Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Vinicius Almeida de Freitas

Sensitive Content Detection in Video with Deep Learning

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Sérgio Colcher

Rio de Janeiro
April 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Vinicius Almeida de Freitas

Sensitive Content Detection in Video with Deep Learning

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee.

Prof. Sérgio Colcher
Advisor
Departamento de Informática – PUC-Rio

Prof. Alberto Barbosa Raposo
Departamento de Informática – PUC-Rio

Prof. Sandra Eliza Fontes de Avila
UNICAMP

Rio de Janeiro, April 1st, 2022

Pedro Vinicius Almeida de Freitas

Bachelor's degree in Computer Science at Federal University of Maranhão (UFMA) in 2019.

Bibliographic data

Freitas, Pedro Vinicius Almeida de

Sensitive Content Detection in Video with Deep Learning / Pedro Vinicius Almeida de Freitas; advisor: Sérgio Colcher. – Rio de janeiro: PUC-Rio, Departamento de Informática, 2022.

v., 68 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia

1. Conteúdo Sensível;. 2. Detecção de Conteúdo Sensível;. 3. Classificação Multimodal de Videos;. 4. Deep Learning.. I. Colcher, Sérgio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgments

Thanks to my advisor, Sérgio Colcher, for all his lessons and guidance during my masters degree.

I am also thankful for having a supportive family, who encouraged me to go pursue my academic career even if that would set us apart.

Thanks for all my friends, old and new, whom helped me get through the loneliest days.

Thanks to all my friends and colleagues at Telemidia Lab for all their support, companionship, and good times.

To all colleagues, faculty and staff of the PUC Rio Department of Informatics for the fellowship, learning and support.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

This study was financed in part by the Rede Nacional de Pesquisa (RNP) and Microsoft, through the AI Challenge of 2019.

Abstract

Freitas, Pedro Vinicius Almeida de; Colcher, Sérgio (Advisor). **Sensitive Content Detection in Video with Deep Learning**. Rio de Janeiro, 2022. 68p. Dissertação de mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Massive amounts of video are uploaded on video-hosting platforms every minute. This volume of data presents a challenge in controlling the type of content uploaded to these video hosting services, for those platforms are responsible for any sensitive media uploaded by their users. There has been an abundance of research on methods for developing automatic detection of sensitive content. In this dissertation, we define sensitive content as sex, extreme physical violence, gore, or any scenes potentially disturbing to the viewer. We present a sensitive video dataset for binary video classification (whether there is sensitive content in the video or not), containing 127 thousand tagged videos, Each with their extracted audio and visual embeddings. We also trained and evaluated four baseline models for the sensitive content detection in video task. The best performing model achieved 99% weighed F2-Score on our test subset and 88.83% on the Pornography-2k dataset.

Keywords

Sensitive Content; Sensitive Video Dataset; Multimodal Video Classification; Deep Learning.

Resumo

Freitas, Pedro Vinicius Almeida de; Colcher, Sérgio. **Detecção de Conteúdo Sensível em Video com Aprendizado Profundo**. Rio de Janeiro, 2022. 68p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Grandes quantidades de vídeo são carregadas em plataformas de hospedagem de vídeo a cada minuto. Esse volume de dados apresenta um desafio no controle do tipo de conteúdo enviado para esses serviços de hospedagem de vídeo, pois essas plataformas são responsáveis por qualquer mídia sensível enviada por seus usuários. Nesta dissertação, definimos conteúdo sensível como sexo, violencia fisica extrema, gore ou cenas potencialmente pertubadoras ao espectador. Apresentamos um conjunto de dados de vídeo sensível para classificação binária de vídeo (se há conteúdo sensível no vídeo ou não), contendo 127 mil vídeos anotados, cada um com seus embeddings visuais e de áudio extraídos. Também treinamos e avaliamos quatro modelos baseline para a tarefa de detecção de conteúdo sensível em vídeo. O modelo com melhor desempenho obteve 99% de F2-Score ponderado no nosso subconjunto de testes e 88,83% no conjunto de dados Pornography-2k.

Palavras-chave

Conteúdo Sensível; Detecção de Conteúdo Sensível; Classificação Multimodal de Videos; Deep Learning.

Table of contents

List of figures

List of tables

List of abreviations

DL – Deep Learning

CNN – Convolutional Neural Network

RNN – Recurrent Neural Network

LSTM – Long Short Term Memory

AUC – Area Under The Curve

ROC – Receiver Operating Characteristics

NPDI – Núcleo de Processamento Digital de Imagens

MAP – Mean Average Precision

GRU – Gated Recurrent Units

SVM – Support Vector Machines

ANN – Artificial Neural Network

MLP – Multi Layer Perceptron

ILSVRC – ImageNet Large Scale Visual Recognition Challenge

ReLU – Rectified Linear Uni

RGB – Red, Green and Blue

FC – Fully Connected neural network

MMA – Mixed Martial Arts

STD – Standard Deviation

NSFW – Not Safe For Work

KNN – K-Nearest Neightbors

ANOVA – Variance Analisys

RNP – Brazil's National Research Net

1
Introduction

The amount of multimedia content on the internet is increasing every year. More than 300 hours of video are uploaded to YouTube every minute.[1] In this context, studies have shown that about 56% of children between 10 and 13 years old have a smartphone (7, 8), and 8 out of 10 teenagers have had a friend who shared some sensitive media through social networks such as Facebook, Twitter, and Whatsapp.[2]

This huge amount of data sharing pattern presents a challenge to the control of the type of content that is loaded to these video repositories. By allowing the upload of sensitive content from malicious users, content providers become exposed to legal issues. This is also a problem for users in those platforms, as they might get exposed to this content without a warning.

In Brazil, the "Cicarely case" was an example resulted in the nation-wide blocking of YouTube.[3] In our research, we are interested in helping to avoid scenarios where sensitive content can be uploaded to education and unsuitable channels, which might expose students, sometimes underage, to this content.[4]. This scenery presents challenges on controlling which type of contents are uploaded to these storage and distribution services, while dealing with great amounts of videos. Our approach is set to be hosted and executed on the platform itself, so that the platform itself can regulate, through retraining, what type of content is allowed.

Methods based on *Deep Learning* (DL) became *state-of-the-art* in various segments related to automatic video analysis. More specifically, Convolutional Neural Networks (CNN) architectures, or ConvNets, have become the primary method used for audio-visual pattern recognition (9, 10, 11).

The term *Sensitive content* is often used as a reference to any media that contains content such as nudity, intercourse, extreme physical violence, gore,

[1] https://biographon.com/youtube-stats

[2] https://www.netnanny.com/the-importance-of-parental-control/

[3] http://g1.globo.com/Noticias/Tecnologia/0,,AA1412609-6174-363,00.html

[4] https://g1.globo.com/sp/sao-paulo/noticia/2020/06/19/professor-de-etec-na-zona-norte-de-sp-e-afastado-apos-se-masturbar-durante-aula-virt ghtml

or any scenes potentially disturbing to the viewer. On the other hand, content is labeled as *Safe* when this content is suitable for the general public.

Figure 1.1: Examples of safe (top row) and sensitive videos (bottom row).

Figure 1.1 illustrates these two categories. There are four scenes with safe content on the top row, and four scenes with sensitive content on the bottom row.

Other works, such as (12), share our motivations and objectives, as described in Section 2. However, most of them do not use both audio and image for classification. We use recent CNNs that have been showing great potential in video recognition and classification. Furthermore, none of them have the same definition of sensitive content as ours. For instance, the violence aspect of sensitive content citemoreira2019multimodal comprises any kind of physical violence, such as fights. In our definition, the violence aspect is defined by only potentially disturbing scenes and extremely violent acts, such as torture, death, suicide, etc.

Our work uses two CNNs: one to extract image sequence features and the other to extract audio features. As we get one feature vector for each second of the video, we can approach the feature classification task as a time series classification, using a Recurrent Neural Network (RNN) as a baseline. We also can combine those features to create a single feature vector for the entire video, which is then used as the input for other baseline classifiers.

Although sometimes we may refer to the task we are addressing as sensitive content detection in video, our task is, specifically, binary classification of video: Finding out whether sensitive content is or is not present in the video. Furthermore, since there is no frame-by-frame annotation, this dataset does not directly support the task of finding (either time-wise or space-wise) sensitive content in the video.

In this work, the main research question is if a generic feature extraction, based on transfer learning, can achieve results that approach fine tuned and hand crafted approaches. The research questions we aim to answer with this work are:

1. Can this transfer learning-based, multimodal approach archieve results within 10% of the results from related work?

2. What is the impact of also using audio in the model's performance?

The main contributions of this work are:

1. To our knowledge, the largest sensitive content detection dataset, when balanced, it has approximately 110.000 videos, it its composed by 67.424 sensitive videos and 59651 safe videos.

2. We trained and tested baseline classifiers (KNN,SVM,MLP and LSTM) on the features extracted from our dataset in order to validate both the dataset and the feature extraction networks.

3. We tested sequential (LSTM) and non sequential (KNN, SVM and MLP) classifiers in this task.

4. We tested the importance of image and audio features in our approach by comparing the results of our approach when input with only one of each type of features.

5. We also validate our approach by testing our best baseline (MLP) in a well known pornography detection dataset, the Pornography-2k (13) dataset.

Our approach yielded an F2-Score of 88.83%, compared to our related works, Moreira et. al. with 93.53% (12) and who also aim at pornography and violence detection, and Wehrmann et. al. with 95.20% (14), aiming at only pornography.

To perform the sensitive detection task, we created a large scale dataset, extracted features from this dataset using a generalist and well known feature extraction for video classification method, and performed experiments such as compare baseline classification models, compare which type of classification model (sequential or not) performs best, and compared the importance of audio and image features, further detailed in Chapter 5.

Although the largest dataset for this task by our knowledge, our dataset is not manually labeled, which begs the question if it is noise-less enough for any training and evaluation in this task. Our intent is not to replace the Pornography-2k dataset, but to be a complement to it, it still is the gold standard for pornography detection, in our dataset the videos were not manually labeled by a human, so we need to validate this dataset. Through this dissertation, we aim to validate our dataset by assembling a baseline approach

to tackle this task and then evaluating our baseline approach on a manually labeled dataset, the Pornography-2k dataset.

This dissertation is organized as follows: In Chapter 2 we discuss some of the related work. In Chapter 3 we present the theory behind some of the techniques used in the feature extraction method we adopted. We present our dataset and metrics in Chapter 4. Then, in Chapter 5, we present baseline models to detect sensitive content in videos.

Then, we evaluate and analyse our baseline models in Chapter 6. Finally, in Chapter 7 we present, our conclusions, currently published papers, and future work. Additionaly, in Appendix A, we show complementary data, such as tables, distributions and a dataset datasheet.

2
Related Work

In this Chapter, we present the most related works to the sensitive content detection in video. As there are few related works using deep learning to tackle this task and as it is composed of two other tasks, violence and pornography, we chose to also include works specific to each sub-task.

Castro (15) shows an implementation of a pornography video classifier using a convolutional neural network from Open pornography (16) and the dataset from Nude Detection in Video using Bag-of-Visual-Features (17) dataset. The CNN does a logistic regression on each frame, resulting in a value from 0 to 1 at each frame. The higher the value is, the higher the likelihood of the frame being pornography. The dataset used contained 90 non-pornography video segments and 89 pornography video segments extracted from 11 movies. The final score for the video is the max value from all frames of the video. The experiment showed an accuracy of 81%, an F1 score, and Matthew's correlation coefficient(MCC) for the pornography class of 0.8047 and 0.6343, respectively. Although the work also approaches pornography content detection in videos problem with CNN like ours, it does not make use of audio features. The method is also different, it performs the regression first, then it takes the max value from all frames of the video, while ours, in the non-sequential approach, combines features from all frames of the video into a single vector of features (mainly by averaging) and then performs classification on the resulting features.

Wehrmann *et al.* (14) classify adult content trained on the NPDI pornography video dataset (18), which consists of 802 videos, totaling 80 hours of videos, half of them with adult content. Those videos were processed by keyframes, varying between 1 and 320 frames per video. The selected keyframes of each video were chosen by a scene segmentation algorithm, resulting in 16727 images. Their architecture consists of a Convolutional Network and a Long-Short Term Memory Network (LSTM) (19). Those models were chosen for feature extraction with CNN and sequence learning with LSTM, taking into consideration modifications on the images such as scaling and distorting. Using this approach the authors achieved a score of 95.6% $\pm$ 1 accuracy and 0.990 AUC(ROC). In our model, we also approached the video analysis using

frame by frame processing, but we also processed the extracted sound from each frame.

Sing *et. al.*(20) proposes a fine-grained approach for child unsafe video representation and detection. One of its main objectives is to optimize the detection of sparsely present child unsafe content and it does so by using a VGG16(21) Convolutional Neural Network (CNN) to encode each frame, at 1-second granularity, in 512 real values. Then an LSTM autoencoder is trained to output the sequence backward on those encoded frames. Once the LSTM autoencoder is trained, then a fully connected layer of neurons is used to fine-tune and classify each frame. The dataset used comprises 109,835 short-duration video clips extracted from four animes. The results for binary classification using safe and unsafe classes were 81% recall for unsafe and 0.88 AUC(ROC) for unsafe class. Although this work also has similar objectives as ours and also uses a CNN-based encoding method, ours uses both visual and audio features to encode a video. The main difference between both works is in the dataset: Theirs consists of small clips of only anime videos. Ours also uses other types of videos such as live-action and other animations.

Song *et. al.* (22) proposed a multimodal stacking scheme for fast and accurate online detection of pornographic content. Their work uses both visual and auditory features as input for their detection method. They use a VGG16 model and a bi-directional LSTM to extract visual features and a combination of a Mel-scaled spectrogram followed by multilayered dilated convolutions to extract audio features. Using only the visual and auditory features, a video classifier and an audio classifier are trained, respectively. By using both features together, one fusion classifier is also trained. Then, these three component classifiers are combined in an ensemble scheme to reduce the false-negative errors and for faster detection. The proposed detection method yields a true positive rate of 95.40% and a false negative rate of 4.60% on the pornography class, totaling a recall for the pornography class of 95.40% and an accuracy of 92.33%. The dataset used was the Pornography-2k(13) dataset plus examples of videos with only pornographic or non-pornographic audio collected by the authors. This work is similar to ours because it also uses a multimodal approach to detection, albeit ours is not for pornography detection only. It also uses the same sampling rate of a frame for each second and uses a deep learning method for extracting high-level features, which are then classified by one or more machine learning models. We also use different feature extraction methods for image and audio features. Finally, in contrast with their ensemble approach, we use a single model to classify the extracted features from our dataset.

Moreira *et.al.* (12) have similar detection focuses as ours: Pornography

and Violence. Their method uses four multi-modal classifiers, two for audio and two for image, those classifiers were fed features from multiple handcrafted feature extraction methods. Their work is geared towards mobile device applications and also allows for sensitive scene localization. The authors propose a method for sensitive scene localization which uses the output of four multi-modal classifiers on snippets of the video, then creates a fusion vector at each second of the video. Finally, they test different classifiers on the fusion vector for each task: detecting pornography and detecting violence. Their best result on the pornography task was 90.75% accuracy and 93.53% on the F2 metric. For the violent videos, they achieved 0.502 on the MAP2014 evaluation metric. Some differences between this work and ours are mainly its objectives: To detect if and at what time the sensitive video occurs. While our only objective is to detect if there is or is not sensitive content in a video. Their method is geared towards mobile devices, while ours is geared towards video hosting platforms. Other differences stand out in the dataset and the methods used for feature extraction and classification. The Violent Scenes Dataset (23) is comprised of violent scenes from movies, while ours contains real violent scenes. We use an authorial dataset and investigate what results a deep learning-based approach to this problem can yield.

Wang *et. al.* (24) propose a pornography method for use in live streams, focusing on real-time processing, their work uses multimodal features, namely, image, audio, and optical flow (25). An Xception (8) model is used to extract spatial features from keyframes. To get the optical flow frames, they also use a CNN to extract the optical flow from the video, then, use another Xception model to extract the high-level optical flow features. Finally, they use a short-time Fourier transformation to create spectrograms and feed those spectrograms to a third Xception model and thus acquiring the extracted audio features. Each of the multimodal features extracted then is passed onto bidirectional GRUs(26), to obtain temporal context, then, to create a better-unified representation, all the features go through three interconnected Attention-gated layers, each with three Attention-gated units proposed in the paper. After obtaining the dense representation of the input types, it is applied a fully connected layer of neurons with a *softmax* function. Their work archives 76.33% accuracy and runs at 66.1 fps. In our work, we strive for detecting both violence and pornography, we use only two types of input data, image, and audio, and we use a specific CNN for each type of data, while their work focused only on detecting pornography and used the same CNN model for all three types of input.

Liu *et. al.* (27) propose a multi-modal approach to pornography detec-

tion, it uses audio frames and visual frames to create handcrafted low-level features based on, respectively, periodic patterns and salient regions. Once those features are extracted, they use k-means clustering to create audio and visual codebooks. Then, low-level audio and visual features of test videos are converted into mid-level semantic histograms via de audio or visual codebook. Finally, the histograms are concatenated to represent the video and a periodicity-based video decision algorithm is used to fuse the classification results of multi-modal codebooks and the results of an SVM trained on the concatenated mid-level semantic features train set. The true positive rate of their approach achieves 96.7% while the false positive rate is about 10%. There are three papers about our work have already been published:

– Freitas et al.(28), which describes our approach and model on a early version of this dataset, containing about 60.000 sensitive and safe videos.
– Freitas et, al.(29), which describes how our approach fairs on pornography detection in educational video-hosting platforms.
– Serra et. al.(30), which describes an method, based on our model for sensitive content detection, for self-monitoring and parental control on mobile phones.

Most related works focus on pornography detection alone, while ours aims at detecting either pornographic or violent content. Moreover, some of them only use image-frame features, whereas we use both audio and image-frame features. We also use deep learning feature extraction methods instead of handcrafted ones. Feature extraction method, classification method, and dataset of each related work are available in Table 2.1. Finally, a central difference is our dataset: Ours contains violent scenes and is significantly larger than most datasets used on other related works.

Table 2.1: Related work comparative table.

Paper	Task	Feature extraction method	Classifier	F2-Score	Recall	Accuracy	F1-Score	AUC (ROC)	Dataset
Castro (15)	Pornography detection	Resnet 50 for image.	Resnet-50	0,7798	0,7640	0,8160	0,8047	NA	Open pornography + Nude Detection in Video using Bag-of-Visual-Features dataset
Wehrmann et al. (14)	Pornography detection	ResNet-101 for image.	LSTM	0,9520	0,9501	0,9560	0,9548	0,9900	NPDI pornography video dataset (800 videos)
Sing et. al. (20)	Sensitive content detection	VGG16 for image.	LSTM + FC	NA	0,8100	NA	NA	0,8800	Author (Animes)
Song et. al. (22)	Pornography detection	VGG16 + BiLSTM for image; Mel-scaled spectrogram + Multilayered dilated convolutions for audio.	Early + Late fusion FC voting	NA	95.40%	0,9233	NA	NA	Pornography-2k
Moreira et.al. (12)	Sensitive content detection	HOG for image; TRoF for space-temporal description; MFCC and prosodic features for audio.	Thresholding (Pornography); SVM (Violence).	93,53% (Pornography)	NA	90,75% (Pornography); 0.502 MAP2014 (Violence).	NA	NA	Pornography-2k + Violent Scenes Dataset (MediaEval 2014)
Wang et al. (24)	Pornography detection	Xception for image; Optical flow + Xception for motion; Short-time Fourier transformation + Xception for audio.	bidirectional GRUs + Attention + FC	NA	NA	76,33%	NA	NA	BJUT streamer dataset (Author)
Liu et. al. (27)	Pornography detection	Skin color detection + Face detection + Salient regions detection + SURF for image.	SVM	NA	NA	96,70%	NA	NA	Author

3
Theory and technical background

This Chapter aims to set a basic understanding of the concepts underlying the techniques used in this Dissertation.

3.1
Artificial Neural Networks

Artificial Neural Networks (ANNs) are machine learning models inspired by biological neurons. The Perceptron (31) is one of the main precursors of modern ANNs. The Perceptron is a mathematical model of the Neuron, it is capable of binary classification. It draws its differentiation power from adjusting a linear function with its weights. It can differentiate any linearly separable problem, that is, any problem in which a hyperplane (a plane in multiple dimensions) can separate the two classes of the data.

A Perceptron receives m inputs, denoted X, it holds m weights W, one weight for each input, and a *bias*. During training, the weights W and the *bias* are adjusted in order to optimize hyperplane separation.

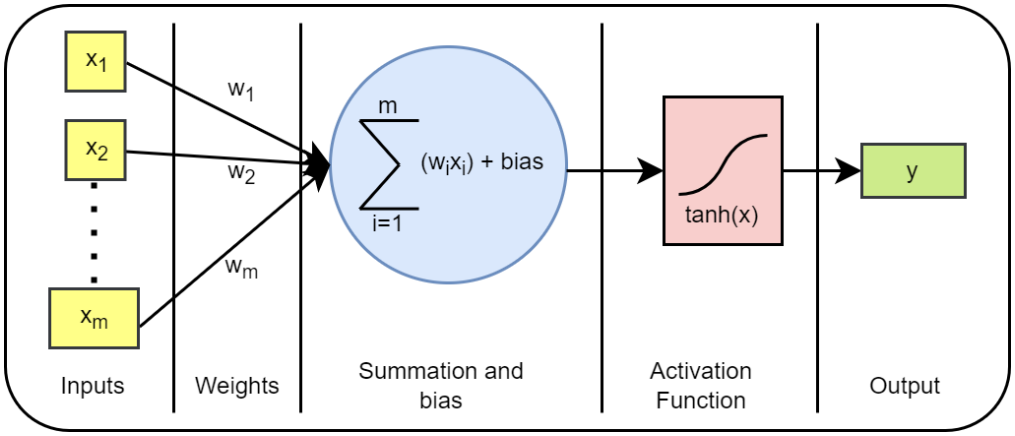


Figure 3.1: The Perceptron and its components, the input layer, the weights, the weighted sum and bias, the activation function, and the output layer.

To obtain the output of a Perceptron (a prediction), its weights are multiplied by the inputs, then the sum of these multiplications is summed and then a bias is added to this result. This weighted sum of the input features can be calculated through Equation 3-1.

$$\sum_{i=1}^{m}(x_i w_i) + bias \tag{3-1}$$

Where m is the number of inputs, x_i and w_i are the inputs and weights and i is the number of the input. Finally, the weighted sum of the input features (and bias) are input through an activation function, which in the original perceptron is a *Step function*, but in Figure 3.1, which shows a diagram of the Perceptron structure, is a *Hyperbolic tangent* function. The result of the activation function is the output of the Perceptron.

The learning process of the Perceptron is adjusting the weights and the bias so that the hyperplane can separate the training data up to a set metric.

By stacking layers of multiple Perceptrons, one can approximate any continuous function, rather than only linear functions, thus being able to solve both linear and non-linear problems. MLPs are also known as Fully Connected (FC) neural networks when combined with other modern neural networks. The general structure of a Multi-Layer Perceptron (MLP), as shown in Figure 3.2 consists of an input layer, one or more hidden layers, and an output layer.

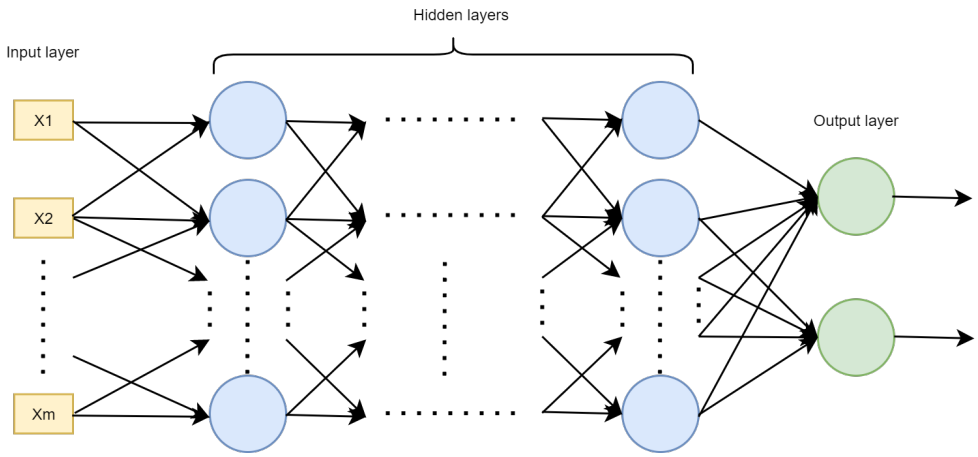


Figure 3.2: The Multi-Layer Perceptron

The training procedure for the MLP is called *Backpropagation*. It is a process in which the loss value is calculated to measure the error rate of the output. The loss value is used to adjust the weights of the neural network in the reverse sequence of the prediction process. Gurney et. al.(32) further detail the Perceptron and the Multi-Layer Perceptron and how the learning on each of them occurs.

When training MLPs, some problems may arise. Overfitting and Underfitting, are, respectively, learning to match the exact pattern of the training data, and not approximating (or learning) the desired pattern enough, in both cases, the network fails to generalize to data outside of the training set. For Overfitting, there are many techniques that mitigate this problem, such as

dropout (when some neurons are randomly deactivated when training), and cross-validation (split training set in chunks and training with random chunks). Underfitting, on the flip side, may mean that the complexity of the model is too small for the train set or that the training data is insufficient.

3.2 Convolutional Neural Networks

The concept of Neural Networks can also be applied to computer vision, by combining the concept of convolutions and neural networks, Kunihiko Fukushima created the precursor of modern Convolutional Neural Networks (CNNs), the "neocognitron"(33) in 1980.

To understand CNNs, one must first understand the convolution operation, used in many image processing techniques.

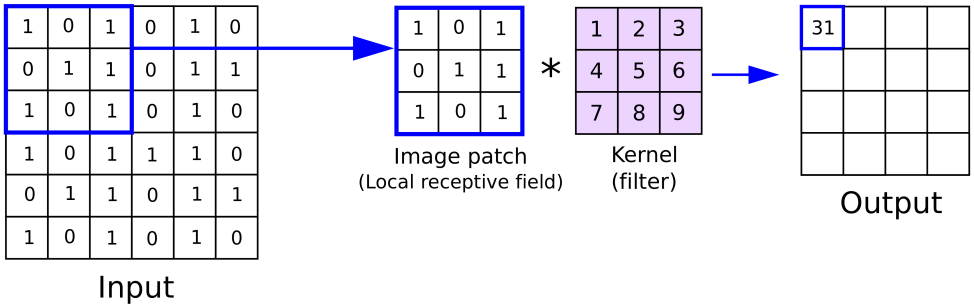


Figure 3.3: The Convolution operation, is the main operation behind Convolutional Neural Networks. Image author: Anh H. Reynolds[1]

Convolutions consist on applying a filter (or mask), a matrix of values, to an image. As exemplified in Figure 3.3. The operations consist of multiplying each value on the mask by the equivalent pixel (on the current image patch) in the image, then summing all results of these multiplications, this will be the new pixel value of the resulting image/feature map.

In CNNs, each value of the filter is learned, as if the weights to be learned in the neural networks are now the values of the convolutional mask. Each convolution layer has its learned weights for its filters, therefore each convolution layer will process the inputs even further, each layer passing its output to the next.

CNNs also use an operation called Pooling in order to reduce the size of the input of a layer (downsample), and consequently speed up computation, by "distilling" the features they become more robust to noise.

[1] https://anhreynolds.com/blogs/cnn.html

The two most common methods of pooling are average and max pooling. Max pooling takes the max value of each neighboring neurons/inputs while average pooling is the average value of each neighboring neurons/inputs. As represented in Figure 3.4.

There are also two different ways to perform Pooling operations. Local pooling reduces the output of the previous neurons/inputs per channel. Global pooling combines values of previous neurons/inputs across dimensions, or channels, in the feature map.

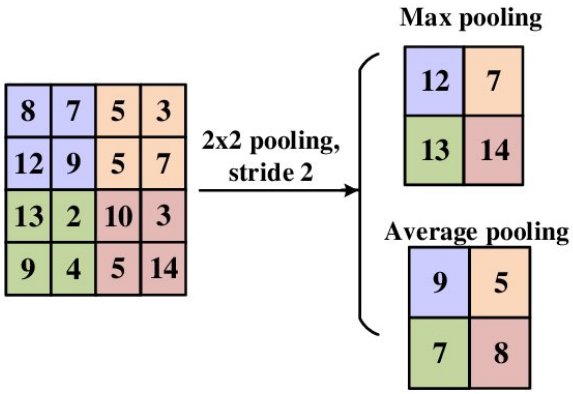


Figure 3.4: Local Max and Average Pooling representation. Image authors are Yingge et. al.(1)

In the first stage of traditional image classification, CNN is comprised of convolutional and pooling layers, extracting and distilling characteristics as the layers get deeper into the model, this is called the feature extraction stage. Then, the features are input to the classification stage, which is usually a fully connected layer of neurons (a Multilayer Perceptron). The classification stage outputs the predicted class. This generic image classification CNN is shown in Figure 3.5.

As convolutional layers get deeper, the level of abstraction also gets higher, as an example, in a generic image classification CNN, the last layers the features may represent more abstract concepts, such as the presence of objects and complex shapes such as cars. These abstractions depend on what images the CNN was trained on and what is it supposed to classify.

Once a CNN is trained with success, it should have learned representations as features that allow it to differentiate between classes. By training a new classifier on the already existing learned concepts (features) in the last convolutional layers, one can modify this generic CNN to classify between dogs and cats or types of cars. One could also use the same trained generic CNN and continue training it on a different task, using the already learned concepts

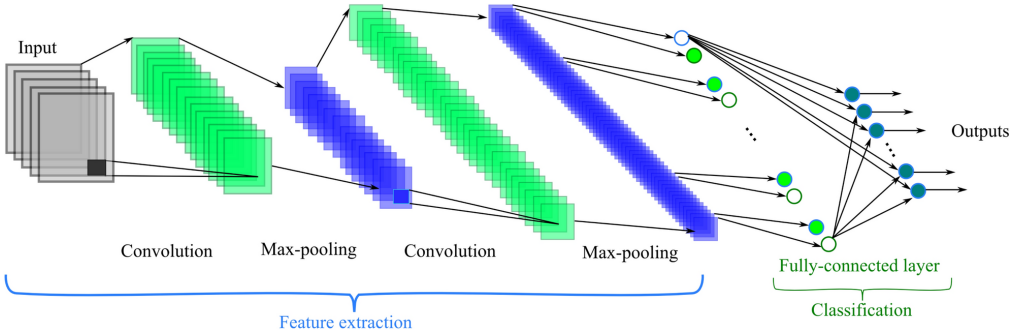


Figure 3.5: A generic image classification CNN architecture represents both the feature extraction stage and the classification stage. Note that as the inputs progress through the CNN the width and height of the inputs become smaller, but the number of feature channels/dimensions increases. Image authors are Khozeimeh et. al.(2)

as a head start, the CNN would also learn more specific concepts for this task as the training continues. This is called transfer learning.

One of the most popular datasets and challenges for CNNs is The ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[2](34), in 2012 it hosted the work that ushered a boom in CNN research and development when AlexNet achieved a top-5 error of 15.3% in the ILSVRC, more than 10.8 percentage points lower than that of the second place. The ImageNet dataset has more than 100.000 "synonym sets" which are sets of words or phrases that represent the image. The dataset holds multiple challenges for tasks such as Image classification, Single-object localization, and Object detection, each of the subsets for these tasks has 1000 classes (or objects).

A CNN trained on the ImageNet dataset can learn a wide variety of abstractions, from cars to dogs, because of the wide scope of their image classification task. This makes CNNs pre-trained in the ImageNet dataset especially performant as transfer learning models (35).

With the success of CNNs, researchers started modifying and applying these models to other domains, such as audio, time series, and natural language processing.

The equivalent of the ImageNet dataset for the audio classification is the Audioset[3] (36). It is an ontology of 632 audio classes and 2,084,320 human-labeled 10-second sound clips collected from YouTube videos. Its classes range from human and animal sounds, musical instruments and genres, and common everyday environmental sounds.

With the advent of deeper CNNs, one problem also surfaced: The vanishing gradient problem, which occurs when the error propagation makes

[2] http://www.image-net.org/
[3] https://research.google.com/audioset/

the training diverge, the values of weights become too small. To avoid this problem, there are multiple techniques, such as the Rectified Linear Unit (ReLU) activation function (37), and lower the learning rate, thus taking smaller steps when adjusting the weights.

3.3 The VGG Convolutional Neural Network

The VGG Convolutional Neural Network (38) was designed for the ImageNet Challenge in 2014, where it won first and second place in localization and classification tasks. Its input is a 224×224 RGB image. The main contribution of this network is that is showed that even with a very small receptive field (3×3, which is the smallest size to capture the notion of left/right, up/down, and center, by increasing the depth of a network, it could still outperform all other CNN based methods at the time.

This architecture has 6 configurations with different depths that are connected to two Fully Connected (FC) layers, two of 4096 channels and a final one with 1000 channels (The Image Net challenge had 1000 classes), followed by a soft-max layer that outputs the predicted class. The configuration of the fully connected layers is the same in all configurations. All hidden layers used rectification (ReLU) (37) non-linearity. Figure 3.6 shows the most popular variation, the VGG-16 (Configuration D), and its layers, as described above.

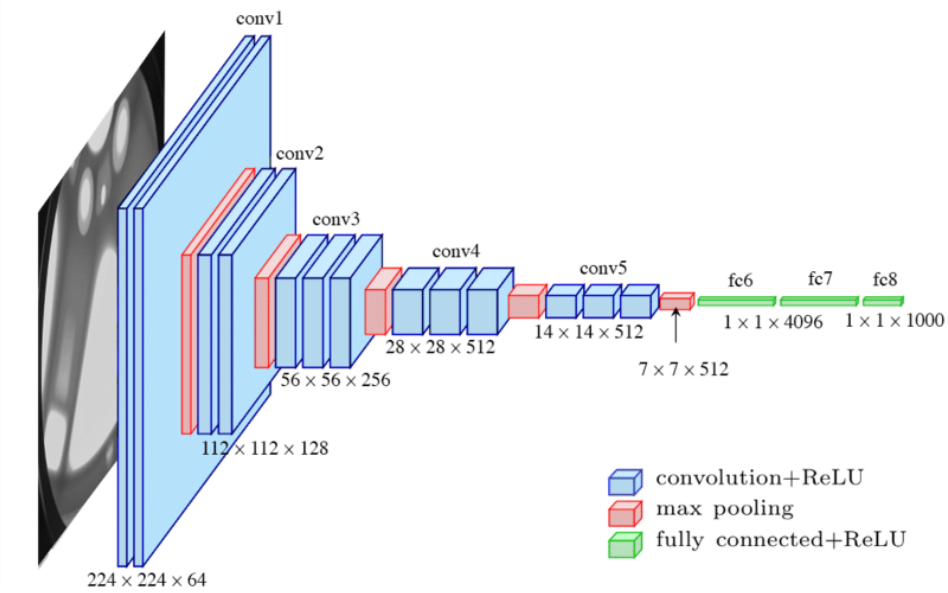


Figure 3.6: VGG-16 architecture, image authors are Ferguson et. al.(3).

3.4 VGGish

The VGGish is a variation of the VGG-11 (Configuration A), created by the authors of the YouTube8m dataset (39), with some modifications to perform audio (spectrogram) classification and embeddings generation).

Specifically, the input size was modified to 96×64 for log Mel spectrogram audio inputs. The last group of convolution and max pool layers was removed. In order to create a compact embedding layer, the 1000 channel wide FC layer at the end was changed to a 128-wide FC layer. This final layer does not have a non-linear activation.

3.5 The Inception Convolutional Neural Network

The Inception Convolutional Neural Network or GoogLeNet, (4) was designed for the ImageNet Large-Scale Visual Recognition Challenge in 2014, it features many techniques in order to increase the efficiency of deep CNNs.

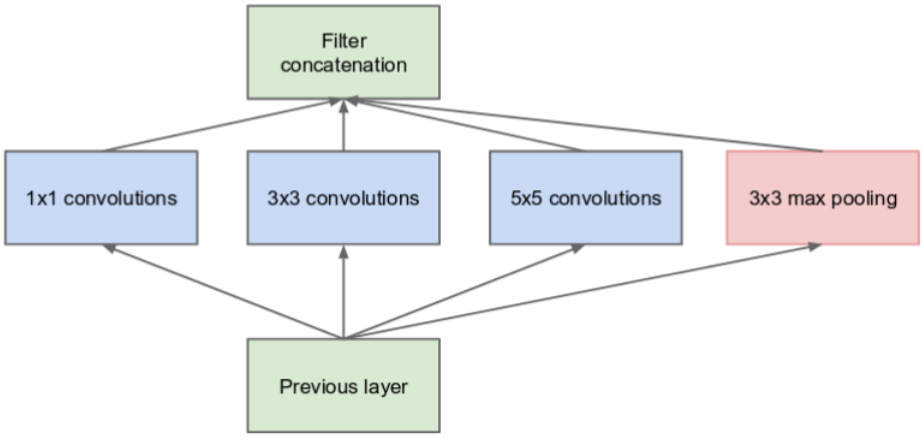
In order to achieve this increased efficiency, the authors created a module to capture as much information as possible, both in the local and global contexts, by using multiple kernel sizes in the same convolutional layer. To optimize for computational cost and speed, the creators also avoided naively stacking layers, for it is computationally expensive.

The solution proposed by the authors of the inception CNN is to use compute multiple filters at the same level, with varying convolutional filter sizes.

The "Naive" inception module, as shown in Figure 3.7 consists of a convolutional layer using 3 different filter sizes, 1×1, 3×3, and 5×5. Along with a max-pooling operation. The outputs are then concatenated by the end of the inception module.

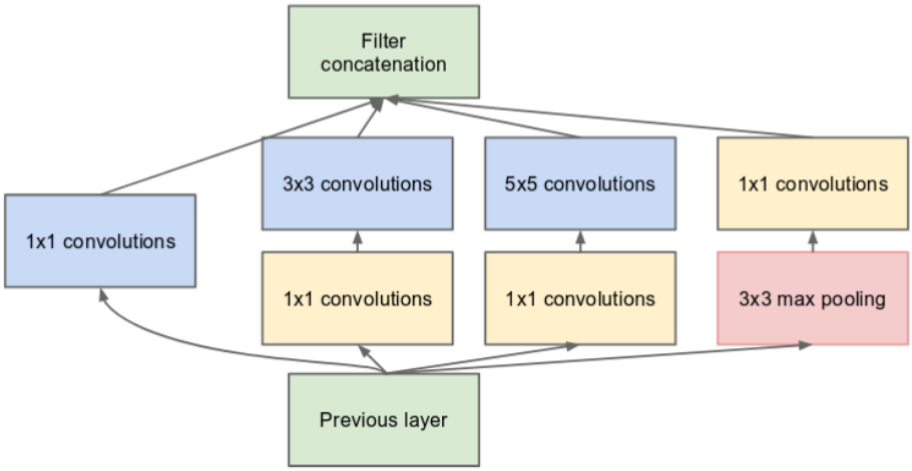
In order to reduce computational costs further, the authors added 1×1 convolutions after the max-pooling step and before the 3×3 and 5×5 convolutions. By doing this the authors reduce the amount of processing done by reducing the number of input channels before the convolutions. This improvement was named "The inception module with dimension reduction" and it is represented in Figure 3.8.

By stacking 9 inception modules with dimension reduction and using 2 intermediate classifiers, essentially computing prediction values and using these values to compute auxiliary losses, which are then used to compose the final loss in the training process in order to avoid the vanishing gradient problem.



(a) Inception module, naïve version

Figure 3.7: The "Naive" inception module, the image authors are Szegedy et. al. (4).



(b) Inception module with dimension reductions

Figure 3.8: The inception module with dimensional reduction, the image authors are Szegedy et. al. (4).

As shown in Figure 3.9 It is still deeper (it has 22 convolutional layers) than the deepest VGG configuration (with 19 convolutional layers).

InceptionV2 and InceptionV3 networks (40) For the inceptionV2 computational efficiency was improved by factorizing the convolutions, convolutions with 5×5 size kernels were factorized into two 3×3 sequential convolutions, this improves computability (because 3×3 convolutions use 2.78 times fewer operations than 5×5) while actually improving performance. They also factorized the 3×3 convolutions into one 1×3 and 1×3 convolutions. These factorized

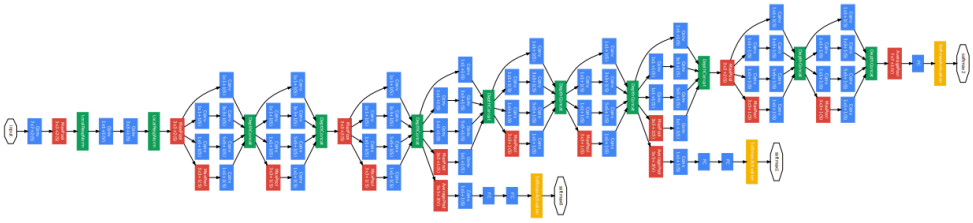


Figure 3.9: The complete inception convolutional neural network, image authors are Szegedy et. al. (4).

convolutions are performed on the same input to avoid excessive dimension reduction (information loss).

The InceptionV3 CNN used all the upgrades of the InceptionV2, and improved performance further by incorporating the RMSProp Optimizer (41), factorized 7×7 convolutions (1×7 convolutions followed by 7×1 convolutions.), batch normalization in the Auxillary Classifiers, and Label Smoothing (It is a modification in the loss function that prevents the network from having high confidence in a class) for preventing overfitting.

The Inception CNNs continued to improve further with InceptionV4 and Inception-ResNet (42), but we will not detail them here for they are not used in the scope of this work.

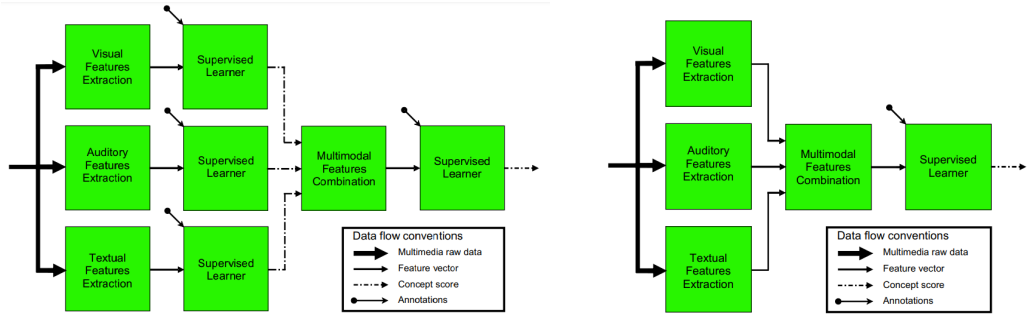
3.6 Feature Fusion

Since we are using information from two different domains, image, and audio, it is also important to think about how one can fuse information from both these domains while losing as minimum information as possible.

Snoek et al. (5) present two main strategies for information fusion in semantic video analysis:

– *Early fusion* methods (Figure 3.10a), which work directly with the extracted features.

– *Late fusion* methods (Figure 3.10b), which operate on classification outputs from specialized models.



(a) In the late fusion approach, in it, there is a machine learning model for each unimodal feature and a final model to fuse the outputs of each unimodal model.

(b) The early fusion approach uses a single multimodal machine learning model to both aggregate and classify all features.

Figure 3.10: The late and early fusion methods for feature fusion. Image authors are Snoek et. al. (5).

In the work by Snoek et. al.(5), the Late fusion approach tends to give better performance on most semantic concepts (multilabel video classification) at the cost of increased computability costs. However, the authors also conclude that the late and early fusion approaches should be compared are per concept (in a multilabel situation).

4
Sensitive Content Dataset

In this chapter, we present how our *dataset* was collected, how it is structured, how the features were extracted, and what metrics we recommend for the main task of this dataset. This dataset is partially weakly annotated, as some of the Safe and all of the Sensitive videos were user-labelled.

4.1
Dataset Collection and Assembly

4.1.1
Safe content

For *safe content*, we chose to sample instances from Youtube8M[1] (39). We chose this dataset because of its size (8 million videos) and because of the wide variety of video classification challenges it supports. We selected 55.000 random videos inside each of the 24 top-level categories, proportionally to the original dataset distribution. As there was no limit for the sample size, we did not have to use any rules to keep small categories.

We successfully collected 50,988 Youtube videos with metadata. 4,012 of the 55,000 sampled videos failed to download or were unavailable. These videos are manually (strongly) labelled, as they were assembled from the yt8m dataset.

We also collected 8,663 videos from Youtube, hereby referred to as “cherry-picked" safe videos, those videos were selected for the purpose of increasing the amount of “hard" videos, as done in (13), which are videos that could possibly be misclassified as sensitive, such as Mixed Martial Arts (MMA), breastfeeding, pool parties, beaches and other videos that have a higher amount of skin exposure. The amount of cherry-picked videos collected is listed by their respective query in Table A.2. The collection was made by automated means, a script automatically searched and tried to download all videos from the first 100 result pages of each query. This means that the main tags, or labels, of the “hard" safe videos are user-generated, or weakly labelled.

[1] https://research.google.com/youtube8m

4.1.2 Sensitive content

For *sensitive content*, we collected pornography and violent videography (hereafter referred to as *gore*) from websites. All videos with sensitive content are weakly labelled, that means that all labels are either user-generated or manually labelled by its site's moderator.

For the pornography, we to sample videos from the XVideos[2] database. We chose this source because of the database size (7 million videos) and because of the amount and variety of annotations. In this database, each video has one main tag, totaling 60 main tags, and tags (user-created).

We sampled 55.000 random videos in each tag in equal proportion to the original distribution in these tags. In particular, to prevent tags with fewer videos from disappearing, we have defined that the minimum sample for each tag is the size of the smallest tag. The smallest tag was 'ASMR' with 63 videos.

We successfully collected 54,549 pornography videos with metadata. 451 of the 55,000 sampled videos failed to download or were unavailable. We also collected 10,519 "hard" videos from XVideos, specifically videos that have low skin exposure, fully dressed people, latex costumes, and cosplay.

For the gore content, we used a web crawler to extract 2,356 gore videos from various websites dedicated to gore media, such as, BestGore[3] and GoreBrasil[4]. As these videos were harder to find and collect, we collected all available videos from each website, no sampling method was applied. Most videos did not stay online for more than one week. As there was no contact with the videos and the videos did not have any tags, all metadata collected was the title of the videos.

Not all video features were successfully extracted for multiple reasons, such as corrupt data, unknown format, and missing audio. For those videos with missing audio or image, the features were still generated, but their respective modal feature were zeros. Those videos which did not have any features successfully extracted were removed from the dataset.

We also removed any duplicated videos that were detected, for duplicate video detection we used, we matched either id, title, or checksum.

[2] https://info.xvideos.com/db
[3] https://www.bestgore.com/
[4] https://www.gorebrasil.com

4.2 Dataset Structure

Our *dataset* is structured into two main classes (or macro-classes): "safe" videos and sensitive videos. The sensitive macro-class is composed of two micro-classes: Pornography and Gore. The safe class is composed of videos from Youtube. Finally, each of the micro-classes has main tags, which are the same main tags from their original metadata in the website, if available. Each instance (a video) of the dataset may also have tags, which are a list of tags that represent the video. The general structure and organization of our dataset are represented in Figure 4.1.

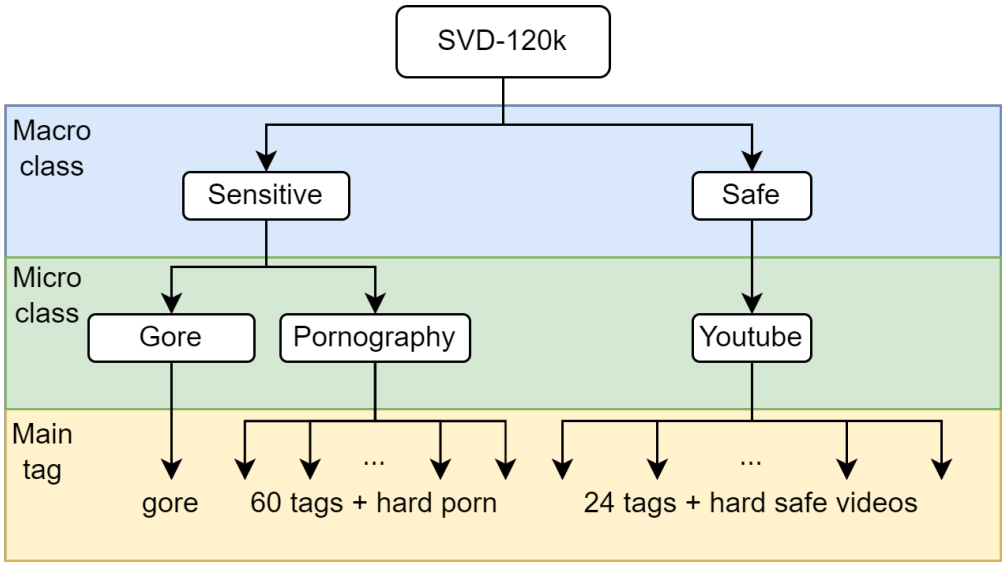


Figure 4.1: Dataset tree structure

There are 59,651 safe videos and 67,424 videos with sensitive content. Table 4.1 presents the general statistics of our dataset, such as total duration (hours, minutes, and seconds) of all videos, total (uncompressed) features size, and tag coverage, the amount of videos with a main tag (videos may also have tags but no main tag). If played in real-time, a person would take approximately 1 year and 127 days straight to flag all videos in this dataset.

The YouTube micro-class has 25 main tags, as presented in Table A.3, in Appendix A, each of the videos in the "hard safe videos" main tag, has one tag, which was the query word used to collect it.

The Pornography micro-class has 60 main tags, as presented in Table A.4, Appendix A, these main tags were defined by the database creators, and the tags were user created. The instances in this micro-class also have user-created titles. There are no main tags or tags (but the 'gore' tag) on the Gore micro-class as they were not available on their site. All instances, however, have user-created titles.

Table 4.1: General statistics of the two main classes of the dataset. Tag coverage is the amount of videos with a main tag, videos may also have tags but no main tag.

	Sensitive	Safe
Video Count	67424	59651
Total Duration	6953:27:41	4852:53:31
Mean Duration	00:06:11	00:04:52
STD Duration	00:04:12	00:03:26
Max Duration	00:30:55	00:30:55
Min Duration	00:00:05	00:00:05
Total Size	1,2TiB	2,2TiB
Mean Size	19,3MiB	39,0MiB
STD Size	35,4MiB	42,3MiB
Features Size	519,4GiB	376,8GiB
Tag coverage	63036	59651
Tag coverage (%)	93,4919	100,0000

By using macro and micro classes of this dataset, our dataset also supports other tasks, other than the binary classification of sensitive content, such as:

- Multi-label classification (or tagging) of pornographic videos;
- Multi-label classification of "Safe" (Videos that do not contain sensitive content);
- Binary classification of extremely violent (gore) videos;
- Binary classification of pornography.

4.2.1 Dataset Distribution

The dataset will be distributed as extracted and processed visual and audio features from the videos. Each instance (features from a video) is associated with a id, a label, and a sequence size. We will not distribute raw video data, but we are open and plan to include other feature extraction methods in our dataset. General details on the dataset distribution are available in Appendix B.

4.2.2 Dataset Balancing

For experimenting, we equally balanced both main labels (sensitive/improper and safe/proper), so that both main classes have the same number of instances. One could also choose not to balance both classes equally, since our

Table 4.2: Granular statistics of the dataset: Videos collected from Youtube, pornographic videos, and gore videos.

	Pornography	Gore	YouTube
Video Count	65068	2356	59651
Total Duration	6900:17:38	53:10:02	4852:53:31
Mean Duration	00:06:21	00:01:21	00:04:52
STD Duration	00:04:10	00:01:26	00:03:26
Max Duration	00:30:55	00:16:56	00:30:55
Min Duration	00:00:05	00:00:05	00:00:05
Total Size	1,2TiB	15,8GiB	2,2TiB
Mean Size	19,8MiB	6,9MiB	39,0MiB
STD Size	35,9MiB	13,9MiB	42,3MiB
Features Size	515,3GiB	4,1GiB	376,8GiB
Tag coverage	63036	0	59651
Tag coverage (%)	96,8771	0	100,0000

main metric already takes label imbalance into account. Additionally, when removing excess sensitive content (while balancing), we removed only pornography videos in order to not lower the number of gore videos.

4.2.3 Dataset splits and Test sets

We hold out our dataset for testing our approach: 10% of the safe videos, then 10% of gore videos, and sample pornography videos to match the number of safe videos minus the amount of gore test samples so that the test subset has a balanced amount of sensitive and safe videos while keeping a valid amount of gore videos. For the micro-classes that have multiple main tags (Youtube and Pornography), we took stratified samples based on the number of each main tag in the dataset. The number of instances by micro-class sampled is presented in Table 4.3.

As a complementary test dataset, we selected the Pornography-2k dataset (18), which contains 1000 non-pornographic videos and 1000 pornographic videos. Those non-pornographic videos are comprised of "hard" and "easy" videos according to the likelihood of misclassification. Some examples of "hard" videos are those with high amounts of exposed skin, such as swimming and sumo fighting videos. Its general statistics are shown in Table 4.4.

4.3 Metrics

To evaluate each experiment and our approach, we will use Precision (P), Recall (R), and, most importantly, the weighted F2 score. In this section, we

Table 4.3: Test subset statistics.

	Pornography	Gore	YouTube
Video Count	5732	236	5968
Total Duration	574:03:29	05:22:20	443:35:36
Mean Duration	00:06:00	00:01:21	00:04:27
STD Duration	00:03:44	00:01:48	00:02:32
Max Duration	00:30:29	00:16:56	00:29:01
Min Duration	00:00:05	00:00:07	00:00:07
Total Size	80,2GiB	1,5GiB	204,8GiB
Mean Size	14,3MiB	6,6MiB	35,1MiB
STD Size	25,6MiB	14,4MiB	35,0MiB
Features Size	42,6GiB	424,1MiB	34,5GiB
Tag coverage	5695	0	5968
Tag coverage (%)	99,3545	0,0000	100,0000

Table 4.4: Pornography-2k dataset statistics.

	Porn	Non-Porn
Video Count	1000	1000
Total Duration	100:30:32	40:26:06
Mean Duration	00:06:01	00:02:25
STD Duration	00:05:49	00:02:17
Max Duration	00:33:40	00:20:16
Min Duration	00:00:05	00:00:02
Total Size	26,4GiB	18,5GiB
Mean Size	27,0MiB	18,9MiB
STD Size	31,1MiB	21,9MiB
Features Size	7,6GiB	3,1GiB
Tag coverage	0	0
Tag coverage (%)	0	0

present a contextualized explanation of these metrics.

In the context of sensitive content detection, *true positives* are videos predicted as sensitive and are in fact, sensitive. Likewise, *true negatives* are videos predicted as safe and are indeed safe. *False positives* are videos predicted as sensitive, but were safe, the same goes for *false negatives*, which are videos that were predicted as safe, but were actually sensitive.

Precision (Equation 4-1) measures how many videos predicted as sensitive (both true positives and false positives) are truly sensitive. The Recall (Equation 4-2) measures how many truly positive videos were correctly identified.

$$P = \frac{TP}{TP + FP} \qquad (4\text{-}1) \qquad\qquad R = \frac{TP}{TP + FN} \qquad (4\text{-}2)$$

Where TP, TN, FP, and FN denote the examples that are true positives, true negatives, false positives, and false negatives, respectively.

$$F_\beta = (1 + \beta^2) \times \frac{P \times R}{(\beta^2 \times P) + R} \tag{4-3}$$

The F_β-score, defined in Equation 4-3, evaluates the classifier by the harmonic mean between Precision and Recall. To account for label imbalance, after calculating the F2-score metrics for each label, we find their average weighted by support (the number of true instances for each label).

Most related works, such as (12, 14, 15), use either F1-score ($\beta = 1$) or F2-score ($\beta = 2$) metrics as their main evaluation metric. While the F1-score represents a balanced performance metric, the F2-score gives twice more weight to the recall than to precision, which means that the metric is more focused on the recall of a solution.

In this work, the F2-score represents an overall performance metric, while the precision and recall metrics can give insights on what the classifier model is doing better and what to improve. We chose the weighted F2 score as our main evaluation metric because when detecting sensitive content it is more important to predict a truly sensitive video than to predict a safe video as sensitive.

5
Method

In this section, we detail our method for sensitive content detection in video. We split our approach into three parts: feature extraction, feature fusion, and feature classification, as illustrated in Figure 5.1.

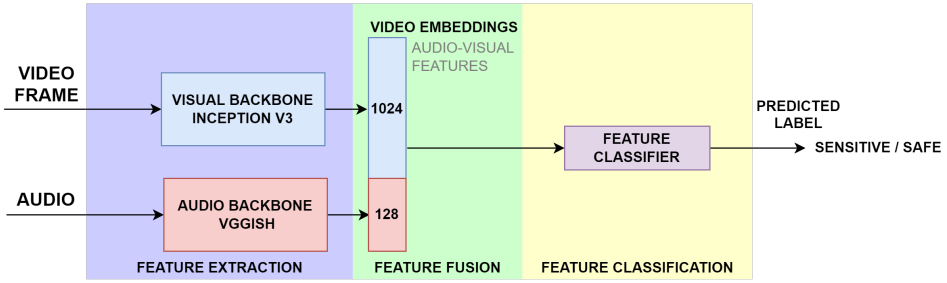


Figure 5.1: Our approach to sensitive video detection (binary classification)

In the feature extraction stage, firstly we split the frames and audio from the video; then, for each media, we use a CNN to extract the features (or embeddings) from each simultaneous video segment. In the second stage, Feature Fusion, we concatenate both audio and frame features. If the classification model is not sequential, we also aggregate the features in this stage. Finally, in the feature classification stage, we feed one of the classification models to be experimented with.

5.1
Video Embeddings Extraction

CNNs tend to learn low-level features (*e.g.*, in the visual domain: edges, corners, contours) at their first layers. At the intermediate and final layers, the combination of these features helps to extract more complex features, resulting in a vector of continuous values, referred to as *embeddings*, that might be used for classification and other tasks. In this work, we use two benchmark CNNs to extract both image and audio *embeddings* by using a transfer learning technique (43).

By using the feature extraction method created for the Youtube-8m benchmark, we can test a feature extraction method that is powerful enough to represent features that can be in multiple tasks, such as multi-label video classification, video recommendation, and human activity recognition.

"Since the video-level representations are unsupervised (extracted independently of the labels), these representations are far less specialized to the labels associated with the current dataset, and can generalize better to new tasks or video domains." (39)

In order to validate our dataset, we used the same feature extraction method used in the Youtube-8m dataset challenges (39), both networks were pre-trained and frozen. They were not retrained for application-sensitive content classification. This gives future works an opportunity to develop even more efficient and smaller feature extraction networks for this specific task.

As described in (39), To generate image frame features and audio features we decode each video at approximately 1 frame-per-second. For the image frame features, we used an InceptionV3 network (40) pre-trained on the ImageNet[1] dataset. We also use a variation of the VGG network (38), called VGGish, with pre-trained weights in the Audioset[2] dataset to extract the audio embeddings.

Each of these CNNs was used as published by their authors; We adopted their respective versions for feature embedding generation, on which the only modification was the removal of classification layers in both CNNs to obtain their respective embeddings.

5.2 Feature Fusion

Once we have the features (embeddings) from both image and audio, we should make a decision about which method is best to fuse the information from these different domains, as described in Chapter 3, Section 5.2.

Although the Late fusion approach tends to give better performance (5), it comes with increased computability costs.

In this work we have high abstraction level features and are making baseline models for this dataset, because of that, we opted to investigate the approach with the lesser computability cost, which is to train a single model on the concatenated features from both media inputs (Early Fusion).

In order to create the final embeddings, we concatenate both image and audio embeddings extracted in the same frame and audio window. This generates a sequence of the same size as the number of seconds of the video. After this concatenation, each time-step has 1,152 features: 128 audio features and 1024 frame features.

[1] `http://www.image-net.org/`
[2] `https://research.google.com/audioset/`

Notice that with this approach, the video is transformed into a time series, and to use it in non-sequential models (*e.g.* SVM, KNN, and MLP) we need to turn this sequence into a single feature vector that represents the whole video. In our setting, we did that by taking the average, median, standard deviation, min, and max values for each feature to represent the entire video. In summary, we turn the sequence of features with size n and shape n by 1,152 into a single feature with shape 1 by 5,760.

5.3 Classifiers

For the feature classification task, we investigate both sequential models (which use the extracted embeddings in a time series format), and non-sequential ones (which use a single aggregated embeddings vector). We want to experiment with both approaches in order to investigate if a more compact format, such as the single embeddings vector, can yield results at least as good (or even better) than the full feature sequence data. As an example, one can think of a long video that has a pornographic scene in one second out of its entirety. In a non-sequential representation of the extracted features, this short pornographic fragment could be left "hidden" among the other non-pornographic frames of the video, as illustrated in Figure 5.2. In a sequential

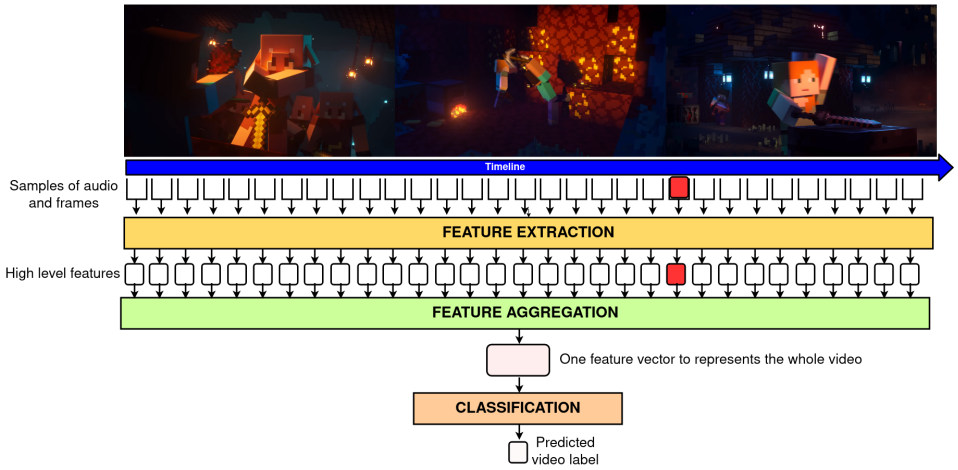


Figure 5.2: Sequential features with aggregation, the sensitive scene (red) might vanish among the other scenes during aggregation.

representation, although time-series classifiers usually output a prediction after reading the entire sequence, the embedding vectors of each second of the video would not be aggregated and thus could be analyzed section by section, as illustrated in Figure 5.3.

Although a sequential representation contains possibly much more redundant data than the non-sequential one, it could give the sequential classification model an important edge of detail over the less granular non-sequential

Figure 5.3: Sequential features with no aggregation. The output, after reading the entire sequence, can also be susceptible to information vanishing.

ones. Furthermore, with a sequential approach, one could train a model to classify the whole video, but when at testing time, to stop classification as soon as the current sample is classified as sensitive, allowing for near-real-time classification.

For the sequential classification model, we chose the Long Short-Term Memory (LSTM)(19) networks. It has been a commonly used time series classification baseline model.

For the non-sequence models, we chose Support Vector Machines (SVM) (44) , K-Nearest Neighbors (KNN) (45), and Multilayer Perceptron (MLP) (46). Among all of the experimented models, the *Support Vector Machine (SVM)* is the most used in the literature. It is a classification model in which the data is mapped into a higher dimension input space, where an optimal separating hyper-plane is constructed. We used the RBF kernel and C=1. The second model, *K-Nearest Neighbors* uses distance measure between training samples so that the k-nearest neighbors always belong to the same class, while samples from different classes are separated by a large margin. It was chosen because it is used also by related work, although it is a simple classification method. The third model is the *Multilayer-perceptron (MLP)*, which contains layers of nodes: an input layer, an output layer, and various hidden layers in between. This one was selected because it is also commonly used as a final classifier on deep neural networks. For model evaluation, we performed 20-fold cross-validation for all baseline models.

5.4 Proposed Analysis

In Chapter 6, we evaluate the performances of baseline classifiers over the video *embeddings* that were extracted from our dataset, described in Chapter 4. Then, we choose the best performing classifier during the validation stage

and test its performance on the *test sets*. We designed a set of cases that might help us find insights and assess the performance and shortcomings of our dataset and approach.

Our objective with these analyses is to attest to the quality of our dataset and approach to detecting sensitive content on video.

We chose not to perform extensive hyperparameter optimization (fine-tuning) on the baseline models, since this work already aims at validating the dataset and the transfer learning-based feature extraction method. Although we performed hyperparameters changes on the SVM model, the most since it is most sensitive to hyperparameters optimization. In future works, we will create a specific model for this task in this dataset and compare it with fine tuned baseline models.

(**E0**): Testing only on image features: In this analysis, we evaluate our approach on our test subset using the visual (frames) features only.

(**E1**): Testing only on audio features: In this analysis, we evaluate our approach on our test subset using the audio features only.

(**E2**): Testing pornography using audio-only videos: In this analysis, we evaluate our approach to the Pornography-2k dataset using the audio features only.

(**E3**): Testing pornography using only image features: In this analysis, we evaluate our approach to the Pornography-2k dataset using the visual features only.

In the next chapter, we present and discuss the results of our baselines and report each analysis.

6
Results

Having performed 20-fold cross-validation, we collected all metrics through all the folds. Table 6.1 presents mean, standard deviation, min, and max. The full results for each fold are available in Table A.1, Appendix A.

Table 6.1: Weighted F2-Score (in percentage) for each model across 20-Fold Cross Validation.

	MLP	LSTM	SVM	KNN
count	20,0000	20,0000	20,0000	20,0000
mean	**99,0743**	98,9899	98,8993	96,4738
std	0,1349	0,1174	0,1347	0,2915
min	98,7945	98,7943	98,5445	95,6932
25%	99,0073	98,9175	98,8499	96,3585
50%	99,0962	98,9848	98,9174	96,4108
75%	99,1708	99,0684	98,9756	96,6564
max	99,2885	99,1915	99,0836	96,9448

6.1
Model comparison

Comparing the models in Table 6.1 the model with the highest mean weighed F2-Score across folds is the Multilayer Perceptron model. To compare the models and test if their results are statistically different.

Afterward, we test if the model with the best model has a statistically significant difference from the second-best by performing a posthoc pairwise hypothesis test.

Figure 6.1 shows the difference in the distribution of results of each model. The simplest model, K-Nearest Neighbors, has the most different from the models. The three other models, however, are relatively close to each other.

To determine what test is better suited for our data distribution (the best-weighed F2-Score), we checked our data for normality and outliers. Which are frequent assumptions for different hypothesis tests. For the normality assumption, we used probability plots, as shown in Figure 6.2 to test if the data is normal.

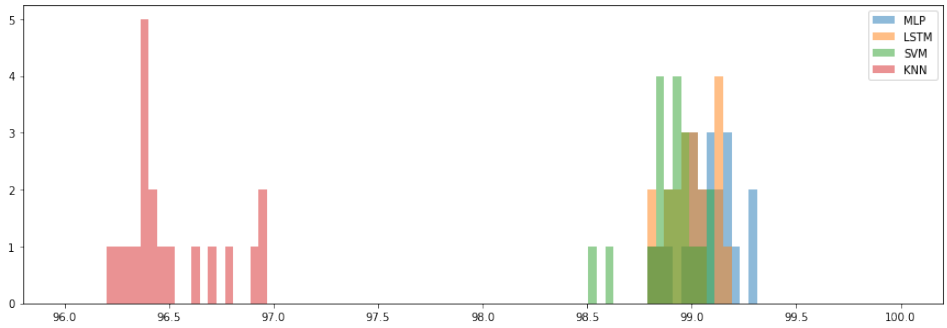


Figure 6.1: Histogram of the results of each model throughout the 20-fold cross-validation.

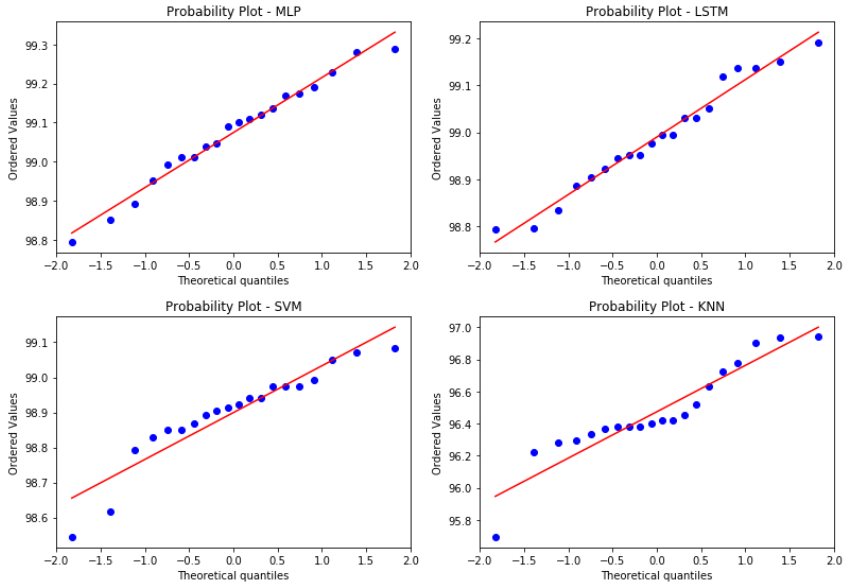


Figure 6.2: Probability plots for each model show the data's quantiles against the quantiles of a theoretical distribution (the normal distribution).

As shown in Figure 6.2, the KNN and SVM models differ from a normal distribution.

To validate what is shown in the probability plots, we also tested the normality assumption with the Shapiro-Wilk test. The null hypothesis is that the data is not drawn from a normal distribution. The alternative hypothesis is that the data is drawn from a normal distribution.

According to the Shapiro-Wilk test, the MLP and LSTM models follow a normal distribution ($p<0.05$). As for the SVM and KNN models, they do not follow a normal distribution ($p>0.05$). So our data does not support tests that require data with normal distribution.

For outliers presence, as seen by the box plot in Figure 6.3, there are outliers in the SVM and KNN data.

When tested with most parametric tests, both outliers and non-normal distributions can bias the results and potentially lead to incorrect conclusions

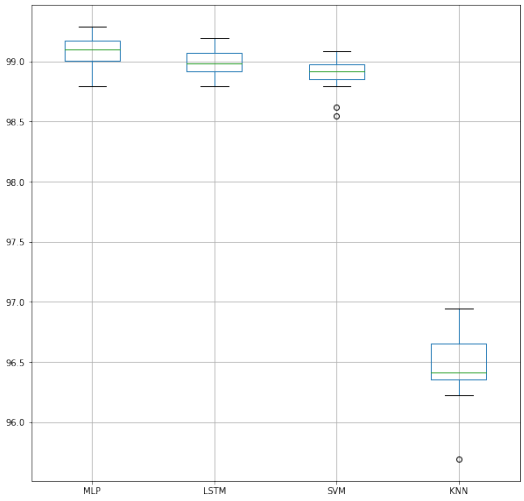


Figure 6.3: Boxplot of the results of each model throughout the 20-fold cross-validation.

if not handled properly.

To choose a test, we chose a non-parametric (sample median) test, because sample medians are less sensitive to outliers, in contrast with the mean or variance-based parametric tests, such as Variance Analysis (ANOVA). We defined the null hypothesis as "All models folds evaluations medians are equal"; The alternative hypothesis is that "At least one model mean rank (median) is different from other groups". We used a two-tailed test since we do not know which model will be higher. We chose our alpha as 0.05. That is a probability of 5% of committing an error, rejecting the null hypothesis when it should be accepted.

We chose the Kruskal-Wallis test (47) as our hypothesis test because it fits all of our requisites (non-parametric, median-based). Our data also supports all of this test's assumptions.

After applying the Kruskal-Wallis test we obtained a p-value of 2.2733e-11, which means that $p<0.05$ and that we can reject the null hypothesis and accept the alternative hypothesis.

To determine which models are statistically different from each other, we performed a posthoc test using the Nemenyi post hoc test (48), because it is a non-parametric (distribution-free) test and our data does not follow a normal distribution. The results from this test are presented in Figure 6.4.

As presented in Figure 6.4, the MLP and LSTM models are not significantly different from each other ($p>0.05$), and the KNN model is statistically different from all other models ($p<0.05$).

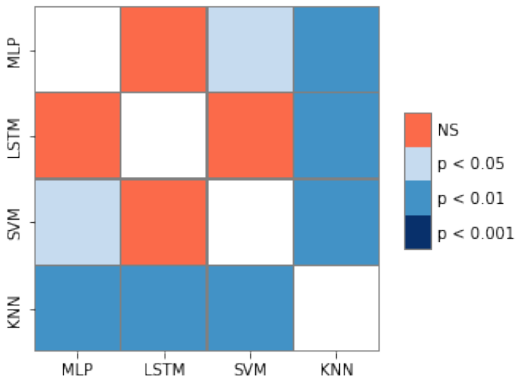


Figure 6.4: Significance matrix. The plot was made with scikit-posthocs(6).

6.2 Tests results

We tested the best performing model, the Multilayer-Perceptron, on the test subset, shown in Table 6.2.

Table 6.2: Test subset results, shown in absolute values).

	precision	recall	f1-score	f2-score	support
Safe	0,9895	0,9906	0,9900	0,9897	5973,0000
Sensitive	0,9906	0,9895	0,9900	0,9904	5973,0000
weighted avg	0,9900	0,9900	0,9900	0,9900	11946,0000

As shown in Table 6.2 and Figure 6.5, the MLP model has performed within the range of the mean of the cross-validation. It can also be noted that the most frequent errors were *false positives* when the model predicted a video as Sensitive when it was in fact, Safe.

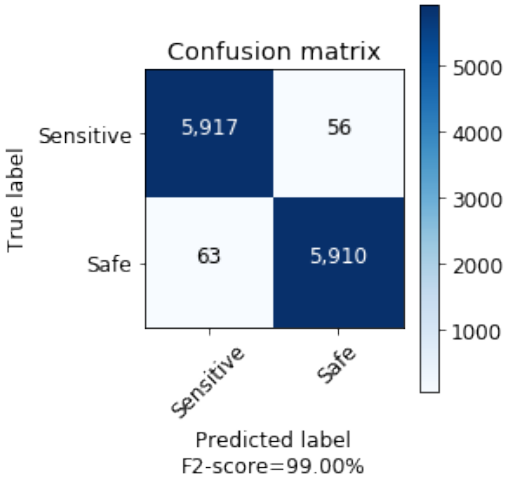


Figure 6.5: Confusion matrix of the predictions of the best model in the test subset.

We also tested our best model in each sub-task, pornography and gore binary classification. For the pornography, shown in Table 6.3, and Figure 6.6a, the most frequent error was the *false negatives*, in which the model predicted that the most samples were predicted as Safe, but were Sensitive.

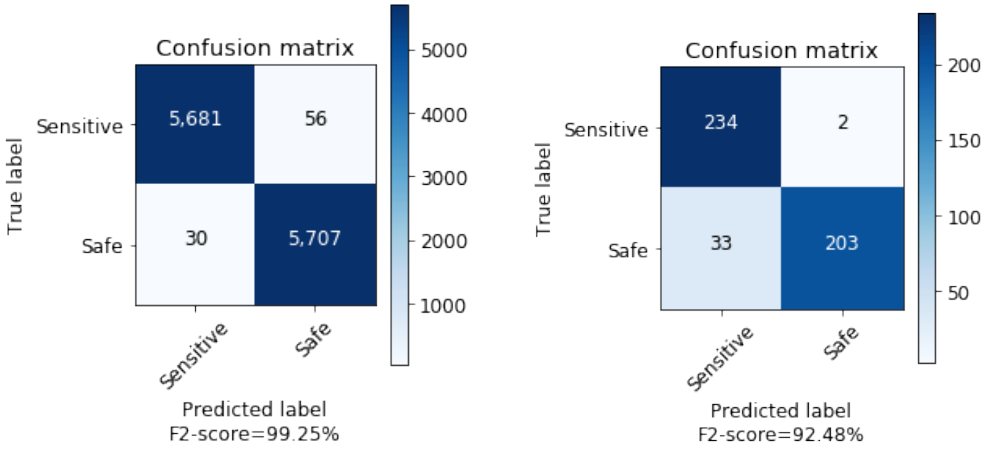
Table 6.3: Results testing pornography only, shown in absolute values).

	precision	recall	f1-score	f2-score	support
Safe	0,9947	0,9902	0,9925	0,9939	5737,0000
Sensitive	0,9903	0,9948	0,9925	0,9911	5737,0000
weighted avg	0,9925	0,9925	0,9925	0,9925	11474,0000

For the gore, shown in Table 6.4, and Figure 6.6b, the most frequent error was the *false positive*, in which the model predicted that the most samples were predicted as Sensitive, but were Safe.

Table 6.4: Results testing gore videos only, shown in absolute values).

	precision	recall	f1-score	f2-score	support
Safe	0,8764	0,9915	0,9304	0,8834	236,0000
Sensitive	0,9902	0,8602	0,9206	0,9661	236,0000
weighted avg	0,9333	0,9258	0,9255	0,9248	472,0000



(a) Confusion matrix of the model on the pornography videos of the test subset.

(b) Confusion matrix of the model on the gore videos of the test subset.

Figure 6.6: Confusion matrices of the best performing model on the pornography and gore subsets.

To evaluate our model and our dataset on the pornography detection (binary classification) task, we also tested our best performing baseline model on a well-known dataset for pornography detection: The Pornography-2k dataset. The results are shown in Table 6.5 and in Figure 6.7. The most common errors were false negatives, in which the model predicts the instance as a Safe, but the true label was Sensitive.

Table 6.5: Test on the Pornography-2k dataset results, shown in absolute values).

	precision	recall	f1-score	f2-score	support
Safe	0,9665	0,8080	0,8802	0,9411	1.000,0000
Sensitive	0,8351	0,9720	0,8983	0,8354	1.000,0000
weighted avg	0,9008	0,8900	0,8893	0,8883	2.000,0000

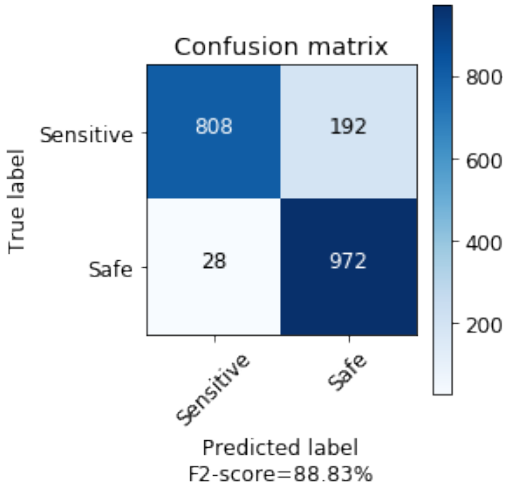


Figure 6.7: Confusion matrix of the predictions of the best model in the Pornography-2k dataset.

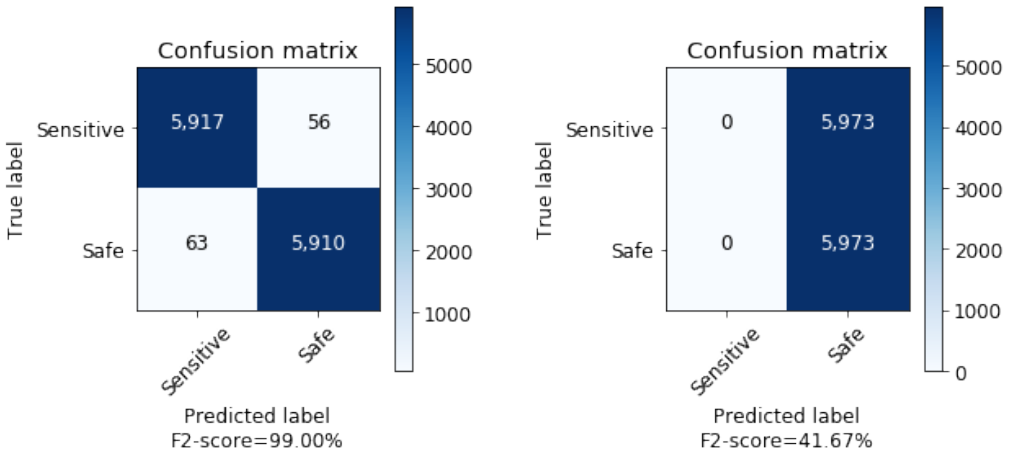
6.3 Analysis cases

As detailed in analysis **(E0)** and **(E1)** (Section 5.4), to further investigate the impact of each multi-modal feature in our best performing model, the Multilayer Perceptron. We tested it on our test subset and on the Pornography-2k dataset, but only using one modal feature at a time. For example, in Figure 6.8a, we tested the MLP model using only visual (frames) features, specifically, were changed all audio features to zero to simulate a video with no audio features.

As observed in Figures 6.8a and 6.8b, our model had the same performance with only visual features, but misclassified all Sensitive videos. This means that the MLP model ignored all audio features for all videos. It is relying only upon visual features, even though there are examples of videos in the dataset, in which the main feature of a sensitive video is audio.

As described in analysis **(E2)** and **(E3)** (Section 5.4), we confirmed the same pattern in the tests with the Pornography-2k dataset, as shown in Figures 6.9a and 6.9b.

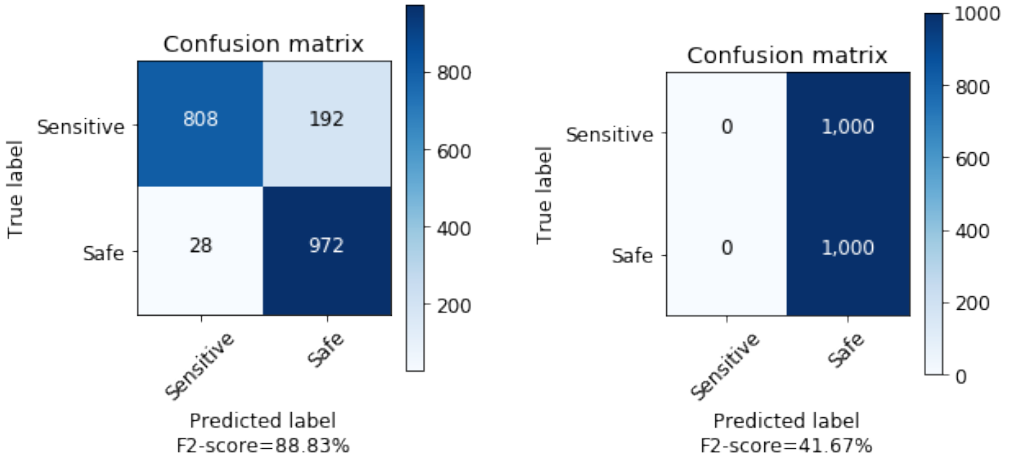
Because of the late fusion approach, in which the model receives both feature types and decides which ones to use the most we were susceptible to



(a) Confusion matrix of the model on the test subset using only image features.

(b) Confusion matrix of the model on the test subset using only audio features.

Figure 6.8: Confusion matrices of the model on the test subset using only one multi-modal feature at a time.



(a) Confusion matrix of the model on the Pornography-2k dataset using only image features.

(b) Confusion matrix of the model on the Pornography-2k dataset using only audio features.

Figure 6.9: Confusion matrices of the model on the Pornography-2k dataset using only one multi-modal feature at a time.

this learning behavior. These results could be due to multiple reasons:

- The difference in the size of visual features and audio features (1024 for visual and 128 for audio);
- This specific model learned to ignore the audio features;
- The audio features did not offer as much differentiation power as much as the visual features in this dataset.
- The audio feature extraction method did not offer as much differentiation power as the visual features.

Using late fusion in the multimodal features could improve our performance in this task because a model is trained for each multimodal feature, assuring the use of each available feature type.

6.4 Discussion

When testing the best performing model, the MLP, there was a little variation in the test subset performance. One explanation for this could be that our dataset is mostly homogeneous (Does not have many variations in videos characteristics). This could also be a consequence of the dataset's sampling, resulting in a test subset similar to the train/validation subset. However, there are steps taken to avoid both these possibilities. The dataset was created with a wide variety of videos within each subclass, such as education and sports for the safe videos, and the test subset was a random sampling that followed the distribution of main tags within each of the subclasses. On the pornography detection task, the results were still within the expected performance and the most frequent errors were false negatives, which is an error we want no minimize the most over false positives. On the gore detection task, there was a significant performance drop, which could be a reflection of the smaller amount of gore examples in the dataset, or could mean that this approach is less adequate for the gore detection task than for the pornography detection task.

When testing the best performing model on the Pornography-2k dataset, there was a significant drop in performance compared to the test subset, this could be due to the model's lack of use of audio features. This could also mean that our dataset misses specific hard instances of either class.

7
Conclusions

In this work, we created a large-scale (110k) video dataset for sensitive content detection and a multi-modal approach to sensitive content detection in video. It uses pre-trained convolutional neural networks and applies an early-fusion feature method, which is simpler than the late-fusion approach since we use a single model to classify both features.

We evaluated our models by testing on a test subset and a popular dataset. We validated our dataset and baseline approach while maintaining similar performance to the existing methods.

It is important to note that our approach is not focused on mobile platforms, therefore memory and disk space were not major constraints.

It is worth mentioning that our overall results on the sensitive content detection are not directly comparable to the related works since their definition of violence does not match ours. However, we could compare our approach to the pornography detection task by testing our best-performing baseline model on the Pornography-2k dataset. Our approach yielded an F2-Score of 88.83%, compared to our related works, Moreira et. al. with 93.53% (12) and who also aim at pornography and violence detection, and Wehrmann et. al. with 95.20% (14), aiming at only pornography.

The answers we obtained for our research questions are:

1. Question: Can this transfer learning-based, multimodal approach archieve results within 10% of the results from related work? Answer: Yes, this approach yielded an F2-Score of 88.83%, compared to our related works, Moreira et. al. with 93.53% (12) and who also aim at pornography and violence detection, and Wehrmann et. al. with 95.20% (14), aiming at only pornography.

2. Question: What is the impact of also using audio in the model's performance? Answer: The audio feature did not affect the results of our approach, which seemed to learn to rely solely on visual features.

3. Our main research question: Can this approach, with a generic feature extraction, based on transfer learning, achieve results that approach fine

tuned and hand crafted approaches. Answer: Yes, with less than 5% difference between the most related work(12).

7.1 Currently published papers

By the time of this writing, three papers about this work have already been published in conferences: (28), (29), and (30). We also have finished the construction of the dataset for sensitive content detection, to be published soon.

7.2 Future Work

Even with generic feature extraction CNNs, we achieved almost 90% on the pornography detection task. One future work is to create a late fusion model and evaluate it based on each feature type. Another possibility is to extend this approach even further, creating new CNNs from scratch to classify the videos based on, audio, visual and motion features. Both training the feature extraction methods from scratch and using a late fusion could help create a model that balances the use of each multimodal feature. Another possible future work is to add motion information, such as optical flow, to the dataset and our approach. One could also test if a sequential model can outperform a non-sequential model in specific cases that demand long-term memory, such as long videos with very small sensitive scenes. Finally, one could investigate misclassified videos in the test sets and use explainability techniques to search for insights into what circumstances our approach fails to correctly detect sensitive content.

7.3 Acknowledgments

This work was supported by the Artificial Intelligence challenge, created by Brazil's National Research Net (RNP) and Microsoft, in 2019.

Bibliography

[1] H. Yingge, I. Ali, and K.-Y. Lee, "Deep neural networks on chip-a survey," in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2020, pp. 589–592.

[2] F. Khozeimeh, D. Sharifrazi, N. H. Izadi, J. H. Joloudari, A. Shoeibi, R. Alizadehsani, J. M. Gorriz, S. Hussain, Z. A. Sani, H. Moosaei *et al.*, "Combining a convolutional neural network with autoencoders to predict the survival chance of covid-19 patients," *Scientific Reports*, vol. 11, no. 1, pp. 1–18, 2021.

[3] M. Ferguson, R. Ak, Y.-T. T. Lee, and K. H. Law, "Automatic localization of casting defects with convolutional neural networks," in *2017 IEEE international conference on big data (big data)*. IEEE, 2017, pp. 1726–1735.

[4] 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*, 2015, pp. 1–9.

[5] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 399–402.

[6] M. Terpilowski, "scikit-posthocs: Pairwise multiple comparison tests in python," *The Journal of Open Source Software*, vol. 4, no. 36, p. 1169, 2019.

[7] J. Harris, "If you don't care about parental controls, here's why you should," 2019, accessed: 2020-07-08. [Online]. Available: https://www.remosoftware.com/info/importance-of-parental-control

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[9] R. Nayak, U. C. Pati, and S. K. Das, "A comprehensive review on deep learning-based methods for video anomaly detection," *Image and Vision Computing*, vol. 106, p. 104078, 2021.

[10] V. Sharma, M. Gupta, A. Kumar, and D. Mishra, "Video processing using deep learning techniques: A systematic literature review," *IEEE Access*, 2021.

[11] H. A. Ullah, S. Letchmunan, M. S. Zia, U. M. Butt, and F. H. Hassan, "Analysis of deep neural networks for human activity recognition in videos–a systematic literature review," *IEEE Access*, 2021.

[12] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Multimodal data fusion for sensitive scene localization," *Information Fusion*, vol. 45, pp. 307–323, 2019.

[13] ——, "Pornography classification: The hidden clues in video space–time," *Forensic science international*, vol. 268, pp. 46–61, 2016.

[14] J. Wehrmann, G. S. Simões, R. C. Barros, and V. F. Cavalcante, "Adult content detection in videos with convolutional and recurrent neural networks," *Neurocomputing*, vol. 272, pp. 432–438, 2018.

[15] M. Torres Castro, "Automatic flagging of offensive video content using deep learning," Master's thesis, Universitat Politècnica de Catalunya, 2018.

[16] J. Mahadeokar and G. Pesavento, "Open sourcing a deep learning solution for detecting nsfw images," *Retrieved August*, vol. 24, p. 2018, 2016.

[17] A. P. B. Lopes, S. E. d. Avila, A. N. Peixoto, R. S. Oliveira, M. d. M. Coelho, and A. d. A. Araújo, "Nude detection in video using bag-of-visual-features," in *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*, 2009, pp. 224–231.

[18] S. Avila, N. Thome, M. Cord, E. Valle, and A. D. A. AraúJo, "Pooling in image representation: The visual codeword point of view," *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453–465, 2013.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, 1997.

[20] S. Singh, R. Kaushal, A. B. Buduru, and P. Kumaraguru, "KidsGUARD: Fine Grained Approach for Child Unsafe Video Representation and Detection," in *Proceedings of the 34th Annual ACM Symposium on Applied Computing*, 2019.

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

[22] K. Song and Y.-S. Kim, "An enhanced multimodal stacking scheme for online pornographic content detection," *Applied Sciences*, vol. 10, no. 8, p. 2943, 2020.

[23] M. Schedi, M. Sjöberg, I. Mironică, B. Ionescu, V. L. Quang, Y. Jiang, and C. Demarty, "Vsd2014: A dataset for violent scenes detection in hollywood movies and web videos," in *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2015, pp. 1–6.

[24] L. Wang, J. Zhang, Q. Tian, C. Li, and L. Zhuo, "Porn streamer recognition in live video streaming via attention-gated multimodal deep features," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[25] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[26] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 2017, pp. 1597–1600.

[27] Y. Liu, X. Gu, L. Huang, J. Ouyang, M. Liao, and L. Wu, "Analyzing periodicity and saliency for adult video detection," *Multimedia Tools and Applications*, vol. 79, no. 7, pp. 4729–4745, 2020.

[28] P. V. de Freitas, G. N. d. Santos, A. J. Busson, Á. L. Guedes, and S. Colcher, "A baseline for nsfw video detection in e-learning environments," in *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, 2019, pp. 357–360.

[29] P. V. de Freitas, A. J. Busson, Á. L. Guedes, and S. Colcher, "A deep learning approach to detect pornography videos in educational repositories," in *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. SBC, 2020, pp. 1253–1262.

[30] A. C. Serra, P. R. C. Mendes, P. V. A. de Freitas, A. J. G. Busson, A. L. V. Guedes, and S. Colcher, "Should i see or should i go: Automatic detection of sensitive media in messaging apps," in *Proceedings of the Brazilian Symposium on Multimedia and the Web*, ser. WebMedia '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 229–236. [Online]. Available: https://doi.org/10.1145/3470482.3479639

[31] F. Rosenbaltt, "The perceptron–a perciving and recognizing automation," *Cornell Aeronautical Laboratory*, 1957.

[32] K. Gurney, *An introduction to neural networks*. CRC press, 2018.

[33] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193—-202, 1980.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[35] M. Huh, P. Agrawal, and A. A. Efros, "What makes imagenet good for transfer learning?" *arXiv preprint arXiv:1608.08614*, 2016.

[36] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

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

[39] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[41] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *Technical Report*, 2017.

[42] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[43] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 270–279.

[44] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[45] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[46] S. S. Haykin *et al.*, *Neural networks and learning machines/Simon Haykin.* New York: Prentice Hall„ 2009.

[47] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[48] P. Nemenyi, *Distribution-free Multiple Comparisons*. Princeton University, 1963. [Online]. Available: https://books.google.com.br/books?id=nhDMtgAACAAJ

A
Complementary tables

Table A.1: Weighed F2-Score for each fold and each baseline model.

	MLP	LSTM	SVM	KNN
1	99,1196	98,9038	98,9917	96,2230
2	99,1017	99,0298	99,0513	96,6345
3	99,1915	99,1374	98,6160	96,4204
4	99,0477	98,9758	98,7941	96,3366
5	99,1735	98,9938	99,0710	96,7749
6	99,0118	98,8861	98,9128	96,2796
7	99,0119	99,1196	98,8926	96,5178
8	99,1374	99,1915	98,8679	96,4022
9	98,8501	98,7963	98,8498	96,4195
10	99,0398	98,9439	98,9219	96,3810
11	98,9937	98,9220	98,8284	96,9448
12	99,2273	98,9938	98,5445	96,3657
13	99,2810	99,1375	99,0836	96,9391
14	99,1103	98,9526	98,9756	96,2957
15	99,0908	99,0316	98,9038	96,4567
16	99,1700	99,1502	98,9399	96,3819
17	98,7945	98,9525	98,9398	96,3807
18	98,8929	98,8338	98,9756	96,9062
19	99,2885	99,0513	98,9756	96,7222
20	98,9517	98,7943	98,8499	95,6932

Table A.2: The amount of youtube videos collected per query.

Query	Video count
amamentacao	987
animation	823
breastfeeding	724
ufc	592
model	541
pool	526
gymnastics	474
pool party	459
ecchi	431
fisiculturismo	426
boxing	416
yoga	368
animação	348
anime	337
surf	321
MMA	314
swimming	297
beach	279
Total	8663

Table A.3: Video distribution per main tag on the Youtube macro-class.

Main tag	# Videos	Total Duration	Mean Duration	STD Duration	Total Size	Tag coverage (%)
Hard Safe Videos	8640	1278:28:34	00:08:52	00:06:46	638.5GiB	100
Arts & Entertainment	8554	598:33:35	00:04:11	00:01:36	221.5GiB	100
Games	7182	530:11:13	00:04:25	00:01:47	250.6GiB	100
Autos & Vehicles	6812	455:38:54	00:04:00	00:01:39	253.3GiB	100
(Unknown)	4054	283:05:42	00:04:11	00:01:37	109.0GiB	100
Food & Drink	3552	262:25:03	00:04:25	00:01:42	129.2GiB	100
Sports	3053	203:38:24	00:04:00	00:01:38	100.4GiB	100
Business & Industrial	2584	14:38:23	00:04:14	00:01:43	83.1GiB	100
Computers & Electronics	2325	169:45:31	00:04:22	00:01:44	84.3GiB	100
Hobbies & Leisure	2110	149:44:31	00:04:15	00:01:44	77.1GiB	100
Pets & Animals	2000	129:44:49	00:03:53	00:01:37	61.9GiB	100
Shopping	1667	22:32:11	00:04:15	00:01:41	57.8GiB	100
Home & Garden	1543	104:37:46	00:04:04	00:01:40	46.9GiB	100
Science	1233	82:13:19	00:04:00	00:01:32	32.5GiB	100
Beauty & Fitness	848	59:25:57	00:04:12	00:01:40	29.3GiB	100
Travel	688	22:03:23	00:04:00	00:01:37	21.1GiB	100
Law & Government	658	45:25:34	00:04:08	00:01:42	19.3GiB	100
Internet & Telecom	427	05:49:49	00:04:11	00:01:37	14.5GiB	100
Books & Literature	362	01:19:18	00:04:11	00:01:36	7.9GiB	100
People & Society	295	20:40:09	00:04:12	00:01:43	7.4GiB	100
Reference	290	21:15:52	00:04:23	00:01:44	8.8GiB	100
News	253	16:40:53	00:03:57	00:01:36	6.4GiB	100
Jobs & Education	235	17:01:50	00:04:20	00:01:39	6.7GiB	100
Finance	196	15:30:18	00:04:44	00:01:49	4.8GiB	100
Real Estate	70	04:52:48	00:04:10	00:01:34	1.9GiB	100
Health	20	01:29:33	00:04:28	00:01:39	563.7MiB	100

Table A.4: Video distribution per main tag on the Pornography macro-class.

Main tag	# Videos	Total Duration	Mean Duration	STD Duration	Total Size	Tag coverage (%)
gay	20604	1888:35:01	00:05:29	00:03:26	170.7GiB	100,0000
teen	18303	1922:49:21	00:06:18	00:03:20	312.9GiB	100,0000
blowjob	4197	503:44:56	00:07:12	00:03:00	88.5GiB	100,0000
other	3348	348:48:55	00:06:23	00:06:16	47.0GiB	91,6502
cumshot	2333	348:27:39	00:08:57	00:06:34	66.6GiB	100,0000
hard_porn	1947	256:37:28	00:07:54	00:04:26	116.0GiB	0,0000
anal	1821	248:31:25	00:08:11	00:06:19	83.0GiB	100,0000
lesbian	1269	150:05:51	00:07:05	00:04:02	26.4GiB	100,0000
sexy	1058	112:49:57	00:06:23	00:05:03	19.9GiB	100,0000
amateur	1021	85:53:07	00:05:02	00:04:46	25.5GiB	100,0000
milf	804	91:00:05	00:06:47	00:04:47	15.0GiB	100,0000
bdsm	800	92:24:09	00:06:55	00:04:02	33.8GiB	100,0000
shemale	753	67:58:58	00:05:25	00:04:14	8.4GiB	100,0000
exotic	672	69:31:01	00:06:12	00:05:15	11.5GiB	100,0000
big_tits	586	68:57:58	00:07:03	00:04:03	19.1GiB	100,0000
ass	571	53:05:27	00:05:34	00:04:51	23.0GiB	100,0000
sex_toys	547	71:14:52	00:07:48	00:04:34	24.6GiB	100,0000
asian_woman	469	50:16:47	00:06:25	00:05:20	16.2GiB	100,0000
lingerie	416	55:24:42	00:07:59	00:03:42	17.6GiB	100,0000
cam_porn	411	52:02:43	00:07:35	00:05:27	9.9GiB	100,0000
stockings	397	52:09:06	00:07:52	00:03:49	18.1GiB	100,0000
blonde	362	51:42:48	00:08:34	00:05:49	16.3GiB	100,0000
bukkake	279	30:40:28	00:06:35	00:06:14	6.4GiB	100,0000
interracial	274	30:32:53	00:06:41	00:04:13	3.3GiB	100,0000
big_ass	237	21:03:45	00:05:19	00:05:04	6.7GiB	100,0000
orgy	196	21:24:24	00:06:33	00:02:38	3.3GiB	100,0000
latina	170	14:26:20	00:05:05	00:04:54	3.4GiB	100,0000
pornstar	162	18:52:39	00:06:59	00:04:53	4.0GiB	100,0000
toons	162	16:46:25	00:06:12	00:05:43	4.4GiB	100,0000
brunette	154	21:10:59	00:08:15	00:05:36	4.2GiB	100,0000
solo_-_masturbation	137	12:26:16	00:05:26	00:05:09	2.1GiB	100,0000
pissing	135	15:43:48	00:06:59	00:04:42	4.1GiB	100,0000
massage	133	12:26:51	00:05:36	00:01:49	1.0GiB	100,0000
squirting	126	13:07:19	00:06:14	00:04:40	2.1GiB	100,0000
creampie	125	14:52:00	00:07:08	00:06:17	3.4GiB	100,0000
heels	114	14:07:21	00:07:25	00:05:17	2.4GiB	100,0000
virtual_reality	113	10:51:17	00:05:45	00:03:09	1.5GiB	100,0000
feet	110	10:11:12	00:05:33	00:04:07	1.6GiB	100,0000
fisting	109	12:20:43	00:06:47	00:05:16	1.4GiB	100,0000
indian	108	07:11:21	00:03:59	00:04:29	564.3MiB	100,0000
facial	108	15:44:01	00:08:44	00:06:05	2.5GiB	100,0000
mature	105	11:14:26	00:06:25	00:05:20	1.4GiB	100,0000
gapes	104	09:30:18	00:05:29	00:03:27	787.8MiB	100,0000
oiled	103	09:28:37	00:05:31	00:04:16	1.0GiB	100,0000
big_cock	102	10:51:22	00:06:23	00:04:23	1.4GiB	100,0000
sex_dolls	101	04:40:37	00:02:46	00:03:21	969.2MiB	100,0000
black_woman	100	06:29:52	00:03:53	00:04:48	1.0GiB	100,0000
redhead	100	11:25:17	00:06:51	00:05:37	2.9GiB	100,0000
bi_sexual	97	11:59:56	00:07:25	00:04:47	1.5GiB	100,0000
bbw	96	05:14:33	00:03:16	00:04:02	803.4MiB	100,0000
workout	95	08:36:48	00:05:26	00:02:58	981.9MiB	100,0000
shaved_pussy	95	09:49:53	00:06:12	00:04:44	1.1GiB	100,0000
gangbang	93	12:12:51	00:07:52	00:06:46	1.3GiB	100,0000
celebrity	93	05:02:23	00:03:15	00:04:00	661.2MiB	100,0000
real_amateur	91	09:26:47	00:06:13	00:06:53	1.3GiB	100,0000
japanese	89	13:46:47	00:09:17	00:06:21	2.0GiB	100,0000
swingers	82	05:35:26	00:04:05	00:05:38	529.5MiB	100,0000
ass_to_mouths	76	11:56:49	00:09:25	00:08:03	2.8GiB	100,0000
arab	72	04:37:33	00:03:51	00:03:28	1.2GiB	100,0000
asmr	63	08:09:12	00:07:45	00:05:57	4.2GiB	100,0000

B
Datasheet

110K Sensitive Video Dataset (110k-SVD) Datasheet

I. Motivation For Dataset Creation

A. Why was the dataset created? (e.g., was there a specific task in mind? was there a specific gap that needed to be filled?)

We define sensitive video as for any video that contains pornography or extremely violent scenes (that usually include but are not limited to the appearance of blood). At the time of creation, we could not find any open access datasets that had this or a similar definition and had more than 10.000 videos. This dataset was designed to be used specifically in the binary classification of entire videos, to determine whether a video contains sensitive content or not.

B. Has the dataset been used already? If so, where are the results so others can compare (e.g., links to published papers)?

No, the results of the baselines of this task are to be ublished.

C. What (other) tasks could the dataset be used for?

Tasks that include binary or multi-label classification of ıe macro and micro classes of this dataset. Such as:

- Multi-label classification (or tagging) of pornographic videos;
- Multi-label classification of ”Safe” (Videos that do not contain sensitive content);
- Binary classification of extremely violent videos (hereby referred to as gore);
- Binary classification of pornography.

D. Who funded the creation dataset?

The creation of the 110K Sensitive Video Dataset database was supported by a joint challenge by Microsoft and Brazil's National Research Net (RNP) in 2019.

II. Dataset Composition

A. What are the instances? (that is, examples; e.g., documents, images, people, countries) Are there multiple types of instances? (e.g., movies, users, ratings; people, interactions between them; nodes, edges)

Each instance is a video (min 5 seconds, max 31 minutes).

TABLE I
General statistics of the two main classes of the dataset

	Sensitive	Safe
Video Count	67424	59651
Total Duration	6953:27:41	4852:53:31
Mean Duration	00:06:11	00:04:52
STD Duration	00:04:12	00:03:26
Max Duration	00:30:55	00:30:55
Min Duration	00:00:05	00:00:05
Total Size	1.2TiB	2.2TiB
Mean Size	19.3MiB	39.0MiB
STD Size	35.4MiB	42.3MiB
Features Size	519.4GiB	376.8GiB
Tag coverage	65392	51011
Tag coverage (%)	96,9862	85,5157

B. How many instances are there in total (of each type, if appropriate)?

As shown in Table II, it is divided into 59,651 safe videos and 67,424 videos with sensitive content. Those sensitive videos are 54,549 Pornographic Videos and 2,356 Gore Videos. Tag coverage refers to main tag annotation existence (videos also may have subtags but no main tag).

C. What data does each instance consist of ? “Raw” data (e.g., unprocessed text or images)? Features/attributes? Is there a label/target associated with instances? If the instances related to people are subpopulations identified (e.g., by age, gender, etc.), what is their distribution?

Each video will be distributed as extracted and processed visual and audio features. Each video file is associated with an id, a label, and a sequence size. There are people in the videos, but subpopulations are not identified.

D. Is there a label or target associated with each instance? If so, please provide a description.

Each video file is associated with a label (proper/improper) and id. Some examples of video data associated with the features: improper_29024487, proper_MqnZqzAxQTk, improper_gore122. There is also a main dataframe, this dataframe is indexed by video id and contains all the other gathered data, such as tags, subtags, file size, duration in seconds, and title.

E. Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

For the main labels (sensitive/improper and safe/proper) there are no missing labels. For tags and subtags, there is some missing information because either the website did not a tag system or the video had no tags on the website. The coverage of tags is shown in Table II.

F. Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

An individual might appear in multiple videos. This relationship was not collected and registered in the dataset. Other than this, there are no known relationships between instances.

G. Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

This dataset is a sample, not necessarily random, of instances from a larger set. The larger set is all the videos from each crawled website. This dataset is representative of the sites we crawled because of the equally sampled variety of video types inside those sites and because of its large mount of instances.

However, the dataset is not a fully representative set of the ntirety of videos on the internet, to be more representative f our definition of sensitive videos, the data should have to e collected from more video hosting sites. The sites used or gathering the videos were the most easily obtainable data t the time.

I. Are there recommended data splits or evaluation measures? (e.g.,training, development, testing; accuracy/AUC)

The instances come bundled in .npz files, each file represents a batch. We split our dataset into training/validation nd testing batches. We publish features for all batches, ut only publish labels for the train/validate batches. The ser is free to split the training/validation as desired. For aining evaluation, we recommend 20-fold cross-validation to perform training and validation.

To report performance in the binary classification of sensitive videos, we recommend Precision (P), Recall (R), and, most importantly, the weighted F2 score. In this section, we present a contextualized explanation of these metrics.

In the context of sensitive content detection, *true positives* are videos predicted as sensitive and are in fact, sensitive. Likewise, *true negatives* are videos predicted as safe and are indeed safe. *False positives* are videos predicted as sensitive, but were safe, the same goes for *false negatives*, which are videos that were predicted as safe, but were predicted as sensitive.

Precision (Equation 1) measures how many videos predicted as sensitive (both true positives and false positives) are truly sensitive. The Recall (Equation 2) measures how many truly positive videos were correctly identified.

$$P = \frac{TP}{TP + FP} \quad (1) \qquad R = \frac{TP}{TP + FN} \quad (2)$$

Where TP, TN, FP, and FN denote the examples that are true positives, true negatives, false positives, and false negatives, respectively.

$$F_\beta = (1 + \beta^2) \times \frac{P \times R}{(\beta^2 \times P) + R} \quad (3)$$

The F_β-score, defined in Equation 3, evaluates the classifier by the harmonic mean between Precision and Recall. To account for label imbalance, after calculating the F2-score metrics for each label, we find their average weighted by support (the number of true instances for each label).

While the F1-score represents a balanced performance metric, the F2-score gives twice more weight to the recall than to precision, which means that the metric is more focused on the recall of a solution.

We chose the weighted F2 score as our main evaluation metric because when detecting sensitive content it is more important to predict a truly sensitive video than to predict a safe video as sensitive.

I. Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

There might be annotation divergences in the tags and subtags of the pornography videos since the videos were tagged by users and not by a centralized annotation group. We can not guarantee that frames and/or audio clips do not appear in other videos since there was no direct contact with the videos during dataset creation. There was however a duplicate removal step in the creation of the dataset, detailed later in this document.

J. Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

Everything needed to perform the proposed tasks is included.

Any other comments?

III. Collection Process

A. What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

There was no direct human curation, the videos were automatically collected based on their titles and tags. We created crawlers to automatically collect the videos.

B. How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

If available, the title, tags, and subtags of the video were collected and stored by the crawler. The video feature extraction process was already validated for multi-label video classification [1]. To validate the feature extraction process for the task we propose we also trained and tested baseline models and archived an F2 score of 99% in our test subset nd 88.83% in a popular pornography dataset [2].

C. If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

The sampling strategy was to collect at least the amount f the less numerous tag for each tag, then, to complete the umber of collected videos the sampling probabilities were roportional to the database distribution of each tag.

D. Who was involved in the data collection process (e.g., tudents, crowd workers, contractors) and how were they ompensated (e.g., how much were crowd workers paid)?

Two graduate students.

E. Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.

The dataset was crawled from January 2019 to December 2019, this timeframe does not match the creation of the data associated with the instances.

IV. Data Preprocessing

A. Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

Once the raw data was crawled, we performed feature extraction in all videos successfully collected. To generate image frame features and audio features we decode each video at approximately 1 frame-per-second and feed an InceptionV3 network [3] pre-trained on the ImageNet[1] dataset.

We also make use of an AudioVGG [4] network with pre-trained weights in the Audioset[2] dataset to extract the audio embeddings. Each of these CNNs was used as published by their authors; the only modification was the removal of classification layers in both CNNs to obtain their respective embeddings.

Next, we apply Principal Component Analysis (PCA) [5] to each of the outputs to reduce the dimensions of both embeddings and to generate feature vectors of size 1024 and 128 for frame and audio embeddings respectively.

We concatenate both image and audio embeddings extracted in the current frame and audio window in order to compose the final embeddings as a sequence of the same size as the number of seconds of the video. After this concatenation, each time-step has 1,152 features: 128 audio features and 1024 frame features.

Notice that with this approach, the video is transformed into a time series, and to use it in non-sequential models (*e.g.* SVM, KNN, and MLP) we need to turn this sequence into a single feature vector that represents the whole video. In our setting, we did that by taking the average, median, standard deviation, min, and max values for each feature to represent the entire video. In summary, we turn the sequence of features with size n and shape n by 1,152 into a single feature with shape 1 by 5,760.

We also filtered out short and long videos. For the short videos, we defined that the minimum length of a video was 5 seconds based on [2], which was 0.09% of the dataset. To define the maximum length of a video in the dataset, first, we removed all videos with less than 5 seconds, then we calculated the mean and standard deviation of each video's duration. The maximum length of a video in the dataset is $mean + 2 * std$, which resulted in approximately 31 minutes and covered 98,94% of the videos.

Not all video features were successfully extracted for multiple reasons, such as corrupt data, unknown format, and missing audio. For those videos with missing audio or image, the features were still generated, but their respective modal feature were zeros. Those videos which do not have any features successfully extracted were removed from the dataset.

We also removed any duplicated videos that were detected, for duplicate video detection we used, we matched either id,

[1] http://www.image-net.org/
[2] https://research.google.com/audioset/

title, or checksum.

We recommend equally balancing both main labels (sensitive/improper and safe/proper) so that both main classes have the same number of instances. One could also choose not to balance both classes equally, since our main metric already takes label imbalance into account. Additionally, when removing excess sensitive content (while balancing), we recommend removing only pornography videos in order to not lower the number of gore videos.

B. Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

We can not provide a link for the raw video data, but we are open to including other feature extraction methods. If there are any suggestions for better or newer feature extraction methods, please, get in contact with us.

C. Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.

Yes, the feature extraction method and our preprocessing are available in the GitHub repository: `https://github.com/TeleMidia/Sensitive-Video-Dataset`.

D. Does this dataset collection/processing procedure achieve the motivation for creating the dataset stated in the first section of this datasheet? If not, what are the limitations?

Although our baseline tests perform well on the gore detection task, there is still a relatively small amount of gore videos in our dataset. Furthermore, the is no manually curated dataset comparative to the gore videos. Mainly because of the difference in our definition of violence, which is just highly violent scenes such as death, mutilation, and torture.

E. Any other comments

V. Dataset Distribution

A. How will the dataset be distributed? (e.g., tarball on website, API, GitHub; does the data have a DOI, and is it archived redundantly?)

The dataset scripts, updates, papers, and additional information will be hosted on GitHub: `https://github.com/TeleMidia/Sensitive-Video-Dataset`. The dataset itself will be hosted by the IEEE Dataport:

- DOI: 10.21227/sx01-1p81
- URL:`https://ieee-dataport.org/documents/sensitive-video-dataset`

The dataset will be distributed in multiple .npz files, organized in multiple directories:

- *train_val_batches*
- *test_subset_batches*
- *non_sequential_train_val_batches*
- *non_sequential_test_subset_batches*

There is also a main dataframe, this dataframe is indexed by video id and contains all the other gathered data, such as tags, subtags, file size, duration in seconds, and title.

Each npz file represents a batch of variable size, but all split to have at max 4 Gbs when loaded to memory. Each npz file has keys and values, the keys are string in the format ¡label¿_¡video id¿. Some examples of keys in the npz file: "improper_29024487", "proper_MqnZqzAxQTk", "improper_gore122".

The values are the videos features stored in NumPy arrays, of varying shapes, depending on the dataset variation (sequential or non-sequential).

The dataset has two variations:

- Sequential: Each sample remains as it was extracted, a single video generates a sequence of N samples. In this variation, inside each npz file, each instance is represented by an N by 1152 NumPy array.
- Non-Sequential: All samples of a video are aggregated into a single sample, resulting in each instance having a shape of 1 by 5760, this single sample summarizes the entire video.

The data is archived redundantly.

B. When will the dataset be released/first distributed? What license (if any) is it distributed under?

It is available on IEEE Dataport (`https://ieee-dataport.org/documents/sensitive-video-dataset`) under Ceative Commons Attribution 4.0 International (CC BY 4.0).

C. Are there any copyrights on the data?

No.

D. Are there any fees or access/export restrictions?

There are no fees or restrictions.

E. Any other comments?

VI. Dataset Maintenance

A. Who is supporting/hosting/maintaining the dataset?

The dataset is hosted by IEEE Dataport and maintained by the authors.

B. Will the dataset be updated? If so, how often and by whom?

There are no expected updates on this dataset.

C. How will updates be communicated? (e.g., mailing list, GitHub)

If any, updates will be communicated via the dataset's GitHub page/repository.

D. If the dataset becomes obsolete how will this be communicated?

Through the dataset's GitHub page/repository.

E. Is there a repository to link to any/all papers/systems that use this dataset?

The links about papers and works using our dataset will be held on the dataset's GitHub repository:`https://github.com/TeleMidia/Sensitive-Video-Dataset`.

F. If others want to extend/augment/build on this dataset, is there a mechanism for them to do so? If so, is there a process for tracking/assessing the quality of those contributions? What is the process for communicating/distributing these contributions to users?

Others are free to use and modify our datasets. Contributions can be discussed via email (pedropva@telemidia.puc-rio.br).

VII. Legal and Ethical Considerations

A. Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

No. The data was crawled from public websites. The reproducible videos were not assessed by anyone and will not be distributed, only the features will be distributed. Those features can not be reverted or recreated into reproducible videos.

B. Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.

No, all videos were crawled from public sources, furthermore, only video the features will be distributed. Those features can not be reverted or recreated into reproducible videos.

C. Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why

No, only video features will be distributed. Those features can not be reverted or recreated into reproducible videos.

D. Does the dataset relate to people? If not, you may skip the remaining questions in this section.

No, only video features will be distributed. Those features can not be reverted or recreated into reproducible videos.

E. Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

Not applicable.

F. Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.

Not applicable.

G. Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.

Not applicable.

H. Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

Not applicable.

I. Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

Not applicable.

J. Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access points to, or otherwise reproduce, the exact language to which the individuals consented.

Not applicable.

K. If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

Not applicable.

L. Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis)been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

Not applicable.

M. Any other comments?

REFERENCES

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Pornography classification: The hidden clues in video space–time. *Forensic science international*, 268:46–61, 2016.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[5] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.