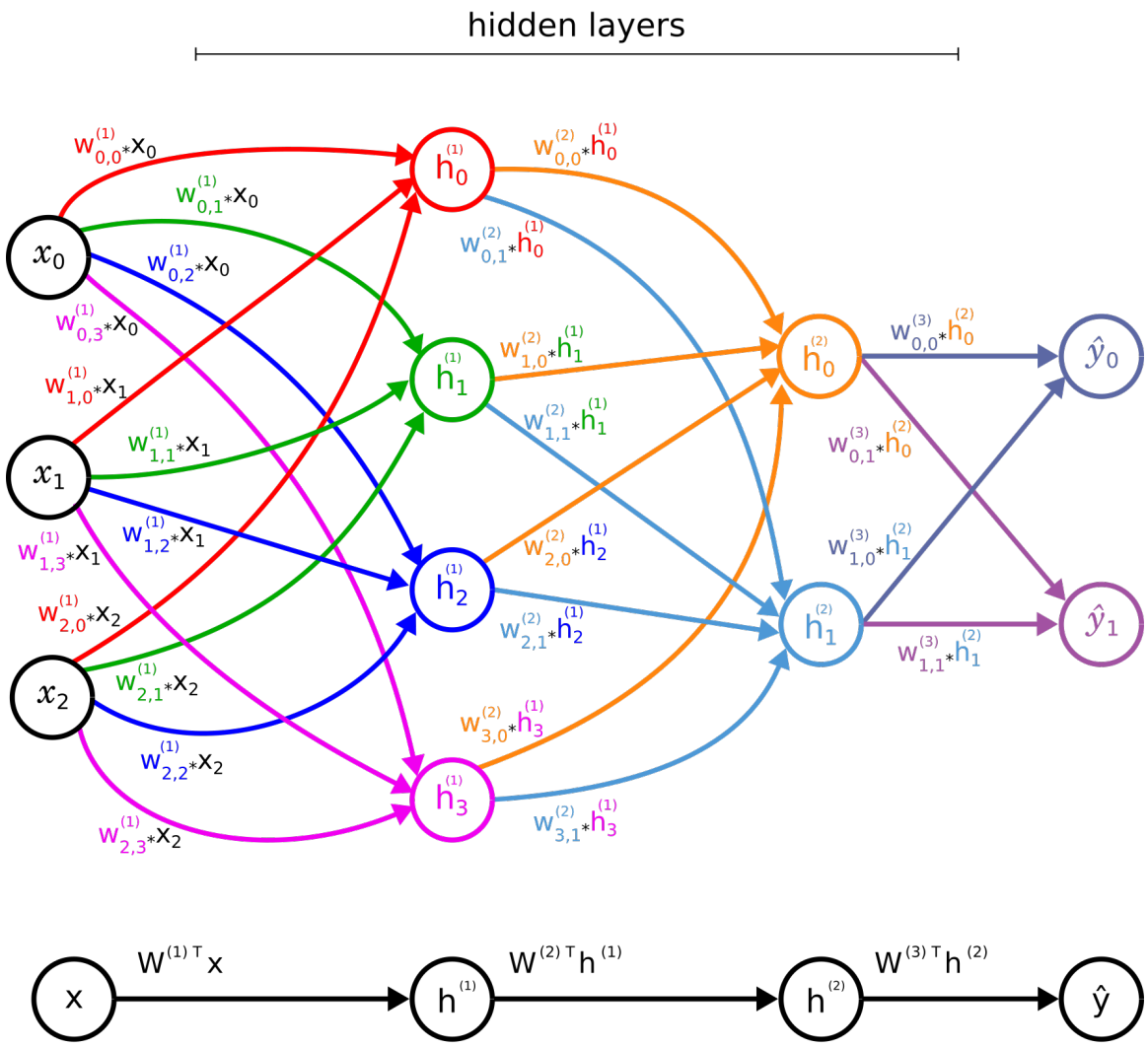

Figure 2.3: An analytical and a concise representation of a feedforward network with two hidden layers.

The activation function of the output units of the network depends on the nature of the problem. For regression problems, for example, the activation function can be the identity $y_k = a_k$. For multinomial (multiclass) classification problems, the activation function can be the softmax function [Bishop, 2009]:

$$y_k(a) = \frac{e^{a_k}}{\sum_j e^{a_j}} \tag{2.14}$$

To complete the definition of the *forward propagation* in a multilayer perceptron network, we need to define a loss function L for the network. The definition of such a function allows to evaluate how well the parameters $W^{(i)}$ are *mapping* the inputs to the correct outputs and will be an essential element to *guide* the optimization process, where the parameters W of the network are learned.

The choice over the loss function depends on the type of the learning problem and it is usually associated with the activation function of the output units [Goodfellow et al.,

2016]. In a regression problem, it can be an MSE function. In a multiclass classification problem, it can be a cross-entropy loss. In this case, assuming that the output y_k is the softmax activation for each class k and y_c is the prediction of the correct class for the input that is being *processed*, the loss function L_i of input i can be expressed by:

$$L_i = -log\left(\frac{e^{y_c}}{\sum_j e^{y_j})}\right) \tag{2.15}$$

That is equivalent to:

$$L_i = -y_c + log\sum_j e^{y_j} \tag{2.16}$$

2.2.3 The Learning Process

The network structure presented so far describes the forward flow of information from the input to the output layer. But for the network to learn, it is necessary to compute and update the parameters of the network, and for this purpose two algorithms are used: the backpropagation algorithm and the gradient descent algorithm (or one of its variations).

Compute the Gradients: the backpropagation algorithm allows to compute the gradients of the network. The algorithm uses a computation graph that represents the network structure, containing all the mathematical expressions that are used in the forward pass of the network, including the computation of the loss function. The algorithm then uses the chain rule to calculate the derivatives of all nodes of the computation graph, starting from the loss function. For a given variable n of the computation graph, with one or more child nodes c_i, its gradient can be computed by the sum of the dot products of the gradient of the loss with respect to the children nodes c_i and the partial derivative of c_i with respect to n [Goodfellow et al., 2016]:

$$\frac{\partial L}{\partial n} = \sum_i \frac{\partial L}{\partial c_i}\frac{\partial c_i}{\partial n} \tag{2.17}$$

When dealing with vector based values, the preferred notation is:

$$\nabla_n L = \sum_i (\frac{\partial c_i}{\partial n})^T \nabla_{c_i} L \tag{2.18}$$

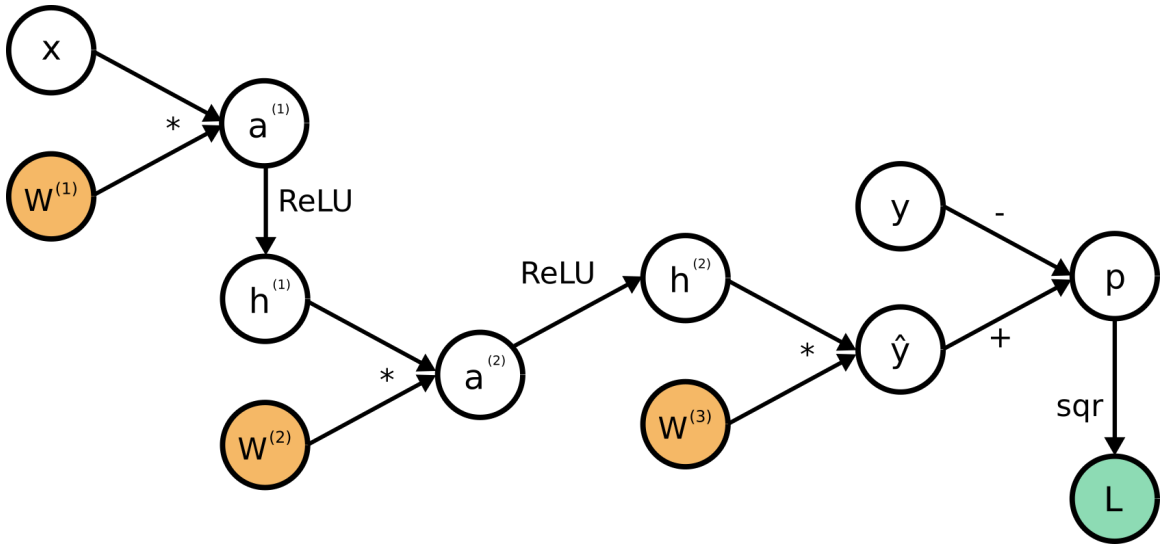

Figure 2.4: Computation graph of the feedforward network of Figure 2.3 using a MSE loss function.

An example of a computation graph that represents the network shown in Figure 2.3 using a MSE loss function is illustrated in Figure 2.4. Using Equation 2.18, the gradients of the nodes of the computation graph of Figure 2.4 can be obtained:

$$\nabla_p L = \frac{\partial L}{\partial p}$$

$$\nabla_{\hat{y}} L = \left(\frac{\partial p}{\partial \hat{y}}\right)^T \nabla_p L$$

$$\nabla_{W^{(3)}} L = \left(\frac{\partial \hat{y}}{\partial W^{(3)}}\right)^T \nabla_{\hat{y}} L$$

$$\nabla_{h^{(2)}} L = \left(\frac{\partial \hat{y}}{\partial h^{(2)}}\right)^T \nabla_{\hat{y}} L$$

$$\nabla_{a^{(2)}} L = \left(\frac{\partial h^{(2)}}{\partial a^{(2)}}\right)^T \nabla_{h^{(2)}} L$$

$$\nabla_{W^{(2)}} L = \left(\frac{\partial a^{(2)}}{\partial W^{(2)}}\right)^T \nabla_{a^{(2)}} L$$

$$\nabla_{h^{(1)}} L = \left(\frac{\partial a^{(2)}}{\partial h^{(1)}}\right)^T \nabla_{a^{(2)}} L$$

$$\nabla_{a^{(1)}} L = \left(\frac{\partial h^{(1)}}{\partial a^{(1)}}\right)^T \nabla_{h^{(1)}} L$$

$$\nabla_{W^{(1)}} L = \left(\frac{\partial a^{(1)}}{\partial W^{(1)}}\right)^T \nabla_{a^{(1)}} L$$

The algorithm computes the gradient of each node n in the backward direction of the computation graph using the gradients of the nodes that are children of n. For

example, when calculating the gradients of the weights $W^{(2)}$, the algorithm's equation $\nabla_{W^{(2)}} L = (\frac{\partial a^{(2)}}{\partial W^{(2)}})^T \nabla_{a^{(2)}} L$ uses the already computed gradient $\nabla_{a^{(2)}} L$ with respect to $a^{(2)}$ and just needs to compute the partial derivative of the child node $a^{(2)}$ with respect to $W^{(2)}$, which is an easier task than directly computing the gradient of L with respect to $W^{(2)}$.

Update the Parameters: once the gradients of the network are computed, an optimization algorithm like gradient descent can be used to update the parameters of the network. The basic idea of the gradient descent algorithm is that the gradient of a variable Y with respect to a variable X, defined by $\nabla_X Y$, points to the direction of X that increases Y. Thus, in the feedforward networks context, after computing $\nabla_W L$, if we make small changes to W in the opposite direction of the gradient, we are minimizing the loss function L. The general form of the update procedure is given by:

$$W_{(i+1)} = W_{(i)} - \epsilon * \nabla_{W_{(i)}} L \tag{2.19}$$

The term ϵ in Equation 2.19 is the *learning rate*, which is an important hyperparameter of the model. This process is repeated until some stop criterion is reached. When the updates of the network parameters are performed with the evaluation of a single sample of the training set, the method is called *Stochastic Gradient Descent* (SGD). When the updates are related to small batches of the training set (usually of 32, 64, 128 or 256 samples), the method is called *Mini-Batch Gradient Descent*, and when the updates are related to the entire training set, it is called *Batch Gradient Descent* [LeCun et al., 2012].

There are many methods that are variations of this basic framework, using different strategies to accelerate the learning process and favor convergence, like *Momentum*, *Nesterov Momentum*, *RMSProp* and *Adam* [Ruder, 2016].

2.2.4 Regularization

As with other machine learning algorithms, the feedforward networks can adopt regularization strategies to achieve better generalization. Some of these strategies are common to other machine learning algorithms and others are specific to the neural networks.

Weight Decay: one of the most basic regularization techniques is the weight decay, which is one technique used to penalize high values for the weights by adding a fraction of the sum-of-squares of the weights to the loss function [Krogh and Hertz, 1992]. The goal

of this technique is to prevent that some few weights assume high values while others are squeezed to values close to zero. It assumes that a more desirable configuration is that all the weights have a contribution to the output, which in general leads to better generalization. Adding this regularization to the softmax loss function presented in Equation 2.16, the loss function turns into:

$$L_i = -y_c + log \sum_j e^{y_j} + \lambda * \sum_k \sum_l |W_{k,l}|^2 \tag{2.20}$$

Data augmentation: training the model with more data is a simple strategy to increase the invariance of a model to the input data, and thus the generalization. Unfortunately, this is not an option in many scenarios, where the amount of data is limited. An alternative option is to increase the amount the training data is to apply a set of transformations to the training samples in order to generate transformed versions of the original data [Bishop, 2009]. This regularization technique is called data augmentation and is very used in small or medium-sized datasets of images, where the original images are randomly cropped, rotated, flipped and/or translated within preset parameters.

Early stopping: early stopping is a simple regularization technique that consists of stopping the training procedure when the network error stops decreasing. A better approach consists in saving the models with the smallest error during training and then selecting the best model.

Dropout: dropout is a simple and very effective regularization technique that consists in temporarily dropping out some units of the network during training, including its incoming and outgoing connections. Each unit has a probability p of being *retained* and the choice over all units is random in each update step of the learning process. Dropout can be viewed as a regularization technique that combines different neural network architectures, producing a model that averages all predictions, leading to less overfitting and better generalization [Srivastava et al., 2014].

Multitask Learning: the idea behind this technique is to train a model that simultaneously learns different and correlated tasks, and part of the model is shared across tasks, which can lead to better generalization. More details in [Caruana, 1997, 2012].

Parameter Sharing: if we analyze the feedforward network illustrated in Figure 2.3, it is possible to verify that all hidden neurons h_i^l have its own set of parameters $W_{j,i}^l$. The parameter sharing regularization technique, in a different way, constrains sets of parameters to be the same [Goodfellow et al., 2016], or *shared*, among different neurons.

A network architecture that encapsulates this regularization technique is the convolutional neural network, which will be discussed in the next section.

The catalog of regularization techniques is very large, and there are important methods that were not mentioned here because they are outside the scope of this work, such as adversarial training.

2.3 Convolutional Neural Networks

Convolutional Neural Networks are a specialization of Deep Neural Networks where task domain constraints are used in the network architecture. The convolutional neural networks were influenced by the original model proposed by Fukushima, called *neocognitron* [Fukushima and Miyake, 1982]. Later, in 1989, Lecun implemented enhancements to the model, including gradient-based learning [LeCun et al., 1989], forming the basis for modern convolutional neural networks.

Like the feedforward networks, the Convolutional Networks are organized in layers, but they have specific constraints. The majority of the CNN architectures, especially for classification tasks, are constructed by alternating convolutional layers (Conv Layers) with pooling layers (Pool Layers). At the end of the network, some fully connected (FC) layers are usually arranged before the output layer. In the following sections, we will describe these layers.

2.3.1 Convolutional Layers

A convolutional layer takes an input tensor and performs convolutional operations using one or more filters, generating one output matrix for each filter. The output matrices are stacked together, producing an output tensor. This output can be followed by a nonlinear operation, such as ReLU.

The convolutional operations are at the center of the architecture of Convolutional Networks. A convolutional operation in an input tensor consists in multiplying a *filter* along different positions of the tensor, generating an output matrix. The operation is controlled by three variables: *filter size*, *stride* and *zero-padding*. The stride controls how the convolution *slides* in the input tensor. It defines the step size of each convolution. The zero-padding variable allows to fill the borders of the input tensor with zeroes, as it will

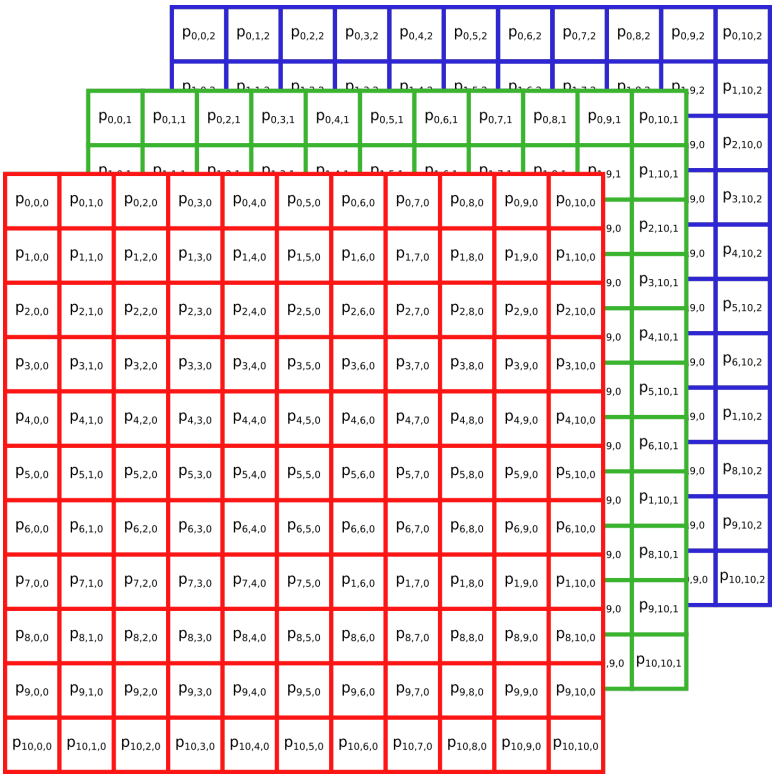

Figure 2.5: Image I with dimension 11 x 11 x 3.

be shown later. For each multiplication at a position of the input, the filter is centered on that position and each element of the filter is multiplied by the corresponding element of the input. The sum of these values is added to a bias term, optionally with a nonlinear operation, resulting in an element of the output matrix.

To illustrate the operation, we will describe a convolution in an input image I, with dimension 11 x 11 x 3: 11 pixels high, 11 pixels wide and three layers (color channels), which we will call the *depth* of our input. The filter depth must be the same as the input, and the height and width can have any value since they are not greater than the input. In this example, we will employ a filter W of dimension 5 x 5 x 3 (we could also say that the filter size is 5x5 or simply 5). The image is represented in Figure 2.5 and the filter is represented in Figure 2.6.

As the filter size is 5x5, the first position that the central element $w_{2,2,0}$ of the first layer of the filter can be aligned in the image is $p_{2,2,0}$. If we want to apply the filter to other positions in the border of the image, such as $p_{0,0,0}$, we need to fill the borders of the input with zeroes, and that is what the zero-padding variable is for. In the example, a zero-padding of 2 is enough to allow our filter to be applied at position $p_{0,0,0}$ or $p_{10,10,0}$. The latest variable that controls the operation is the stride, which tells how the filter *slides* in the input. If we use a stride of 1, after applying the filter to position $p_{0,0,0}$, the next position will be $p_{0,1,0}$. If we use a stride of 2, the next position will be $p_{0,2,0}$.

In the example illustrated in Figure 2.7, we use a stride of 2 and a zero-padding of 2. With this configuration, the convolution of the image I with the filter f produces an

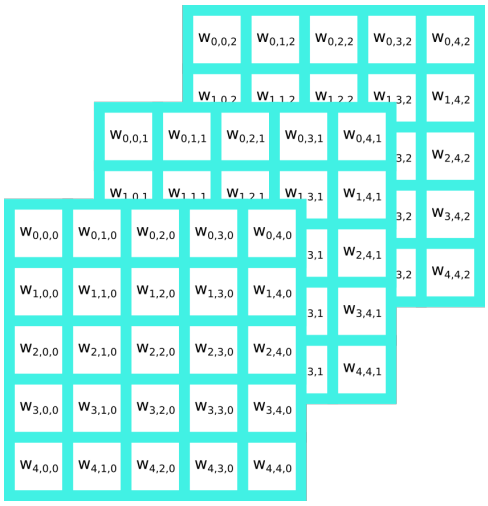

Figure 2.6: Filter with dimension 5 x 5 x 3.

output of dimension 6 x 6. In the image's red color channel, the squares with turquoise background ($p_{0,0,0}$, $p_{6,2,0}$ and $p_{8,8,0}$) are targets of the multiplication by the filter f. The arrows point the resulting neurons in S. The operation will also occur in the remaining squares with a yellow background.

In each multiplication in Figure 2.7, a target element at the first layer of the input is aligned with the central element of the first layer of the filter ($w_{2,2,0}$). These multiplications can be interpreted as a dot product, similar to Equation 2.10. If we reshape the elements of the filter W to a single column, which we will call c, and before each multiplication we do the same with the *affected* elements of the image I, which we will call x, the output of each multiplication can be calculated by Equation 2.21, with the bias term incorporated in c.

$$s_{i,j} = c^T x \tag{2.21}$$

In this way, the elements $s_{i,j}$ of the output matrix S can be viewed as neurons, considering that the convolution is usually followed by a nonlinear operation, such as ReLU. One important difference from the feedforward networks is that here all neurons share the same set of parameters W, while in the feedforward networks each neuron has its own set of parameters.

A Convolutional Layers uses several filters, generating one output matrix for each filter. Within each output matrix, the units refer to different *volumes* in the input tensor and are generated using the same filter, as shown in Figure 2.7. All along the generated output matrices, the units in the same position refer to the same *volume* in the input tensor, but they are generated by different filters. These matrices are stacked together, generating an output tensor with three dimensions, where the height and width depend on the height and width of the input tensor combined with the zero-padding and stride used in the convolution. The depth corresponds to the number of filters used.

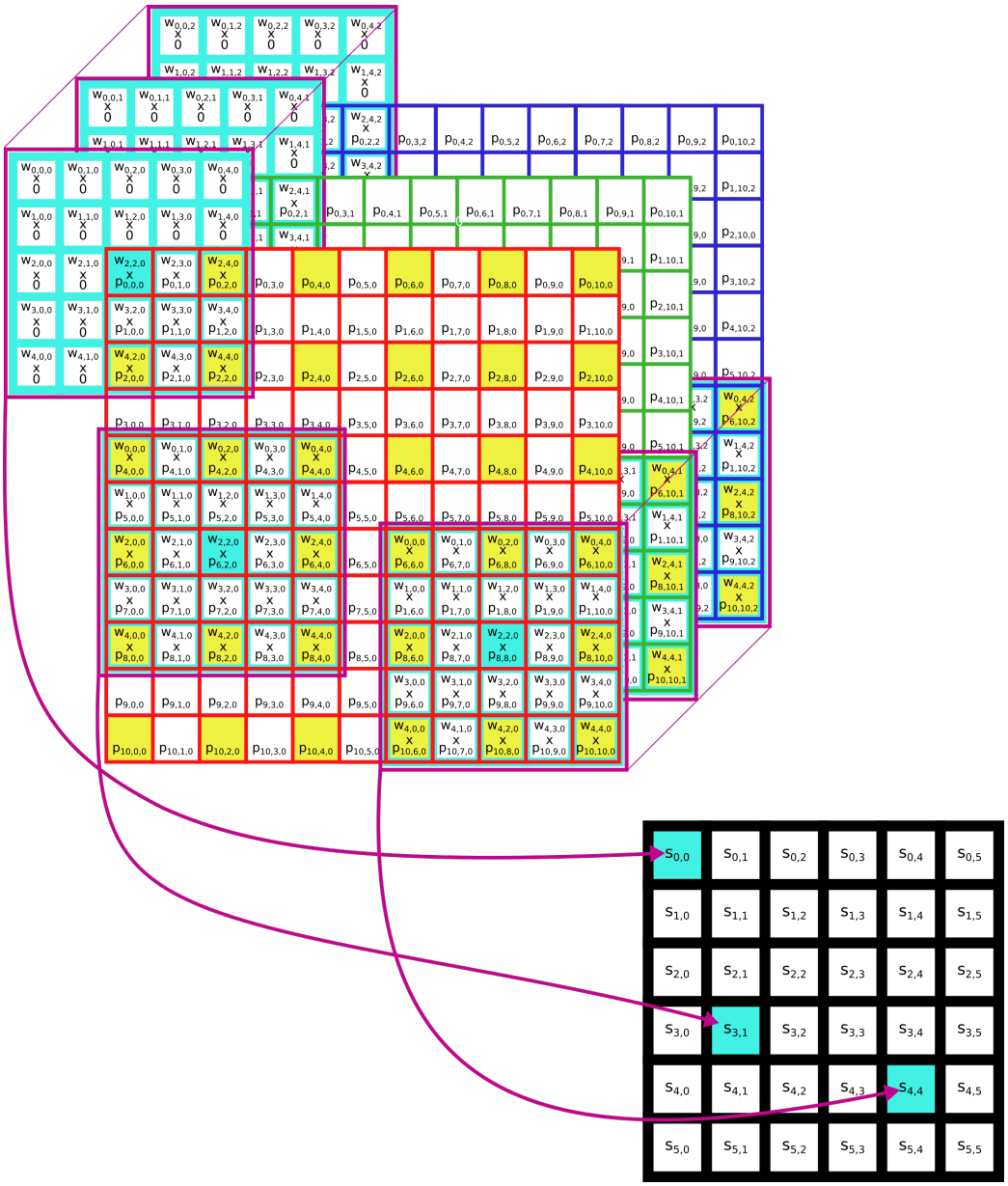

Figure 2.7: Convolution of image I by filter f using zero-padding of 2 and stride of 2.

A convolutional layer applied to the image I (Figure 2.5) using 10 filters f (Figure 2.6) with stride of 2 and zero-padding of 2 generates an output tensor of dimension 6 x 6 x 10, as shown in Figure 2.8. This tensor can be passed to other convolutional layer or to another layer in the convolutional network.

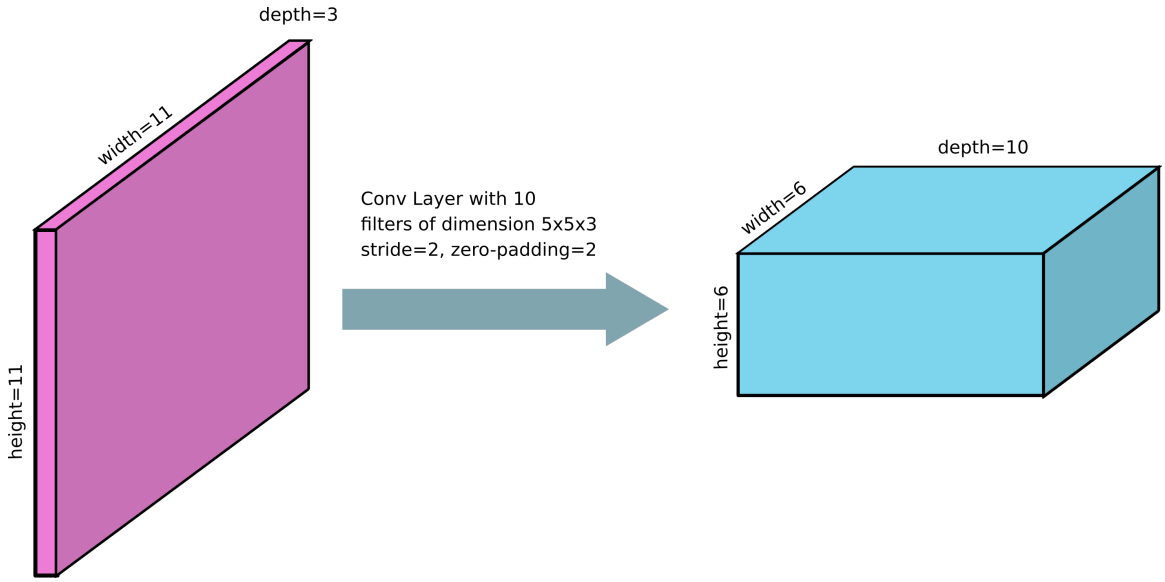

Figure 2.8: Convolutional layer using 10 filters.

2.3.2 Pooling Layers

A pooling layer takes an input tensor, typically the output of a Convolutional Layer, and applies a pooling operation to each layer along the depth of the tensor independently. It produces a downsampled output tensor with the same depth of the input tensor.

The polling operation is controlled by four variables: stride, zero-padding, window size and the pooling function. The first two work in the same way as in the convolutional layer. The window size is similar to the filter size, except that it is bidimensional since every layer of the input tensor is processed independently. The pooling function is the function used to produce the output value corresponding to the current window. Two common functions used in pooling operations are average pooling and max pooling. The main advantage of using pooling layers is to make the model invariant to small translations [Goodfellow et al., 2016].

The max pooling function selects the greatest value as the pooling window slides over each layer in the input tensor. Figure 2.9 illustrates how the max pooling works in each layer of the input tensor, generating a corresponding layer in the output tensor.

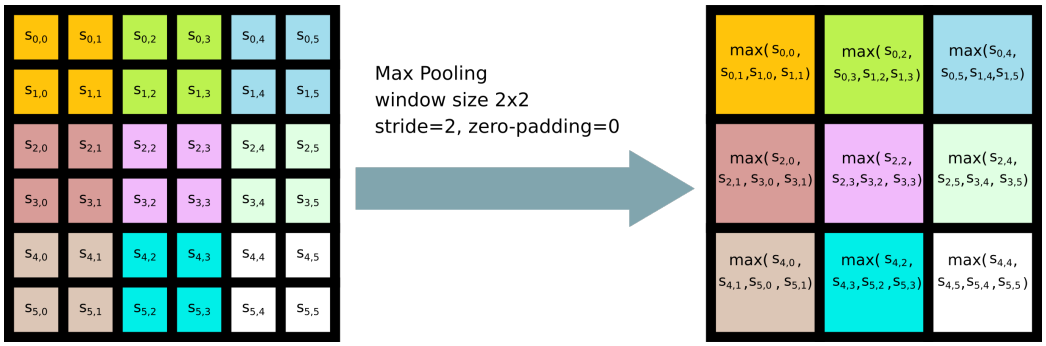

Figure 2.9: Max pooling operation in a layer (a matrix) of the input tensor.

2.3.3 Fully-connected Layers

A fully-connected is a layer with all neurons organized in a single column, just like in a feedforward network, as the hidden layers in Figure 2.3. The fully-connected layers are used at the end of the convolutional network. In many architectures, the neurons of the last pooling or convolutional layer are *flattened* to a column and the following layers before the output layer are fully-connected.

Other types of layers are possible in a convolutional network, but a full description is unfeasible and it is outside the objectives of this work.

2.3.4 The Learning Process

A Convolutional Network learns its parameters, in special the filters of the convolutional layers, with the same framework described for the feedforward networks, using a loss function, backpropagation and an optimization algorithm, like gradient descent.

In the forward pass, the convolutional operations have the same format of the feedforward networks (Equation 2.21), so the gradients can be computed using the backpropagation algorithm and the parameters can be updated according to the chosen optimization algorithm, as described in Subsection 2.2.3.

One specificity of the CNNs is the parameter sharing scheme. Each filter in the convolution produces a grid of neurons (an output matrix). The filter acts as the weights or the parameters of these neurons. In the backpropagation step, the gradients of all the neurons are used to update the weights of the corresponding filter, what is nothing more than the application of the Equation 2.18.

Chapter 3

Related Work

This chapter presents some approaches targeted for the CSAM detection task in the literature. It also covers works on age estimation from faces and pornography detection since our work aims to evaluate and design a solution for CSAM detection based on these techniques.

3.1 CSAM Classification

Due to the nature of child sexual abuse imagery, which most countries consider illegal, the development of works in this area is relatively scarce. The inaccessibility of data for training and even for testing models is one of the main factors for the small number of works developed specifically for the CSAM classification task.

The classification methods are usually built as a single-tier classifier, which directly detects CSAM, or as a multi-tier classifier, which combines auxiliary classifiers and detectors, such as a pornography detector and an age estimator, to predict child sexual abuse content.

The first works on CSAM classification were based on the statistical analysis of texture and color information. In [Polastro and Eleuterio, 2010], the authors propose a nudity detection method to assist forensic examiners in the search for child pornography images. Despite the article's title mentioning child pornography, the method makes no distinction between regular pornography and child pornography. The method's evaluation methodology reflects this aspect: the first dataset used to evaluate the technique contains 556 images without nudity and 334 images of regular pornography obtained from the internet. The second evaluation was performed on a seized hard drive, and the results included regular pornography and child pornography files in the same group.

Sae-Bae et al. [Sae-Bae et al., 2014] proposed an approach that considers skin color filters to identify human skin tones and explicit content. It also detects the faces in the image and their corresponding landmarks. Then, it estimates the face's ages using an

anthropometric approach, where an SVM classifier uses distance and texture features to classify the faces as belonging to a child or an adult.

The authors evaluated the method on a private dataset that did not contain actual CSAM due to the infeasibility of working with this kind of data. Instead, they set up a dataset of explicit-like child images comprising 105 images collected from the internet. These images involved semi-naked children in non-sexual contexts. The authors stated that they expected a true positive rate of 74.19% in detecting CSAM. That may not correspond to reality since they did not evaluate the method against actual data. The authors could have reduced this shortage by providing other valuable information about the method with tests on accessible images, such as regular pornography or images containing people without pornography.

A drawback of these approaches based on nudity detection or color and texture analysis is that they are too simplistic to capture the nuances of the CSAM classification problem. They have difficulty distinguishing CSAM from regular pornography or even images containing many people. Thus, they can lead to high rates of false positives and may not be sufficiently suitable for the objective they intend to accomplish, which is to significantly narrow down the number of images that go to visual inspection.

In [Ulges and Stahl, 2011], the authors used color-enhanced visual word features with an SVM classifier to detect CSA images. The experiments performed training and testing rounds involving 5 data sources: CSAM, Porn, Flickr, Corel, and Web. The three latter datasets comprise images without regular pornography or CSAM. The method achieved good performance in distinguishing Porn's files from those of Flickr, Corel, and Web datasets, with error rates varying from 6.0% to 9.7%. The error rates increased in the discrimination of CSAM's files from those of the three latter datasets, reaching an error rate of 21.8% against Flickr images. The error rate was even worse when distinguishing CSAM's files from Porn's files, reaching 24.0%. Nonetheless, a distinctive aspect of the work was comparing the method's behavior in different scenarios.

In [Yiallourou et al., 2017], the authors set up a dataset with non-illegal images collected from the internet showing possibly suspicious content. Then they performed a content-based analysis of the images and determined features associated with the suspicious images. Some volunteers evaluated the images and determined each image's inappropriateness level (IL) using a scale from 1 to 5. They also indicated which features influenced their classification. Finally, the authors used those results to create a synthetic dataset incorporating such features and proposed an approach to detect CSAM based on those features.

The proposed method first detects faces using Haar feature-based cascade classifiers. Then it estimates the age and gender of the detected faces and calculates the lighting intensity of the image. Combining these steps yields five image features that denote child presence, number of people, age diversity, gender distribution, and lighting

intensity. These features were used to train a linear regression model to predict the inappropriateness of the images. They evaluated the model with 200 images downloaded from the internet, which had the inappropriateness level rated by three volunteers, using a scale from 1 to 5. The method achieved a mean absolute error of 0.9974. A downside of the experiment was not using actual CSAM images for testing and not evaluating the method's behavior on images of regular pornography or images with people without pornography.

In [Vitorino et al., 2018], the authors developed a method that uses a CNN to predict CSA from images directly, i.e., without using a combination of auxiliary predictors.Um aspecto relevante The proposed method uses a GoogLeNet architecture [Szegedy et al., 2015], pre-trained in the ImageNet classification task for 1000 classes. The network is then adapted to the new task, including two new auxiliary classifiers. The best model, referred a *2-tiered SEIC Detector* is first fine-tuned in a dataset of pornographic and non-pornographic content, to *learn* the adult content classification task, and then the is fine-tuned in a dataset of sexually exploitative imagery of children (SEIC), where the images are tagged as related to SEIC or not. After the network converges, the last layer is replaced by an SVM classifier using a radial basis function (RBF) kernel.

The work distinguishes normal images from adult pornographic content and from child pornographic content, indicating levels of pornography. As far as we are concerned, this is the unique method that uses CNNs to predict CSAM, probably because of the illegal nature of the data, which implies that there are no datasets available for training. These datasets in general only exist in police agencies, such as the Child Exploitation Obscene Reference File [NIST, 2017], maintained by the FBI.

A relevant aspect of these last referenced works is that they exploit models with greater complexity and capacity. Such models can result in more refined classifications when compared to models based on texture and skin tones. With this improvement, it becomes possible to achieve one of the main challenges in this area of research: the differentiation of CSAM images from standard pornographic material or images containing children without sexual connotations.

3.2 Pornography detection

The first works on pornography detection were primarily based on the statistical classification of texture and color information [Lin et al., 2003; Ap-Apid, 2005; Sathish and Sengamedu, 2008]. The main disadvantage of these methods is the high false-positive rate due to the similarity between human skin color patterns and the color of other existent

things in a scene.

Some approaches based on bag-of-visual-words (BoVW) to the pornography detection task [Deselaers et al., 2008] were proposed to overcome the disadvantages of skin-based methods. In [Wang et al., 2009], a Support Vector Machine (SVM) classifier [Cortes and Vapnik, 1995] based on local features, such as SIFT (Scale Invariant Feature Transform) [Lowe, 2004] visual words is used to classify pornographic content.

Some more sophisticated works were developed, such as in [Caetano et al., 2014], where the authors introduced a descriptor to detect pornographic content in videos. Instead of using image descriptors in the search for video content, the authors developed a video descriptor that combines local binary descriptors with a mid-level representation called *BossaNova*.

More recently, some approaches based on convolutional networks were proposed, achieving state-of-the-art results in the pornographic image and video classification tasks.

In [Nian et al., 2016], the authors introduced a convolutional neural network architecture to detect pornographic content in images. The model is pre-trained in the ImageNet challenge dataset with 1000 classes and fine-tuned in a private dataset containing pornographic and non-pornographic images.

In 2016, Yahoo! open-sourced a CNN architecture to classify images as *suitable/safe for work* or not. The method is based on the ResNet-50 network, using half number of filters in each layer. The network was pre-trained in the ImageNet challenge dataset, adapted to the target task and fine-tuned with a proprietary and not publicly available dataset of positive (not suitable/safe for work, or NSFW) images and negative (suitable/safe for work, or SFW) images [Mahadeokar and Pesavento, 2016]. The output of the network is a number between 0.0 and 1.0 that can be viewed as the probability that the image is pornographic.

Convolutional neural network based models were also proposed to detect pornography in videos. In [Wehrmann et al., 2018], the authors propose a method that combines convolutional neural networks and long short-term memory (LSTM) recurrent networks [Hochreiter and Schmidhuber, 1997] to detect pornographic content in videos. In [Perez et al., 2017], a distinct method is proposed. Unlike the traditional frame-wise approaches that depend on the classification of individual frames to determine the class of the video, the authors introduce a method that exploits the combination of static and motion information using optical flow [Brox et al., 2004] and MPEG motion vectors [Richardson, 2004] to detect pornography in video files.

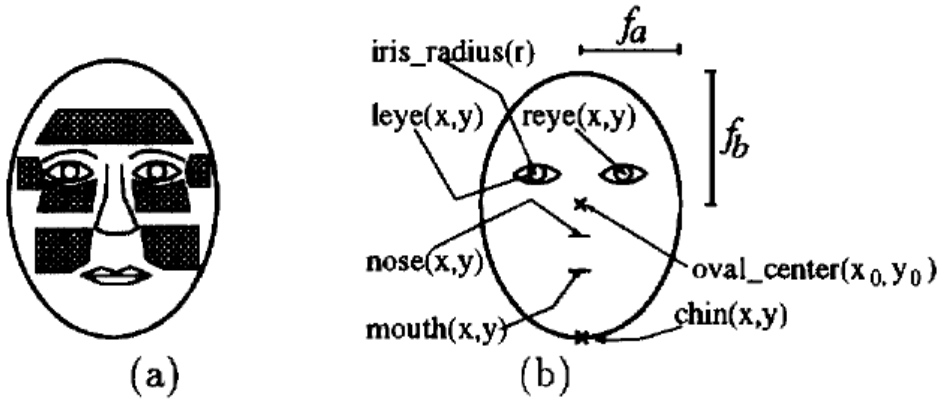

Figure 3.1: Wrinkle geography (a) and face template (b) [Kwon and Lobo, 1994].

3.3 Age Estimation from Face Images

Since the pioneering works in age estimation from face images, many different methods and techniques have been proposed to target the problem. The approaches adopted in those methods somehow reflect the advances of the visual pattern recognition techniques.

Anthropometric Approaches: the first attempts to the age estimation problem were mainly based on image processing techniques aimed at extracting visual information such as facial landmarks and its ratios and proportions, which were directly used to predict the age. The method proposed in [Kwon and Lobo, 1993] is based on the craniofacial growth development theory, that is used to distinguish babies from adults, and on wrinkle analysis, which is used to evaluate the skin aging process in order to distinguish adults from senior adults.

These techniques are referred to as anthropometric approaches to age estimation. Figure 3.1 illustrates the face regions used to search for wrinkles and the face template used in the oval and eye-fitting processes.

Active Appearance Models: instead of searching for local features, such as geometric proportions and wrinkles, part of the research community started to apply efforts to develop matching techniques based on models. These techniques usually adopt a statistical approach to represent the face through dimensionality reduction and in a second step apply regression techniques to perform age estimation. The active appearance model (AAM) is one of these techniques, which tries to understand new images based on similar synthetic images, using a parametrized model of appearance [Cootes et al., 1995]. The method proposed in [Lanitis et al., 2004] generates a statistical model of facial appearance based on the face images and performs Principal Component Analysis (PCA)

[Jolliffe, 2011] for dimensionality reduction. The representation of faces is then directly used for age estimation using different classifiers.

In the AGES (AGing pattErn Subspace) [Geng et al., 2006], the method aims at building aging patterns of different people in order to model a representative subspace of aging patterns.

Bio-Inspired Features: in the bio-inspired features methods [Guo et al., 2009], the age estimation method is based on an object recognition technique that models the visual processing in the cortex proposed in [Riesenhuber and Poggio, 1999] and later extended in [Serre et al., 2007].

The original method can be described as a feed-forward neural network with four simple and complex alternated layers, called S_1, C_1, S_2 and C_2. These layers create an increasing complexity from the input layer to the output layer with a nonlinear maximum operation MAX over the S units. In the extended method, the last layers are modified to perform a template matching approach, which has the advantage of being highly selective and position invariant.

In the adaptation of the method to the specific task of age estimation, the layers S_2 and C_2 are discarded because they do not work well in the domain of the age estimation task. Besides that, the output of the C_1 units is concatenated to form a long representation of the face image, which is later reduced through PCA. The method is tested against the FG-NET [FGnet, 2002] and the Yamaha Gender and Age dataset (YGA). In the first one, it achieves a mean absolute error (MAE) of 4.77, using a leave-one-person-out cross-validation strategy. In the second dataset, the method achieves a MAE of 3.91 for females and 3.47 for males, using a 4-fold cross-validation setup.

General Purpose Descriptors: some other methods make use of general purpose descriptors, such as Local Binary Patterns (LBP) [Ojala et al., 1996] and Histograms of Oriented Gradients (HOG) [Dalal and Triggs, 2005], and then choose one machine learning algorithm, such as SVM for training a classifier with the extracted features.

Some general purpose descriptors, like LBP and its variants, have long been used for many face analysis related tasks [Huang et al., 2011], including age estimation. In [Yang and Ai, 2007], for example, an LBP variant, called LBPH, is used as the main descriptor to train an AdaBoost [Freund et al., 1999] classifier for age estimation and for gender and ethnicity classification. For the first task, a snapshot dataset of 9000 Chinese face images is used for cross-validation and the datasets FERET [Phillips et al., 2000] and PIE [Sim et al., 2003] are used for testing. The proposed method with better results achieved an error rate of 6.7% and of 8.9% for the FERET and PIE datasets, respectively.

Deep ConvNets Approaches: recently, some researchers have proposed deep-learning based approaches. The first work to use a Deep ConvNet architecture for the age estimation from face images task was [Huerta et al., 2015], where a fusion of well-known descriptors was compared to a Deep ConvNet approach. The authors used the LeNet CNN [Lecun et al., 1998] as the base architecture for the proposed network and modified its parameters, the number of convolution layers and fully connected layers to create a new architecture, better suited for the age estimation task. The best architecture achieved a MAE of 3.88 for the MORPH dataset [Ricanek and Tesafaye, 2006] and of 3.31 for the FRGC dataset [Phillips et al., 2005].

Levi and Hassner [Levi and Hassncer, 2015] proposed a small CNN to predict age and gender using the Adience dataset [Eidinger et al., 2014], achieving an accuracy of 50.7 $\pm$ 5.1 using a 5-fold cross-validation protocol. Duan et al. [Duan et al., 2018] combined a CNN and an extreme learning machine to perform age and gender classification using the Morph-II and Adience benchmarks, achieving an accuracy of 52.3 $\pm$ 5.7 in the latter one.

In [Chen et al., 2017], the authors exploited the ordinal relation between ages, proposing an architecture that consists of a series of CNNs, one for each age or age group, where each CNN yields a binary output that tells if the face's age is higher or lower than a certain value. All the binary outputs are aggregated to make the final age prediction. Age order information is also exploited in [Niu et al., 2016]. In [Liu et al., 2018], a label-sensitive deep metric learning method was proposed, using a deep residual network to learn distances between ages.

Ranjan et al. [Ranjan et al., 2017b] extended a previous work [Ranjan et al., 2017a] and designed a multi-purpose CNN model that performs face detection, landmarks localization, pose estimation, gender recognition, smile detection and age estimation, exploring the idea of joint learning correlated tasks, sharing parameters in the lower layers.

In [Rothe et al., 2018], the IMDB-WIKI dataset is introduced, containing 523,051 face images of celebrities, 460,723 from IMDb[1] and 60,723 from Wikipedia[2], labeled with real age and gender. The work also proposed an age estimation method based on a VGG-16 [Simonyan and Zisserman, 2014] CNN pre-trained on the ImageNet challenge dataset, which is fine-tuned on the IMDB-WIKI dataset and later fine-tuned on the Adience dataset, achieving an accuracy of 64.0 $\pm$ 4.2. The model was also evaluated without fine-tuning on the IMDB-WIKI dataset and achieved an accuracy of 55.6 $\pm$ 6.1.

In [Zhang et al., 2017], a CNN method model is proposed using Residual Networks of Residual Networks (RoR) [Zhang et al., 2018], that allegedly present better optimization ability than conventional CNNs for the age and gender estimation tasks. The model is

[1]http://www.imdb.com
[2]http://www.wikipedia.org

also pre-trained on the ImageNet classification task, fine-tuned on the IMDB-WIKI-101 data set and later fine-tuned on the Adience dataset, achieving an accuracy of 67.34 ± 3.56.

The methods based on CNNs have a better generalization than the previously developed methods, making them more suitable for use in uncontrolled scenarios since they do not depend on faces in a frontal and aligned position. The intrinsic characteristics of automatic feature learning provided by CNNs explain this improvement in part. But these methods also benefited from the larger and more complex training datasets that this technique can exploit.

Chapter 4

Methodology

In this chapter, we present the proposed approach for CSAM detection and the benchmark dataset to assess and compare child sexual abuse classifiers.

4.1 Proposed Method

The proposed method combines a pornography classifier, a face detector, and an age estimator to determine whether an image is related to child pornography. The face detector and the age estimator are used to find child faces in the image and they can be interpreted as a *child face detector*. The combined approach assumes that if the pornography classifier indicates that an input image is pornographic and the child face detector finds a child's face, the image must be classified as related to CSAM. Otherwise, if the output of any auxiliary method is negative, the method concludes that the image does not contain CSAM.

Figure 4.1 illustrates the pipeline of the proposed approach, presenting an example of how an image with two adults showing their faces would be processed. The pipeline highlights the auxiliary methods that comprise the proposed approach. In the first step, a pornography classifier applied to the image generates a value between 0 and 1 that indicates the probability that the image contains pornographic content.

The proposed method uses a threshold τ to identify whether the predicted probability indicates pornography in the image: the image is considered not pornographic if the probability is lower than the threshold, and pornographic otherwise.

If the predicted pornography probability of the image is lower than τ, the execution will finish at this point with a negative answer to child sexual abuse content. If the predicted pornography probability is equal or greater than τ, the child detection module represented by the face detection and the age estimation methods will be executed. In the face detection method, the faces are detected, extracted, aligned and submitted to the age estimation method. In the example depicted on Figure 4.1, the method would

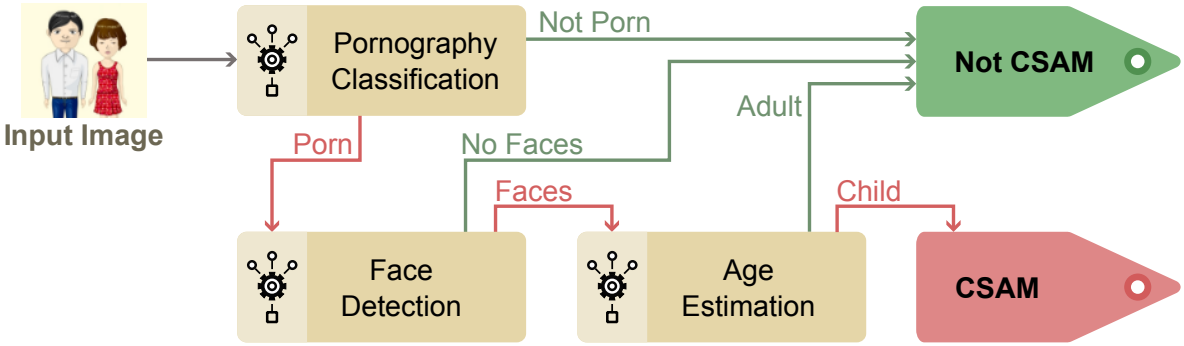

Figure 4.1: Overview of the proposed methodology for CSAM classification. The method firstly detects if an image has pornographic content. Then, the method detects the faces in the image and estimates its corresponding ages. Finally, we perform a classification to identify whether an image contains CSAM. The proposed method is based on two hypothesis: a) many child sexual abuse images and videos include the face of the victims; and b) pornography detection is an easier task than CSAM detection and assumes the hypothesis that a CSAM detector can be built through the combined analysis of a pornography detector and an age estimator.

report two adults in the image, leading to a negative answer to the CSAM classification task.

In the following subsections, the pornography classifier, the face detector and the age estimation method will be described in more detail, followed by a description of how they are combined to perform the CSAM classification task.

4.1.1 Face Detection

For the face detection task of the pipeline, the proposed method uses the MTCNN face detector [Zhang et al., 2016], that employs a cascaded architecture consisting of three CNNs, each one responsible for executing one of the three consecutive steps of the face detection method. In the first step, one CNN predicts potential faces, in the second step, the initial detection is refined, and in the last step, the third CNN further refines the results and adds faces landmarks.

The MTCNN face detector returns the coordinates of the bounding boxes of the detected faces in an image. For each detected face, the method also provides a set of landmarks, with the location of the eyes, nose and mouth.

A pre-processing step that is shared among many age estimation methods is the face alignment [Fu et al., 2010; Guo and Mu, 2014; Chen et al., 2017], which is also adopted in the proposed age estimation model, described in the next subsection. This technique is

used to improve the performance of the classification task. For this reason, after detecting the faces in an image, we use the eye landmarks returned by the face detector to perform the alignment of each face. The aligned faces are used either in the fine-tuning step of the age estimation method as in test, where the method is evaluated with new images.

4.1.2 Age Estimation

The age estimation task that integrates the CSAM classification method is performed by a convolutional neural network adapted for this work, fine-tuned and evaluated on the Adience dataset [Eidinger et al., 2014] using a five-fold cross-validation protocol.

The Adience dataset contains real-world unconstrained images with faces classified by age groups ($0-2$, $4-6$, $8-13$, $15-20$, $25-32$, $38-43$, $48-53$, $60-$) and by gender. There is no label for child and adult, and this information was inferred by the age groups. For the purposes of this work, we considered the three first age groups as child ($0-2$, $4-6$, $8-13$) and the other five age groups as adult.

Due to the small number of samples in the target dataset, we started with a VGG-16 [Simonyan and Zisserman, 2014] architecture pre-trained for the ImageNet dataset [Russakovsky et al., 2015] to avoid over-fitting and to accelerate the learning process.

The three last fully connected layers were replaced by two fully connected layers with 4096 channels followed by three soft-max layers which are used in the multi-task learning strategy [Caruana, 2012] to simultaneously learn age groups classification, child detection and gender classification. The overview of the architecture, showing the three final prediction layers, is depicted in Figure 4.2.

4.1.3 Pornography Classification

For the pornography classification task of the pipeline, we used a recently published adult content classification network open sourced by Yahoo [Mahadeokar and Pesavento, 2016]. The network is based on the ResNet-50 architecture [He et al., 2016] with half the number of filters in each layer. It was pre-trained with ImageNet dataset and fine-tuned with a proprietary not suitable/safe for work (NSFW) dataset.

For a given input image, the pornography classifier outputs a number between 0.0 and 1.0, that can be viewed as the probability that the image is pornographic. When

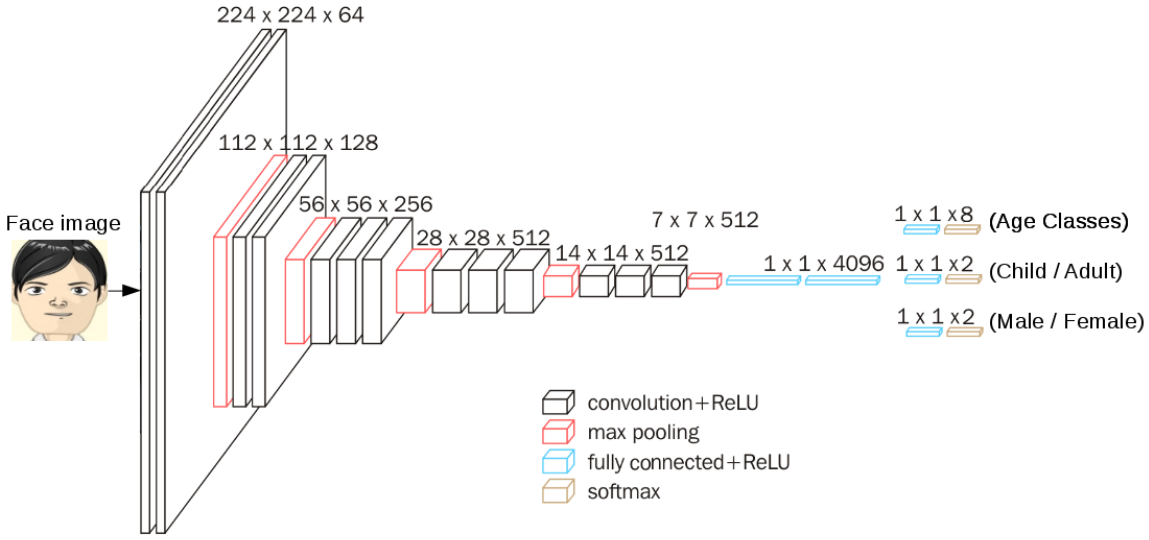

Figure 4.2: VGG-16 adapted architecture.

the pornography classifier is used in the proposed method, we need to specify what is a pornographic image. In the context of this work, we consider that an image belongs to the *pornography* class if it has nude or sexual content. We also defined a broader category called *seminude+*, which comprises images containing seminude, nude or sexual content. The threshold values for these two categories (pornography and seminude+) were experimentally chosen using a validation dataset containing images labeled as pornographic, seminude and others.

4.1.4 CSAM Classification

Putting it all together, the pornography classification method combined with a face detector and an age estimator builds up a model to classify images as related or not to child pornography.

A step-wise illustration of the proposed approach with the analysis of an image is provided in Figure 4.3. In the first step, the pornography classifier determines the pornographic content of the image. Assuming that the image is pornographic and that its class is correctly predicted by the pornography classifier, the second step performs the detection of the faces of the image, which are extracted and aligned. In the third sted, the faces are fed to the age estimator, which determines if the image belongs to a child of to an adult. The last step is the output, where the information of the auxiliary classifiers are used to determine if the image is related or not to child pornography. In the example, assuming that the face is classified as Adult, the conclusion is negative.

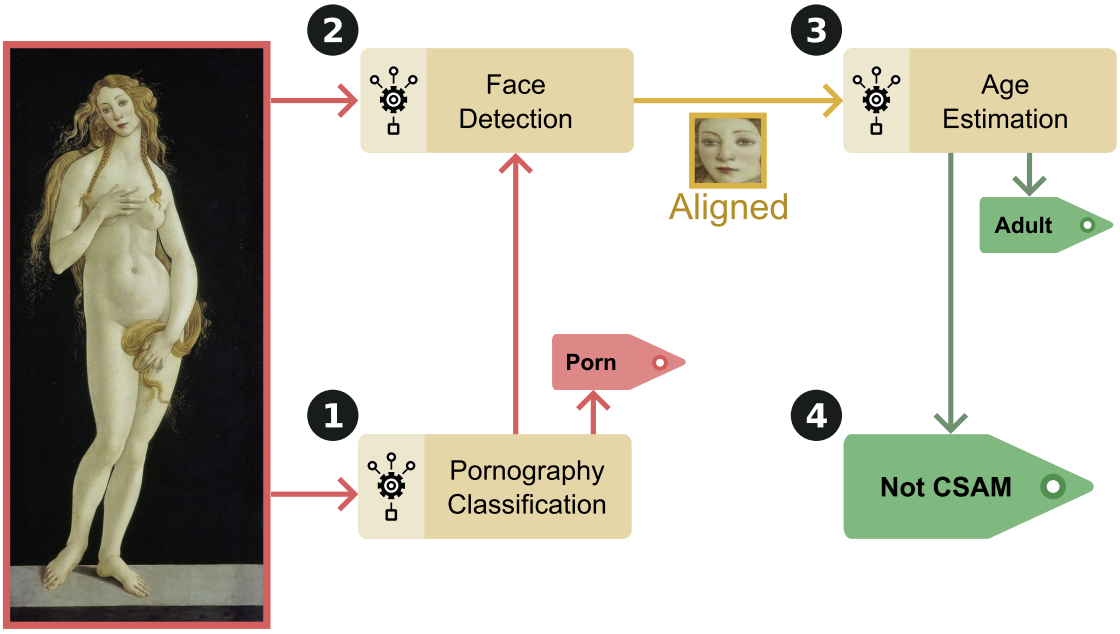

Figure 4.3: CSAM classification example. Results from pornography classification, face detection and age estimation methods are combined to detect CSAM.

4.2 Region-based Annotated Child Pornography Dataset

The region-based annotated child pornography dataset (RCPD) [Macedo et al., 2018] was created in collaboration with the Brazilian Federal Police. The child sexual abuse images within it were gathered and labeled internally the Brazilian Federal Police and its contents cannot be illustrated in this work or made public due to its illicit nature.

The aim of the dataset is to assess and compare the performance of CSAM detection methods, to boost the development of new approaches to this important application of the forensic field. Researchers may submit their algorithms following the instructions provided at `http://www.patreo.dcc.ufmg.br/rcpd`.

The dataset consists of 2138 images, including 508 images with no people and 1630 images containing individuals showing or not the face. The dataset includes a reasonable number of images not containing any person to allow the evaluation of false positive rates.

4.2.1 Usage of the Dataset

Each image of the RCPD dataset has one of the following labels: normal, adult seminude, adult pornography, child seminude or CSAM. When a method is evaluated against the dataset for CSAM detection, it must *detect* the images of the child sexual abuse class, which are the images that contain at least one child and have nude or sexual content, not necessarily associated with the child. The method must classify the CSA images as positive and the others as negative.

The dataset can be used to evaluate a method for the child seminude+ classification task, in which the method must detect the images of the child seminude or CSA classes. In this case, the method must classify the images containing child seminude or CSA as positive and the others as negative.

Each image in the dataset may contain no person or it may contain one or more people. Each person may have multiple labels, such as age, gender and nudity exposure (no nude, seminude, nude or sex). The face and relevant parts of the individuals are also annotated with its respective region in the images of the adult seminude, adult pornography, child seminude or CSAM classes. So it is possible to know if there is a breast in the image, where it is located and the gender and age group of the individual to whom it belongs. This information can be used to evaluate a method that detects certain parts of the human body. For example, if a breast detector is evaluated in the RCPD dataset, it must detect each image that contains a breast or even inform its location in the image. If the detector is targeted to a specific age range and/or to a specific gender, the dataset information can still be used to evaluate it.

4.2.2 Structure of the Dataset

From the image labels, it is possible to answer many queries, such as if there is any child in the image, or even more complex queries, such as if the image has nude or sexual content and a child showing her face.

Figure 4.4 shows the overlapping labels of the images in the dataset, where it is possible to verify that some images contain faces, while others contain children, nude or sex, and so on. And these categories can be present in a single image, creating many possibilities in the dataset.

The dataset's total number of images and the number of images in the two major categories are shown in Table 4.1. The numbers returned by some queries of interest to

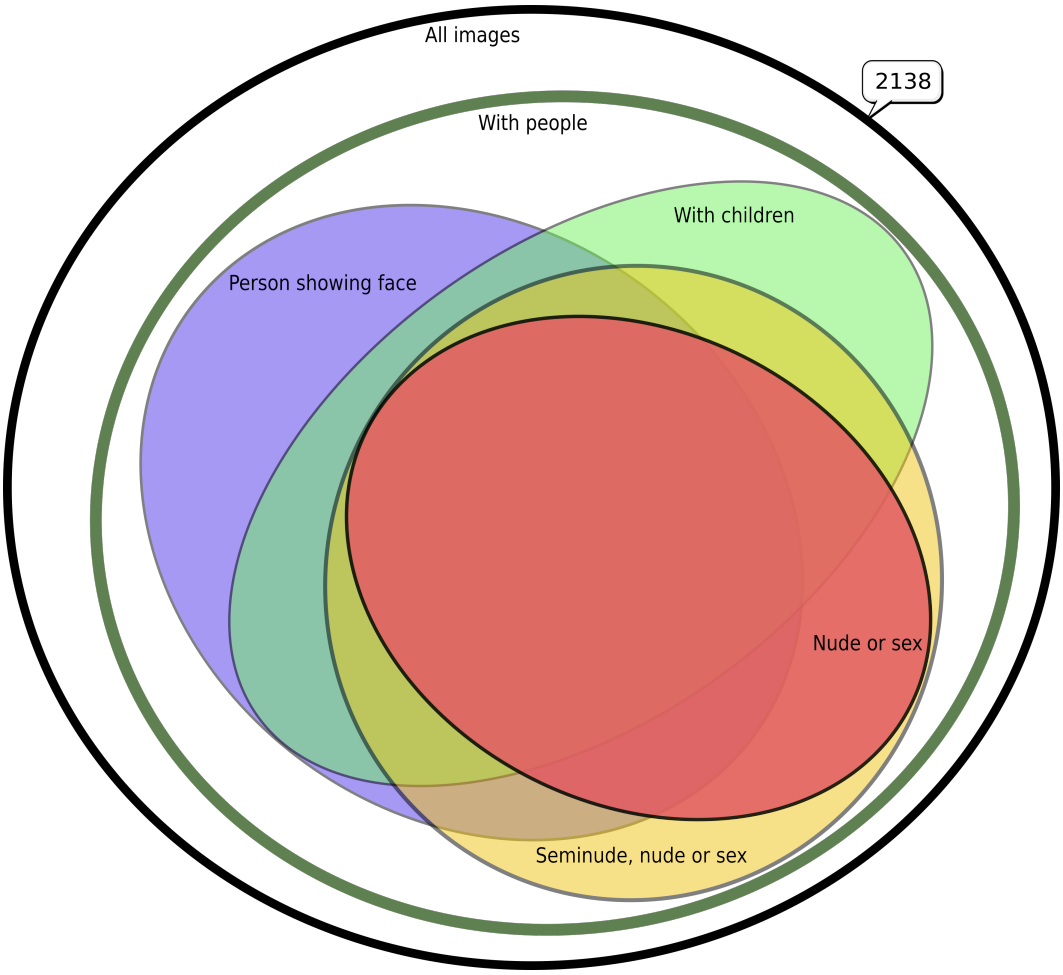

Figure 4.4: The images in the dataset cover a wide range of situations.

the dataset's objectives are shown in the nine last lines of the table.

Table 4.1: Images in dataset.

Category	Number of images
All	2138
No person	508
Person	1630
Person showing face	1455
Child	1238
Child face	1065
Seminude, nude or sex	1407
Face + seminude, nude or sex	1233
Child + seminude, nude or sex	1051
Child face + seminude, nude or sex	879
Child + nude or sex	836
Child face + nude or sex	664

Table 4.2: Parts annotations in the dataset.

Age	Individuals	Faces	Breasts	Privates	All
0-2	91	71	7	82	251
4-6	210	195	42	101	548
8-13	963	923	309	383	2578
15-20	232	226	161	78	697
25-32	186	180	130	111	607
38-43	67	55	42	58	222
48-53	25	23	11	14	73
60-	46	46	22	21	135
Total	1820	1719	724	848	5111

The age ranges in the dataset correspond to the age ranges used in the Adience dataset [Eidinger et al., 2014]: $0-2$, $4-6$, $8-13$, $15-20$, $25-32$, $38-43$, $48-53$ and $60-$. Tables 4.3 and 4.4 illustrate how the nudity categories in the dataset's images are distributed by the youngest person in the images and by the youngest shown face in the images, respectively.

Table 4.3: Nudity exposure in age groups.

Category	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-	All
Nonude	9	21	157	22	5	1	1	7	223
Seminude	5	25	185	46	16	4	5	2	288
Nude	80	128	454	100	108	25	8	9	912
Sex	44	51	78	8	14	4	0	8	207
Total	138	225	874	176	143	34	14	26	1630

Table 4.4: Nudity exposure in age groups by face.

Category	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-	All
Nonude	9	20	157	22	5	1	1	7	222
Seminude	5	25	185	46	16	4	5	2	288
Nude	42	100	404	98	107	26	8	10	795
Sex	10	35	73	8	12	4	0	8	150
Total	66	180	819	174	140	35	14	27	1455

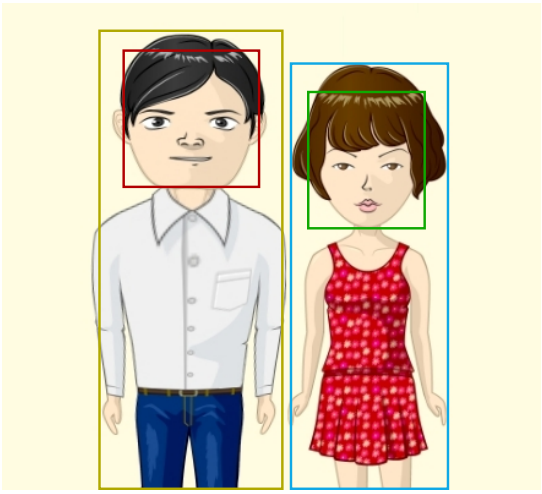

Figure 4.5: Illustration of region-based annotations of the bodies and faces of a man and a woman.

4.2.3 Region-based Annotations

The bounding boxes of the individuals and their parts are annotated, totaling 5111 objects in the whole dataset. The annotated regions are the body, face, breast and private parts.

Figure 4.5 presents a scheme of the annotation process of two people and their faces. The annotation indicates the bounding boxes or regions of a 25-year-old man and of a 20-year-old woman (none of them with the seminude, nude or sex tag), and two smaller bounding boxes of their faces. With this information, it is possible to infer that the image has no seminude or nude person, that the younger person depicted has 20 years old and that the younger face depicted belongs to a 20-year-old.

In Figure 4.6, the annotation indicates a bounding box of a 25-year-old man with the *nude* tag, represented in the figure by the greater rectangle, and the bounding boxes of his face, breast and privates.

The choice over the design of the labels allows assessing CSAM classification methods that perform whole image classification and methods that combine facial age estimation with pornography detection. It is also possible to assess methods designed to detect sensitive body parts.

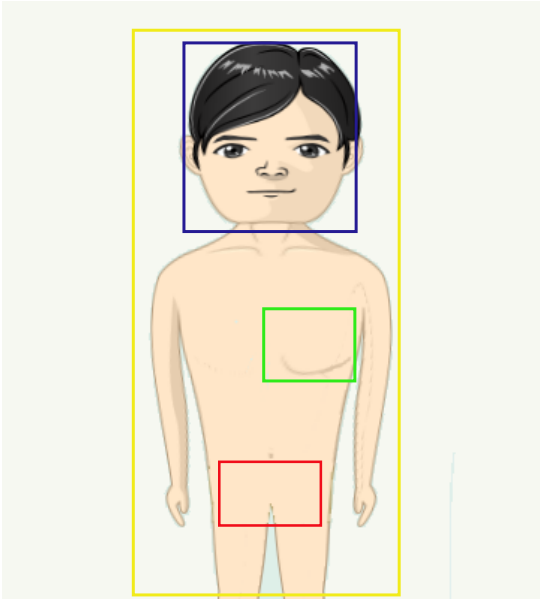

Figure 4.6: Annotation representation for nude parts of an image. The dataset has bounding boxes for the face, breast and genitals.

Chapter 5

Experiments

This chapter describes the setup of the auxiliary methods used in the CSAM classification technique and the experiments of the proposed method on the RCPD dataset introduced in this work.

5.1 Auxiliary Methods

The proposed approach combines auxiliary methods to perform the CSAM classification task. Each of these methods has its own specificities and configurations, which will be described in the next subsections.

5.1.1 Face Detection

The child face detection is composed of a face detection method and an age estimation method. For the face detection task, the MTCNN face detector [Zhang et al., 2016] was chosen. This detector was compared to other methods commonly used for this task, such as Haar Feature-based Cascade face detection and Dlib [King, 2009], and it was chosen for its better detection rate and good performance.

As it was described in the last chapter, the MTCNN detector informs the position of the face in the image without alignment, which is done manually using landmarks provided by the detector. An example of the alignment of two faces from the Adience dataset is illustrated in figure 5.1. The first column contains the original image. The second shows the bounding box of the detected face and the landmarks. The third column shows the aligned image and the last column shows the aligned image of the face.

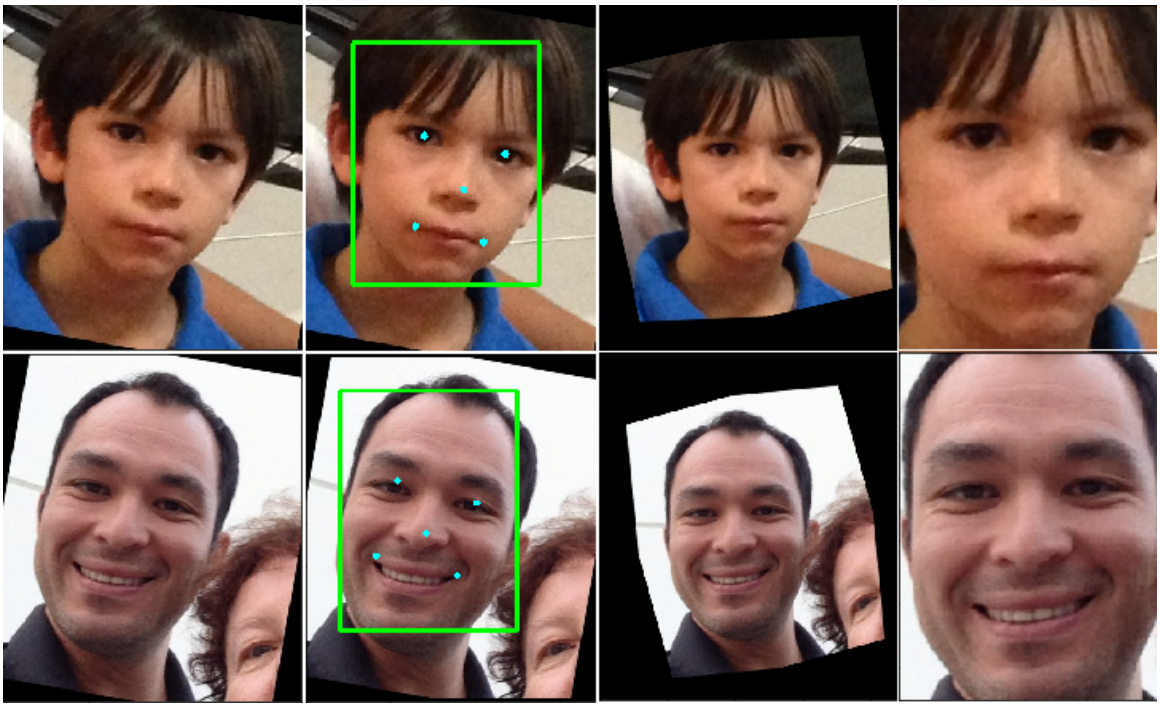

Figure 5.1: Face detection using the MTCNN face detector and the alignment process.

5.1.2 Age Estimation

For the age estimation task, a model was specifically trained in this work to fit with the chosen face detection method. We have adapted a CNN pre-trained on the ImageNet challenge dataset to simultaneously learn the age groups, child detection and gender classification. The model was fine-tuned on the Adience dataset [Eidinger et al., 2014].

The same pre-processing steps used to extract and align the faces to fine-tune the age estimation model are used to evaluate the method on the unseen images of the RCPD dataset. The MTCNN face detector was used to extract the faces from the Adience dataset and a data augmentation strategy based on flipping and slightly rotating the faces was employed, aiming at avoiding over-fitting and improving the classification performance.

The results of the adapted network for the age estimation task are presented in Table 5.1. Although the adapted network accuracy is below the state-of-art results, it should be noticed that the above 60.0% accuracy methods perform an extra fine-tuning step using the IMDB-WIKI-101 dataset [Rothe et al., 2018; Zhang et al., 2017; Hou et al., 2017], which is the largest in the wild face images dataset up to date, with more than 500.000 images. It is important to notice that for child/adult classification, which is the main interest for this work, the adapted model achieved an accuracy of 94.1 $\pm$ 2.3.

Table 5.1: Age estimation results on the Adience benchmark.

Method	Accuracy ± Std. Dev. (%)
[Eidinger et al., 2014]	45.1 ± 2.6
[Levi and Hassncer, 2015]	50.7 ± 5.1
[Rothe et al., 2018] w/o IMDB-WIKI pretrain	55.6 ± 6.1
[Rothe et al., 2018] w/ IMDB-WIKI pretrain	64.0 ± 4.2
[Hou et al., 2017] w/ IMDB-WIKI pretrain	67.3
[Zhang et al., 2017] w/ IMDB-WIKI pretrain	67.3 ± 3.6
Proposed method	56.8 ± 6.0

5.1.3 Pornography Classification

The pornography classification method used in the proposed method outputs the probability that one image is pornographic. To determine the value from which an image should be classified as pornography (containing nude or sex) or seminude+ (containing seminude, nude or sex), we experimentally chose the threshold values for the two categories using a validation dataset containing images labeled as pornographic, seminude+ and others.

Figure 5.2 shows the accuracy, precision and recall rates for different threshold values for the pornography classification task. We compared candidate threshold values in the graph and chose the threshold $\tau = 0.3$, which had similar accuracy to the threshold 0.35, but presented a better recall rate.

Figure 5.3 shows the accuracy, precision and recall rates for different threshold values for the seminude+ classification task. Similarly, compared candidate threshold values in the graph and chose the threshold $\tau = 0.1$.

5.2 Proposed Approach

The proposed method was evaluated on the RCPD dataset described in this work. The accuracy in CSAM detection on the RCPD dataset with this configuration ($\tau = 0.3$) was 79.84%. The auxiliary classification models used in the pipeline had an accuracy of 82.55% for the child detection task and 85.78% for the pornography (nude or sex) classification task step, as shown in Table 5.2.

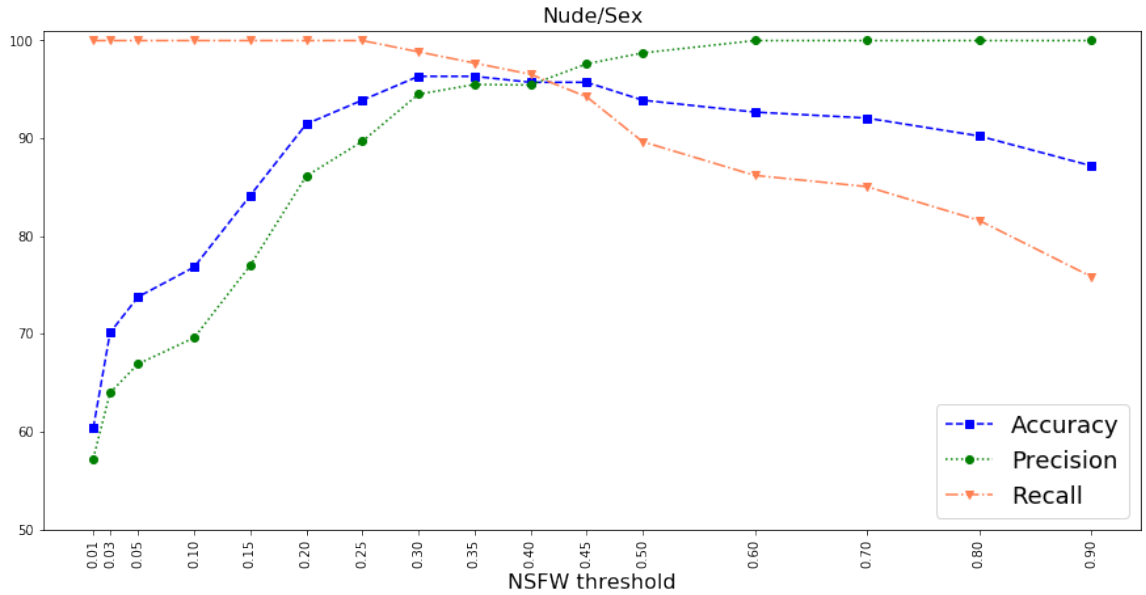

Figure 5.2: Accuracy, precision and recall x thresholds for the pornography classification task.

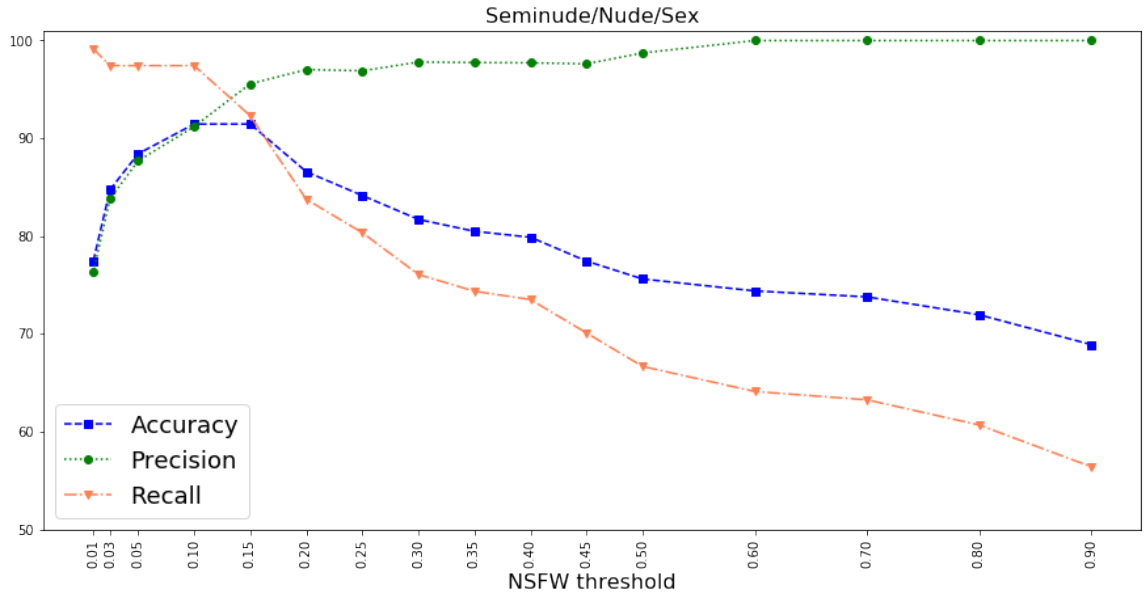

Figure 5.3: Accuracy, precision and recall x thresholds for the pornography classification task.

We evaluated the proposed method to perform a looser classification to also identify seminude content, besides nude and sexual content. This configuration uses a lower threshold ($\tau = 0.1$) for the NSFW classifier and targets child related images containing seminude, nude or sexual content (seminude+). This kind of broader classification can be of interest in specific forensic scenarios where it may be necessary to care about seminude pictures involving children. The results are shown in Table 5.3.

Two forensic tools from the Brazilian Federal Police were evaluated against the RCPD dataset. The first of them was a tool named NuDetective [Polastro and Eleuterio,

Table 5.2: CSAM Detection (Nude and Sex).

Metric	CSAM	Child Detection	Porn Detection
Accuracy (%)	79.84	82.55	85.78
Precision (%)	80.73	90.80	80.99
Recall (%)	63.64	72.30	95.17
F1-score (%)	71.17	80.50	87.51

2010], which is based on skin color analysis, and the other was a tool named LED, which aims to find already known digital evidence. NuDetective achieved an accuracy of 57.43%, while LED achieved a better accuracy of 76.47%, with lower precision and higher recall rates. The results are presented in Table 5.4, where we added the results of our proposed method for comparison purposes.

Table 5.3: CSAM Detection (Seminude, Nude and Sex).

Metric	Child Seminude+	Child Detection	Seminude+
Accuracy (%)	81.15	82.55	94.11
Precision (%)	83.90	90.80	94.45
Recall (%)	67.01	72.30	96.73
F1-score (%)	74.50	80.50	95.57

The experiments were performed on an Intel Xeon E5-2630@2.6Hz with an NVIDIA GTX 1080 TI graphics card. The proposed method takes an average time of 0.38 s per image: 0.05 on the pornography classification task, 0.32 s on the face detection and alignment tasks, and 0.01 s on the age estimation task.

Table 5.4: Evaluation of Forensic Tools.

Metric	NuDetective [Polastro and Eleuterio, 2010]	LED	Proposed Approach
Accuracy (%)	57.44	76.47	**79.84**
Precision (%)	**47.46**	75.34	**80.73**
Recall (%)	**82.54**	59.21	63.64
F1-score (%)	60.26	66.31	**71.17**

5.3 Discussion

Experimental evaluation with the proposed method on the RCPD dataset yielded an accuracy of 79.84%. It demonstrates the suitability of the proposed face-based child detection combined with a pornography detector. At the same time, it shows that there is room for improvement in this challenging application.

Although the method succeeded in achieving better results than the analyzed forensic tools, it requires more resources to process the images, which we do not consider a limitation factor, and it has a weakness when it comes to CSA images without child's faces.

In respect to the forensic tools, one of the main drawbacks of pornography or CSAM detection methods based on skin color analysis is the high rate of false positives, and this behavior was observed in the results of NuDetective: the tool selected 1456 images, while there are just 836 files related to child pornography. It classified as positive 155 (almost 31%) files that did not contain any person, and had a relatively low recall rate, which is not desirable for this kind of application.

On the other hand, the analysis of LED results indicates that the tool had a good performance on the dataset, achieving an accuracy of 76.47% with relatively good precision and a recall rate better than the one achieved by NuDetective, while presented just a small number of false positives. These results demonstrate the power of this category of tools when the search is conducted against already known files, as it happens with RCPD dataset, that had its files gathered from real cases. In these situations, hash-based tools can achieve good results with the additional advantage of being faster than any other approach, which makes it especially suited for use in the field.

Despite the good performance of LED, it is important to emphasize that hash-based approaches usually fail when there are few known files. In these cases, a method like the proposed in this work usually has better results. In the specific case of the RCPD, the proposed method achieved a better accuracy, with higher recall rate, even considering a dataset that had its images selected from real cases.

5.4 Video Analysis

The method has been extended for video analysis, using a frame sampling strategy that selects a number of frames according to the length of the video. For videos lasting

Figure 5.4: Result of the classification of a video file. The eight most relevant frames are reported.

less than 60 seconds, two frames per second are selected. For videos up to 10 minutes, one frame per second is selected. For larger videos, one frame is selected every two seconds. Half of the frames is extracted sequentially from the video and the other half is extracted randomly.

The selected frames are classified using the procedure described in section 4.1. Then, a ranking of the selected frames is carried out according to the level of relevance to the CSAM classification task. Frames containing pornography and containing at least one child face are placed in the first group of the ranking, sorted in descending order by the probability of pornography of the frame. If the video contains at least two frames classified as child pornographic, the video is classified as related to child pornography.

The frames containing pornography that do not contain a child's face are placed in the second group of the ranking, sorted in descending order by the probability of pornography of the frame. The remaining frames are placed in the third group of the ranking and are also sorted in descending order by the probability of pornography of the frame. The three groups are concatenated, giving an ordered representation of the video frames with respect to child pornography and pornography.

This approach was tested experimentally in a forensic scenario, where the eight most relevant frames of the video were reported, presenting good results in the classification of videos. An example of the output of this approach to a video file without pornography is shown in Figure 5.4. Each frame if tagged with the pornography classification probability and the faces are detected and classified as child or adult.

Due to the lack of a CSA video dataset, the method was not formally tested. The

method was only evaluated in forensic cases.

Chapter 6

Conclusions and Future Work

In this work, we proposed a combined method to detect CSAM using a child face detection module and a pornography detection method. The proposed method classifies images as related to child pornography only if the images contain pornography and if a child's face is detected. The child face detection module used in the proposed method was built through the integration of a face detector and an age estimation method, which was specifically developed for this work.

To evaluate the proposed method, we used the child pornography region-based annotated dataset (RCPD) [Macedo et al., 2018], which belongs to the Brazilian Federal Police and was set up during the development of this work. The dataset contains CSA images gathered and labeled internally the Brazilian Federal Police. These images are mixed up with groups of images not related to CSAM, including images with no person, images with adults, images with children and pornographic images with adults, creating a diversity of situations that can be evaluated by the dataset.

Besides age, gender and nudity exposure labels of the individuals that appear in the images, the RCPD dataset also contain region-based annotations of the face and relevant parts of the individuals in the images. These data can be used to evaluate CSAM classification methods, as well as detectors of human body parts, such as a detector of woman's breast.

The evaluation of the proposed CSAM classification method against the RCPD dataset achieved an accuracy of 79.84%, which was better than the results achieved by two forensic tools that we evaluated in the same dataset: LED and NuDetective. The experiments demonstrated that the proposed method achieved a reasonable accuracy and proved the usability of the described dataset to compare CSAM detection methods. The proposed method does not exploit the full potential of the dataset, which has region-based annotations of parts of the individuals in the image and can be used to evaluate detection methods. However, the simple classification analysis already showed that the existence of a benchmark dataset in this area can form a basis for further research, comparison and optimization of methods.

These results also showed the viability of the strategy of combining different classifiers and detectors to perform the CSAM detection task, but there is still much room for

improvement, either in the age estimation task as in the pornography classification task. The age estimation task can be further improved with a fine-tuning step in the IMDB-WIKI dataset, as it was done with significant performance improvements in [Zhang et al., 2017; Hou et al., 2017; Rothe et al., 2018].

As a future work, we plan to evaluate other CSAM classification methods in the RCPD dataset and we plan to extend the dataset to include video files.

A work to be explored is the design of a single network to simultaneously learn the correlated face detection, alignment and age estimation tasks. This strategy could bring better execution time and performance gains that could result from the joint learning of correlated tasks.

One promising area to explore is the development of a method to detect private parts of children in images using or improving existing methods [Redmon et al., 2016; Liu et al., 2016]. This approach could compensate the weakness of our proposed method, which can not detect child sexual abuse in images that do not have a child's face.

Finally, another research area to be explored is the usage of image captioning techniques [You et al., 2016] for child sexual abuse problems. The objective is to provide textual description of the images, which may include the presence of adults, the presence of objects, like a bed or a pillow and scenary details, including hints of the place where the image was taken.

Bibliography

A. M. Albert, K. Ricanek, and E. Patterson. A review of the literature on the aging adult skull and face: implications for forensic science research and applications. *Forensic science international*, 172:1–9, October 2007. ISSN 1872-6283. doi: 10.1016/j.forsciint.2007.03.015.

E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2010.

R. Ap-Apid. An algorithm for nudity detection. In *5th Philippine Computing Science Congress*, pages 201–205, 2005.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2009.

T. Brox, A. Bruhn, N. Papenberg J., and Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

Peter Bruce and Andrew Bruce. *Practical Statistics for Data Scientists: 50 Essential Concepts*. " O'Reilly Media, Inc.", 2017.

C. Caetano, S. Avila, S. Guimarães, and A. d. A. Araújo. Pornography detection using bossanova video descriptor. In *Proc. 22nd European Signal Processing Conf. (EUSIPCO)*, pages 1681–1685, September 2014.

R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

R. Caruana. A dozen tricks with multitask learning. In *Neural Networks: Tricks of the Trade*, pages 163–189. Springer, 2012.

S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao. Using ranking-CNN for age estimation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 742–751, July 2017. doi: 10.1109/CVPR.2017.86.

M. Chopra, M. V. Martin, L. Rueda, and P. C. k. Hung. Toward new paradigms to combating internet child pornography. In *Proc. Canadian Conf. Electrical and Computer Engineering*, pages 1012–1015, May 2006. doi: 10.1109/CCECE.2006.277790.

T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. doi: 10.1109/CVPR.2005.177.

T. Deselaers, L. Pimenidis, and H. Ney. Bag-of-visual-words models for adult image classification and filtering. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

Y. Dong, Y. Liu, and S. Lian. Automatic age estimation based on deep learning algorithm. *Neurocomputing*, 187:4–10, 2016.

M. Duan, K. Li, C. Yang, and K. Li. A hybrid deep learning cnn–elm for age and gender classification. *Neurocomputing*, 275:448–461, 2018.

R. J. Edelmann. Exposure to child abuse images as part of one's work: Possible psychological implications. *The Journal of Forensic Psychiatry & Psychology*, 21(4):481–489, 2010.

E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, December 2014. ISSN 1556-6013. doi: 10.1109/TIFS.2014.2359646.

FGnet. Fg-net aging database, 2002. URL `http://www-prima.inrialpes.fr/FGnet/`.

Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, November 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.36.

K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

D. Ganguly, M. Mofrad, and A. Kovashka. Detecting sexually provocative images. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 660–668. IEEE, 2017.

X. Geng, Z.-H. Zhou, Y. Zhang, G. Li, and H. Dai. Learning from facial aging patterns for automatic age estimation, 2006.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

S. Greijer and J. Doek. *Terminology guidelines for the protection of children from sexual exploitation and sexual abuse*. ECPAT International, 2016. URL `http://luxembourgguidelines.org/english-version/`.

G. Guo and G. Mu. A framework for joint estimation of age, gender and ethnicity on a large database. *Image and Vision Computing*, 32(10):761–770, 2014.

G. Guo, Guowang Mu, Y. Fu, and T. S. Huang. Human age estimation using bio-inspired features. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 112–119, June 2009. doi: 10.1109/CVPR.2009.5206681.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

L. Hou, D. Samaras, T. Kurc, Y. Gao, and J. Saltz. Convnets with smooth adaptive activation functions for regression. In *Artificial Intelligence and Statistics*, pages 430–439, 2017.

D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen. Local binary patterns and its application to facial image analysis: A survey. *Part C (Applications and Reviews) IEEE Transactions on Systems, Man, and Cybernetics*, 41(6):765–781, November 2011. ISSN 1094-6977. doi: 10.1109/TSMCC.2011.2118750.

I. Huerta, C. Fernández, C. Segura, J. Hernando, and A. Prati. A deep analysis on age estimation. *Pattern Recognition Letters*, 68:239–249, 2015.

Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

M. Krause. Identifying and managing stress in child pornography and child exploitation investigators. *Journal of Police and Criminal Psychology*, 24(1):22–29, 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

Y.H. Kwon and N.V. Lobo. Locating facial features for age classification. In *Proceedings of SPIE-the International Society for Optical Engineering Conference*, pages 62–72, 1993.

Y.H. Kwon and N.V. Lobo. Age classification from facial images. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition 1994*, pages 762–767, June 1994. doi: 10.1109/CVPR.1994.323894.

A. Lanitis, C. Draganova, and C. Christodoulou. Comparing different classifiers for automatic age estimation. *Part B (Cybernetics) IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):621–628, February 2004. ISSN 1083-4419. doi: 10.1109/TSMCB.2003.817091.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

Y. A. LeCun, L. Bottou, G. B. Orr, and K.-B. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

G. Levi and T. Hassncer. Age and gender classification using convolutional neural networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 34–42, June 2015. doi: 10.1109/CVPRW.2015.7301352.

Y. Lin, H. Tseng, and C. Fuh. Pornography detection using support vector machine. In *16th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP 2003)*, volume 19, pages 123–130, 2003.

H. Liu, J. Lu, J. Feng, and J. Zhou. Label-sensitive deep metric learning for facial age estimation. *IEEE Transactions on Information Forensics and Security*, 13(2):292–305, 2018.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

J. Macedo, F. Costa, and J. Santos. Proceedings of the 30th sibgrapi conference on graphics, patterns and images (sibgrapi). In *SIBGRAPI*, Foz do Iguaçu, PR, Brazil, October 2018. IEEE.

J. Mahadeokar and G. Pesavento. Open sourcing a deep learning solution for detecting nsfw images, 2016. URL `https://yahooeng.tumblr.com/post/151148689421/open-sourcing-a-deep-learning-solution-for`.

T. M. Mitchell. *Machine learning*. McGraw-Hill, 1997.

K. Murphy. *Machine learning: a probabilistic approach*. 2012.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

F. Nian, T. Li, Y. Wang, M. Xu, and J. Wu. Pornographic image detection utilizing deep convolutional neural networks. *Neurocomputing*, 120:283—-293, 2016.

NIST. Forensic database tech digital evidence table, 2017. URL `https://www.nist.gov/forensics/forensic-database-tech-digital-evidence-table`.

Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Ordinal regression with multiple output CNN for age estimation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 4920–4928, June 2016. doi: 10.1109/CVPR.2016.532.

T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha. Video pornography detection through deep learning techniques and motion information. *Neurocomputing*, 230:279–293, 2017.

P. J. Phillips, Hyeonjoon Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, October 2000. ISSN 0162-8828. doi: 10.1109/34.879790.

P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, volume 1, pages 947–954. IEEE, 2005.

M. C. Polastro and P. M. S. Eleuterio. Nudetective: A forensic tool to help combat child pornography through automatic nudity detection. In *Database and Expert Systems Applications (DEXA), 2010 Workshop on*, pages 349–353. IEEE, 2010.

R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1, 2017a. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2781233.

R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *Proc. 12th IEEE Int. Conf. Automatic Face Gesture Recognition (FG 2017)*, pages 17–24, May 2017b. doi: 10.1109/FG.2017.137.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

K. Ricanek and T. Tesafaye. Morph: a longitudinal image database of normal adult age-progression. In *Proc. 7th Int. Conf. Automatic Face and Gesture Recognition (FGR06)*, pages 341–345, April 2006. doi: 10.1109/FGR.2006.78.

I. E. Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex, 1999.

R. Rothe, R. Timofte, and L. Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018.

S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

N. Sae-Bae, X. Sun, H. T. Sencar, and N. D. Memon. Towards automatic detection of child pornography. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5332–5336. IEEE, 2014.

S. J. Sathish and S. H. Sengamedu. Texture-based pornography detection, July 3 2008. US Patent App. 11/715,051.

N. Sebe, I. Cohen, A. Garg, and T. S. Huang. *Machine learning in computer vision*, volume 29. Springer Science & Business Media, 2005.

T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, 29(3):411–426, March 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.56.

M. C. Seto, C. Buckman, R. G. Dwyer, and E. Quayle. Production and active trading of child sexual exploitation images depicting identified victims: Ncmec/thorn research report. *Alexandria, VA: National Center for Missing and Exploited Children*, 2018.

A. Shupo, M. V. Martin, L. Rueda, A. Bulkan, Y. Chen, and P. C. Hung. Toward efficient detection of child pornography in the network infrastructure. *IADIS International Journal on Computer Science and Information Systems*, 1(2):15–31, 2006.

T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, December 2003. ISSN 0162-8828. doi: 10.1109/TPAMI.2003.1251154.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

M. Thakor. How to look: Apprehension, forensic craft, and the classification of child exploitation images. *IEEE Annals of the History of Computing*, 39(2):6–8, 2017. ISSN 1058-6180. doi: 10.1109/MAHC.2017.25.

A. Ulges and A. Stahl. Automatic detection of child pornography using color visual words. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

P. A. Vitorino, S. Avila, M. Perez, and A. Rocha. Leveraging deep neural networks to fight child pornography in the age of social media. *Journal of Visual Communication and Image Representation*, 50:303–313, 2018.

Y. Wang, Q. Huang, and W. Gao. Pornographic image detection based on multilevel representation. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(08):1633–1655, 2009.

J. Wehrmann, G. Simões, R. C. Barros, and V. F. Cavalcante. Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing*, 272:432–438, 2018.

Z. Yang and H. Ai. Demographic classification with local binary patterns. *Advances in Biometrics*, pages 464–473, 2007.

E. Yiallourou, R. Demetriou, and A. Lanitis. On the detection of images containing child-pornographic material. In *Proc. 24th Int. Conf. Telecommunications (ICT)*, pages 1–5, May 2017. doi: 10.1109/ICT.2017.7998260.

Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

K. Zhang, C. Gao, L. Guo, M. Sun, X. Yuan, T. X. Han, Z. Zhao, and B. Li. Age group and gender estimation in the wild with deep ror architecture. *IEEE Access*, 5: 22492–22503, 2017. doi: 10.1109/ACCESS.2017.2761849.

Ke Zhang, Miao Sun, Tony X Han, Xingfang Yuan, Liru Guo, and Tao Liu. Residual networks of residual networks: Multilevel residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(6):1303–1314, 2018.