PEDA 376K: A Novel Dataset for Deep-Learning Based Porn-Detectors

Danilo Coura Moreira
Federal University of C. Grande (UFCG)
Campina Grande, PB, Brazil
Email: danilocoura@copin.ufcg.edu.br

Eanes Torres Pereira
Federal University of C. Grande (UFCG)
Campina Grande, PB, Brazil
Email: eanes@computacao.ufcg.edu.br

Marco Alvarez
University of Rhode Island (URI)
Kingston, RI, USA
Email: malvarez@cs.uri.edu

***Abstract*—The rapid expansion of the digital world introduces complex challenges within the forensic and security domains. In particular, the wide availability of online pornographic media is a huge problem for applications that seek to prevent exposure to inappropriate/undesired audiences, or that aim to automate the detection of any illegal behavior. There is a thin veil separating the definition of pornographic and non-pornographic media, making it difficult, even for humans, to agree on a consistent interpretation. Most of the available APIs for detecting NSFW (not-safe-for-work) media are not able to infer clearly whether a file contains pornographic content or not. In general, given an input file, APIs return a set of probability scores, leaving the responsibility of a final binary decision to the users. What is more, NSFW APIs do not publicly share their training datasets.**

Aiming to mitigate these issues, we introduce a novel dataset of images: *Pornographic and Explicit Dataset 376K* (PEDA 376K), which was labeled using a well-defined criteria, to aid the development of machine learning for detecting whether an image is pornographic or not. We also trained decision trees for transforming the probabilistic output of standard APIs into binary decisions. We conducted experiments with two datasets, PEDA 376K and RedLight, and found that when APIs are optimized with decision trees, their average accuracies increase. Finally, we propose a *deep learning* architecture trained directly on the PEDA 376K dataset. When comparing this model against state-of-the-art models and their corresponding optimized outputs, we outperform five existing neural architectures, reaching an overall accuracy of 99.2%.

***Index Terms*—Pornography Detection, Convolutional Neural Networks, Computer Vision, Computer Forensics.**

I. Introduction

As the amount of data being produced and shared online is increasing at enormous rates, traditional techniques for detecting pornographic content, such as keeping black lists of file names or URLs [1], no longer scale. Computer vision and deep learning technologies have become crucial for this task, shifting the focus from metadata to the actual media contents [2]. With recent advances, porn detectors can now be embedded into applications to automatically filter sensitive media, preventing exposure to inappropriate/undesired audiences and environments [3], [4].

Although recent deep learning technologies are very powerful in computer vision applications, researchers and engineers have to deal with the presence of subjectivity in their models. For example, a single media file can be deemed as pornography or not by two different individuals [5]. This lack of natural consensus in categorizing media can lead to a wide variety of trained models. In practice, most of the state-of-the-art NSFW APIs do not provide a binary decision as to whether an image is pornographic or not. Rather, APIs return a set of probabilities for certain features characterizing the media file, leaving the final decision to the user.

In this paper, we introduce a pornographic dataset (PEDA 376K) carefully labeled using a set of well-defined rules to determine whether an image is pornographic or not. We performed extensive experiments involving the training of convolutional networks to detect pornography, and compared their results with five start-of-the-art NSFW APIs. Overall, the contributions of this paper include:

- A novel image dataset (PEDA 376K), containing more than $376,000$ images labeled into two categories: (i) pornographic and (ii) non-pornographic, using an objective definition of pornography, mitigating the subjective nature of each class;
- A neural network architecture trained from scratch for detecting pornography using the novel PEDA 376K dataset;
- An approach for transforming probabilistic outputs from state-of-the-art NSFW moderation APIs into binary decisions.

The rest of this paper is distributed as follows: Section II exposes related work about the evolution of pornography detection. Section III describes the development of our dataset, our proposed neural network architecture, and details about the interpretation of the probabilistic outputs from NSFW APIs. Section IV shows details about our experiments and results. Section V highlights our major findings and final remarks.

II. Related Work

Originally, porn detectors were based on the analysis of skin features. These approaches focused on several techniques to establish whether pixels in a given image belong to skin or not [6], [7]. The least sophisticated porn detectors estimated the proportion of skin pixels in the whole image and used a threshold to classify the image. Subsequently, more complex features were added, such as, the number of skin regions or the area of the largest skin region. Thresholds for classification were inferred empirically by experimentation and observations [8]. In addition to skin features, other works leaned on texture analysis and human geometry to mitigate

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

false positives, as skin pixel values are sometimes similar to ordinary pixels [9].

Over time, these "classifiers" became deprecated and were replaced by machine learning models that could find better thresholds compared to previous methods [10], [11]. For example, models based on the bag-of-visual-words approach achieved better results compared to naive models [2]. Although these machine learning approaches were more powerful, just looking for skin features may not be enough, as some pornographic images may not always contain a large amount of skin pixels, or some non-pornographic images may contain large amounts of skin pixels [12].

More recently, the advent of deep learning allowed scientists and engineers to improve machine learning models. In [14], the authors compared different machine learning models using the TI-UNRAM Pornographic Image Dataset [13]. Such models included convolutional neural networks (CNNs), bag-of-visual-words methods, and traditional shallow machine learning algorithms combined with manually extracted features (i.e., local binary patterns, histogram of oriented gradients, SIFT). A CNN, the ResNet architecture, achieved the best accuracy compared to all other approaches.

Models based on a combination of skin detection and bag-of-visual-words were compared to CNNs to detect pornography in [2]. Using a private dataset of approximately $650,000$ images, the authors observed a superior accuracy of the CNN, the AlexNet architecture [15], over the other techniques. Another example of superior performance of CNNs is presented in [16], where a bag-of-visual-words technique was compared to a CNN. This time, the authors used a GoogLeNet architecture [17], using a private and the public NPDI Pornography [18] datasets for training and testing, respectively. The GoogLeNet architecture was also used to detect pornography in [16]. The authors introduced a novel approach that uses Multiple Instance Learning (MIL). In this approach, the model is trained to label arbitrary parts of an image, and if one of these is considered pornography, the entire image will also be. They used a private balanced dataset containing about $234,000$ images.

Neural networks were also used to leverage ordinary pornography detection to develop forensics applications. One example is the detection of child sexual exploitation. Porn detectors can help to find evidence on child pornography investigations. To this end, in [12], five architectures trained to detect pornography were evaluated in a new role, the detection of child pornography. The highest accuracy was achieved by a publicly available model, OpenYahoo [19], a 50-layer ResNet. On a similar work, in [20] the authors explored different transfer learning strategies for CNNs. Transfer learning allows a particular model to be specialized, with little effort, on a different domain or application.

III. Methodology

In this section, we introduce a novel dataset for pornography detection and present details about the materials and methods used in our experiments.

A. Pornographic and Explicit Dataset 376K

As a rule-of-thumb, when deep neural networks exhibit a large number of parameters, a huge amount of data is necessary for achieving satisfactory results. In practice, finding huge amounts of annotated data may be difficult for several application domains [1]. In the context of pornographic images, this issue is magnified as there is no reliable and structured dataset that is publicly available. Given the sensitive nature of the materials, most of successful projects do not provide their datasets [21], [1], [16], [22], and when they do, either the amount of data may be insufficient or the labeling appears subjective [12].

In this paper, we introduce a dataset[1] that aims to address the aforementioned limitations. Naturally, collecting this type of data is daunting, as it involves downloading representative examples according to an objective definition of pornography, followed by a tedious manual inspection before labeling the data. We decided to gather data from a social network, where contents are already organized by topics [23], providing an starting point for our manual inspection.

Images were scrapped from *Reddit* posts[2] that contained images under specific topics, denominated *subreddits*. Our goal was to collect representative samples of current pornography maximizing variety. To this end, we scraped images with different photo angles, body shapes, distinct environments, ethnicities, gender, age, sexual modalities, number of people, image quality, among others. For the non-pornographic images, we collected images related to animals, arts, foods, sports, places, memes, nature, objects, vehicles and people in ordinary situations. It is worth noting that these ordinary situations contemplate images that are hard to distinguish from pornography. These examples include people wearing swimsuits, or doing some kind of sports which involves less clothes and/or physical contact between players.

Although images were scraped from explicit pornographic subtopics, some images are not necessarily pornographic. An objective rule to define pornography, based on [16] was defined. Images were treated as pornographic if any of the following apply: (a) depicts a sexual act (regardless of clothing); and (b) contains individuals showing a sexual organ, buttocks or female breast. A graphical representation of our decision process can be seen in Figure 1.

We also provide a split of the data for further experimentation and reproducibility. The data is separated into training, validation, test sets, as shown in Table I. Validation and test sets are split in a way that each set includes 50% instances of each class. Data instances were shuffled before splitting.

TABLE I: Data distribution for PEDA 376K splits.

Class	All	Train (95%)	Val. (2.5%)	Test (2.5%)
Pornography	$150,940$	$141,540$	$4,700$	4.700
Non-pornography	$225,094$	$215,694$	$4,700$	$4,700$
Total	$376,034$	$357,234$	$9,400$	$9,400$

[1]Database available at https://sites.google.com/site/peda376k.

[2]http://www.reddit.com

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

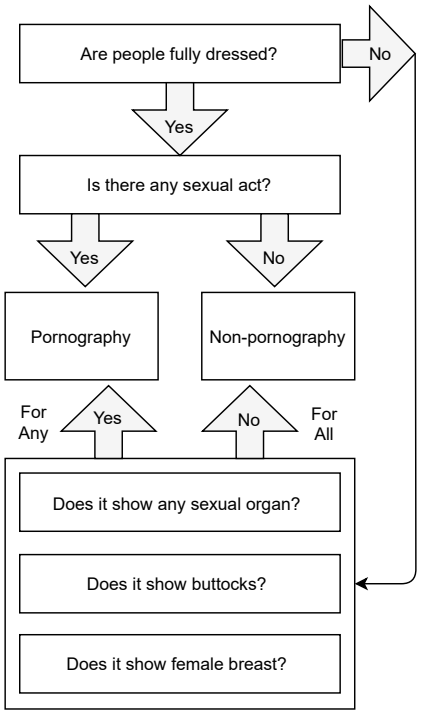

Fig. 1: Flowchart of the decision-making process for labeling an image as pornographic or not.

B. NSFW Image Moderation APIs

With the spread of online media, mainly through social networks, there has also been an increase on the number of companies providing content filtering services. Most of these services are cloud-based and aim to automatically identify if a given image is appropriate or not, according to their own criteria. These filtering services do not usually indicate whether an image is pornographic or not, and most of the times, return a set of probabilities related to specific features characterizing the image (e.g., adult, explicit, nudity, partial nudity, racy, suggestive, swimwear, etc).

In this context, when using these APIs, the decision for labeling an image as pornographic or not is given to the user. Furthermore, as the outputs of different services lack standardization, comparing the performance of these models is a difficult task. Our work aims to address the subjectivity inherent to these APIs by incorporating a decision tree on top of each API, transforming their outputs into binary decisions. We conducted experiments using two different datasets, PEDA 376K and RedLight [26] to compare all services. We considered five different APIs in our experiments.

1) Amazon Rekognition [27]: Every prediction of this service returns twenty-two probability values for an input image. There is a total of eighteen labels divided into four categories. For each category, the prediction includes one probability for each label, and the maximum probability value within the category. We discard the two unrelated categories: "Violence" and "Visually Disturbing" and maintain the two remaining: "Explicit Nudity" and "Suggestive". Thus, we only use ten labels, six from the first category (Nudity, Graphic Male Nudity, Graphic Female Nudity, Sexual Activity, Illustrated Nudity or Sexual Activity and Adult Toys) and four from the second category (Female Swimwear or Underwear, Male Swimwear or Underwear, Partial Nudity, Revealing Clothes), for a total of twelve probabilities.

2) Clarifai [28]: This service provides two content filtering modules: "NSFW" and "Moderation". The first module returns the probability of an input image having pornographic content. The second module returns five likelihoods related to moderation filtering: "Safe", "Explicit", "Suggestive", "Gore" and "Drug", but we discarded the last two. In total, we consider only four probability values.

3) Google Vision [29]: Using a different approach, the service provided by Google returns categorical values instead of probabilities. The possible values are: Very Unlikely, Unlikely, Possible, Likely, and Very Likely, which are encoded as a single numerical value: 1, 2, 3, 4 and 5, respectively. This service provides five categories, from which we only used two: "Adult" and "Racy". The other three (Spoof, Medical and Violence) were not related to pornography.

4) Microsoft Azure [30]: This services focuses on handling pornography. For an input image, the service returns two probability values: "Adult" and "Racy". It is the only service that also returns a boolean value for each label indicating if the image has pornographic or racy content.

5) OpenYahoo [19]: OpenYahoo is a publicly available neural network that has been trained to detect NSFW images. The model returns a continuous probability value that indicates whether an input image is pornographic or not.

C. CNN Proposed Model

Inspired by the massive transition of pornography detection models to deep learning approaches [31], [21], [16], [2], [32], [14], we decided to use convolutional networks with PEDA 376K. We also aimed to provide a baseline for future experiments using the same data.

As training training deep neural networks and making decisions about their hyperparameters can be very time consuming, we defined an strategy that avoids the exploration of the entire search space of hyperparameters. Our methodology is fully depicted in Figure 2 and is composed of three major steps, that aim to greedily select the CNN architecture, the batch size, and the optimizer, in that order.

1) Selecting the architecture: Based on results from ILSVRC [33], we selected four architectures, each with a variable number of layers (in parenthesis). All models are size compatible with our image dataset (224x224 pixels).

- ResNet (18, 34, 50, 101, 152) [34]: This architecture was developed in order to avoid the gradient vanishing in deeper layers. It includes a residual block composed of two convolutional layers connected also through a skip connection. This CNN was developed in 2015 and won the ILSVRC challenge in the same year;
- Wide ResNet (50, 101) [35]: This architecture is a variation of the ResNet and was developed in 2017. The

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

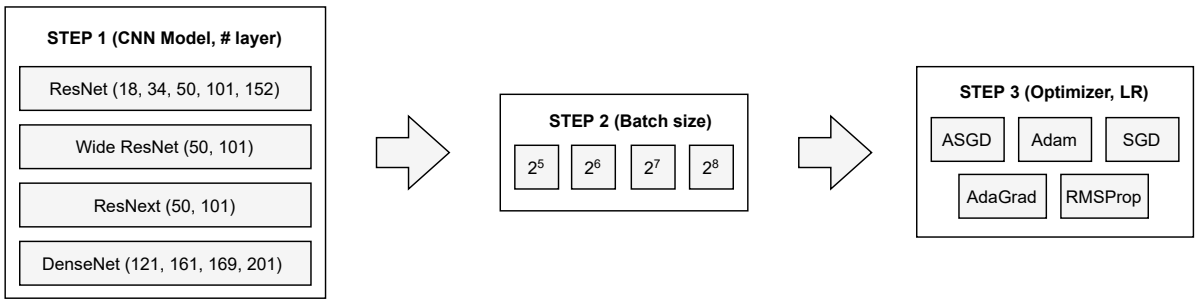

Fig. 2: Steps performed for exploring the search space of hyperparameters. The learning rate (LR) is given by $\frac{2^n}{m}$, where n and m values are different for each optimizer.

original CNN was improved by doubling the number of channels in every block, minimizing the bottleneck caused by the lack of channels;

- ResNext (50, 101) [36]: This is an extension of the ResNet architecture, which uses an split-transform-aggregate strategy. Compared to a ResNet block, this novel approach splits the convolutional layer path in a determined cardinality C, generating new paths with the same structure which are added together in the end. The authors secured 2nd place in the ILSVRC 2016 classification task;
- DenseNet (121, 161, 169, 201) [22]: This architecture proposes the use of a set of layers denominated Dense blocks which concatenate each output layer with the next output layer. This proposition results in $\frac{L(L+1)}{2}$ connections among the layers, instead of L connections as we see in traditional deep networks with L layers. It was developed in 2016 and received the Best Paper Award at CVPR 2017.

2) Selecting the batch size: This hyperparameter sets the number of samples to be used in forward and backward passes. The batch size is usually set to powers of two (2^n) due to underlying architectural attributes. Based on this premise, we explored a range of values to the batch size, where $n = \{5, 6, 7, 8\}$:

3) Selecting the optimizer and learning rate: We adopted five different optimization methods to evaluate our models: three adaptive methods, AdaGrad, RMSProp and Adam; and two non-adaptive, Stochastic Gradient Descent (SGD) and Asynchronous Stochastic Gradient Descent (ASGD). The adaptive methods converge faster than the non-adaptive, and are becoming very popular for training deep neural networks. Speed-ups are due to the use of iteration history in the local optimization [37].

In order to fine-tune the learning rate, we used the function $\frac{2^n}{m}$ based on [37]. This approach uses a different range in a power of two divided by a specific number for each optimizer. The different values for m and n for each case are shown in the Table II.

TABLE II: Learning rate exploration. For each optimizer a different set of learning rate values was explored, according to the table below. Each value is defined by $\frac{2^n}{m}$.

Optimizer	m	n
AdaGrad	10	$\{-2, -1, 0, 1, 2\}$
RMSProp	100	$\{-5, -4, -3, -2, -1, 0, 1, 2\}$
Adam	100	$\{-6, -5, -4, -3, -2, -1, 0, 1\}$
SGD	1	$\{-2, -1, 0, 1, 2\}$
ASGD	1	$\{-2, -1, 0, 1, 2\}$

IV. Experiments and Results

In this section, a detailed description of experiments, methods, and datasets is provided. Additionally, a summary of the results and major findings is presented at the end.

A. Training a CNN

In order to find the "best" CNN, the first round of experiments was conducted to decide the underlying architecture, the batch size, and the optimizer/learning rate. All models were trained with the PEDA 376K dataset and all models were initialized with pretrained weights using ImageNet [38].

1) CNN architecture: For the purpose of finding the best CNN architecture, models were training during 30 epochs with early stopping. Training was interrupted if validation accuracy showed no increase for 5 consecutive epochs. As a default configuration, the AdaGrad optimizer was used with a learning rate of 0.01 and a batch size of 128. Table III shows resulting training and validation performance.

TABLE III: Training and validation accuracies of each CNN model with their respective number of layers.

Model	Layers	Train	Validation
ResNet	18	99.5%	97.6%
	34	99.4%	97.7%
	50	98.8%	97.4%
	101	98.7%	97.3%
	152	99.2%	97.7%
Wide ResNet	50	97.6%	97.2%
	101	99.3%	97.4%
ResNext	50	99.2%	98.1%
	101	99.4%	97.8%
DenseNet	121	99.4%	**98.5%**
	161	99.5%	98.3%
	169	99.6%	**98.5%**
	201	98.6%	98.2%

For subsequent steps, the Densenets-121 was chosen as it achieved the highest accuracy in the validation set: 98.5%. Ties were broken by model complexity.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

2) Batch size: In this step, the same protocol for early stopping was used. The goal now is to find a good batch-size. Table IV presents the corresponding results.

TABLE IV: Training and validation accuracies when batch size is adjusted.

Batch size	Train	Validation
32	99.2%	98.1%
64	99.4%	98.4%
128	99.4%	**98.5%**
256	99.6%	98.4%

The model trained with a batch size of 128 showed the highest accuracy: 98.5%. For the last step, the Densenet-121 with a batch size of 128 was selected.

3) Optimizer and learning rate: In this step, five optimizers and their respective learning rates are evaluated. The learning rate ranges are detailed in Table II. Additionally, if the best validation accuracy was at one extreme of the ranges, the search was expanded to consider values beyond the initial limits. Table V shows the final training and validation performance.

TABLE V: Training and validation accuracies when varying the optimizer and learning rate. The best learning rate for each optimizer is shown together with their corresponding performances.

Optimizer	Train	Validation	Learning Rate
AdaGrad	100%	98.6%	$\frac{2^{-4}}{10}$
RMSProp	99.8%	99.1%	$\frac{2^{-7}}{100}$
Adam	99.8%	99.1%	$\frac{2^{-7}}{100}$
SGD	99.9%	**99.2%**	2^{-8}
ASGD	99.9%	99.0%	2^{-5}

The best configuration of hyperparameters found includes a Densenet-121 with batch size of 128 trained with an SGD optimizer and a learning rate of 2^{-8}. This configuration achieved 99.2% in the validation set. We also evaluated a model trained with this configuration with a test set, reaching an accuracy of 99.2%. Instances in the test set were never used in any of the training/validation steps.

B. Combining Decision Trees and NSFW APIs

In order to transform the probabilistic output of image moderation APIs into binary decisions, a number of decision trees were trained. Experiments were performed using two different datasets, as shown in Figure 3.

1) PEDA 376K: As NSFW APIs do not provide binary outputs, we first defined a baseline (raw) model. In this model, the PEDA 376K validation set is used to get the APIs' outputs ($h(x)$), normalizing the values to the $[0, 1]$ interval when necessary. In this raw model no decision tree is used. The inference of a final binary decision is rather a simple thresholding function. For each API, the most representative feature is selected. If the value for this feature is greater than 0.5 then the whole image is deemed as pornographic, and not pornographic otherwise. This rule was not applied to Microsoft Azure, as this is the only service providing a binary output.

The proposed approach for transforming probabilistic outputs into binary decisions involves training decision trees to adjust API scores to our objective definition of pornography. In this approach, $M + \lceil \frac{M-1}{M} \rceil$ decision trees are trained for each API, where M is the length of each API's output. A binary decision tree is trained for each feature separately and a final tree is used to combine all other trees. Experiments were conducted with 5-fold cross-validation and, to prevent overfitting, the minimum number of samples required to split an internal node was varied using an exponential range. This range was given by 2^n, where $n = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. For each API and their respective decision trees, the test set was use to assess performance.

It is worth noting that experiments were repeated twice by switching the validation and test sets in order to achieve more reliable results. The average accuracy and respective standard deviations are reported in Table VI.

TABLE VI: Accuracies and standard deviations for the use of decision trees (raw and optimized) and the CNN for each dataset.

Dataset	Model	Raw	Optimized
PEDA 376K	Proposed	**99.1 ± 0.1**	-
	Amazon [27]	96.9 ± 0.0	97.9 ± 0.0
	Clarifai [28]	96.4 ± 0.0	97.4 ± 0.0
	Google [29]	96.5 ± 0.1	96.7 ± 0.2
	Microsoft [30]	93.6 ± 0.1	95.8 ± 0.1
	Yahoo [19]	95.1 ± 0.1	96.1 ± 0.0
RedLight [26]	Proposed	95.2 ± 0.3	95.6 ± 0.3
	Amazon [27]	**95.3 ± 0.3**	**96.5 ± 0.3**
	Clarifai [28]	92.6 ± 0.3	93.0 ± 0.1
	Google [29]	94.7 ± 0.2	95.1 ± 0.2
	Microsoft [30]	94.4 ± 0.3	95.5 ± 0.2
	Yahoo [19]	94.2 ± 0.3	94.2 ± 0.3

2) RedLight [26]: To exclude any possibility of biased results introduced by the use of our dataset, we also experimented with an external dataset. The RedLight dataset [26] contains a set of non-pornographic and pornographic images which are further divided into several subcategories. We disregarded the subtype information for all images and only used their pornography label. Unreadable and duplicate images were removed, leaving a filtered set of $25,616$ images ($10,223$ pornographic and $15,393$ non-pornographic). The entire dataset was split into six partitions in order to perform the same amount of experiments with a distinct combination of train and test sets. Average accuracy and standard deviations were calculated for all experiments. Similarly to PEDA 376K, for RedLight, we collected results for both the baseline and the optimized models trained with decision trees. In this case, we also trained decision trees on top of our proposed CNN architecture. This step was not necessary for the analysis of PEDA 376K because the CNN was already trained with the same data.

3) Analysis and discussion: For both datasets, the highest accuracies were given by decision trees trained with all available features. As shown in Table VI, our proposed approach increased the accuracy of all NSFW APIs, when compared with their raw baselines. When considering only PEDA 376K, the CNN architecture outperformed all APIs, including their optimized versions. The second row in Table VI shows the performance of the APIs and the CNN when using RedLight. The CNN beats four of all five APIs in both the raw baseline

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

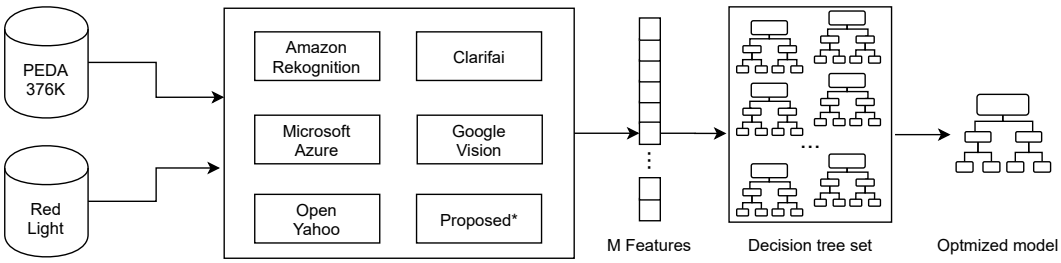

Fig. 3: The diagram illustrates the use of decision trees to transform API's outputs into binary decisions. (*) The proposed approach is only optimized over the RedLight Dataset [26]

and their respective optimized versions. The Amazon API achieved the highest accuracy in this evaluation.

In order to perform a fair evaluation of the performance of optimized APIs versus the CNN using both datasets, we calculated a weighted accuracy based on the number of predictions given for each dataset. We considered $18,800$ images from PEDA 376K and $25,616$ from RedLight. Table VII shows these results.

TABLE VII: Overall (weighted) accuracy for both datasets.

Model	Overall accuracy
Proposed	**97.1%**
Amazon [27]	**97.1%**
Clarifai [28]	94.9%
Google [29]	95.8%
Microsoft [30]	95.6%
Yahoo [19]	95.0%

Finally, in order to analyze the behavior of our proposed architecture over the PEDA 376K dataset, we fed the CNN model with all the images in the test set and extracted features from the last convolutional layer. As the dimensionality of each feature set is $1,568$, we used a combination of PCA (principal components analysis) and t-SNE (t-Distributed Stochastic Neighbor Embedding) [39] to visualize all instances, as shown in Figure 4. We can see two well-defined clusters, except for a few misclassified instances which represent less than one percent.

V. Conclusion

In this paper we introduced PEDA 376K, a novel dataset for aiding the development of research and practice in the task of automatic detection of pornographic images. Our dataset is composed of approximately $400k$ images and provides trainig/validation/tests splits, enabling comparative studies and reproducibility of experiments. Additionally, we carefully annotated each image using an objective definition of pornography.

We introduced a CNN architecture and conducted experiments with our dataset, achieving an accuracy of 99.2% in the test set. We also proposed a symbolic machine learning approach using decision trees to compare our model to existing NSFW moderation APIs under different scenarios and datasets. Our findings indicate that superior performance of the CNN architecture over all state-of-the-art NSFW APIs, even when using the decision tree optimization. Figure 5 shows a random sample of misclassified images by all of the NFSW APIs, but were correctly classified by our CNN.

Acknowledgments

The authors would like to thank the Federal University of Campina Grande (UFCG), the University of Rhode Island, and the Scientific Police Institute of Paraíba (IPC-PB). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] F. Nian, T. Li, Y. Wang, M. Xu, and J. Wu, "Pornographic image detection utilizing deep convolutional neural networks," *Neurocomputing*, vol. 210, pp. 283–293, 2016.

[2] K. Li, J. Xing, B. Li, and W. Hu, "Bootstrapping deep feature hierarchy for pornographic image recognition," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 4423–4427.

[3] S. Raaijmakers, "Artificial intelligence for law enforcement: Challenges and opportunities," *IEEE Security & Privacy*, vol. 17, no. 5, pp. 74–77, 2019.

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

[5] M. D. Putro, T. B. Adji, and B. Winduratna, "Adult image classifiers based on face detection using viola-jones method," in *2015 1st International Conference on Wireless and Telematics (ICWT)*, Nov 2015, pp. 1–6.

[6] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[7] J. Kovac, P. Peer, and F. Solina, "Human skin color clustering for face detection," in *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, vol. 2, Sept 2003, pp. 144–148 vol.2.

[8] R. Ap-Apid, "An algorithm for nudity detection," in *Proceedings of the 5th Philippine Computing Science Congress*, Cebu City, Philippines, Mar. 2005.

[9] B. Ma, C. Zhang, J. Chen, R. Qu, J. Xiao, and X. Cao, "Human skin detection via semantic constraint," in *Proceedings of International Conference on Internet Multimedia Computing and Service*. ACM, 2014, p. 181.

[10] C. Platzer, M. Stuetz, and M. Lindorfer, "Skin sheriff: A machine learning solution for detecting explicit images," in *Proceedings of the 2Nd International Workshop on Security and Forensics in Communication Systems*, ser. SFCS '14. New York, NY, USA: ACM, 2014, pp. 45–56.

[11] D. C. Moreira and J. M. Fechine, "A machine learning-based forensic discriminator of pornographic and bikini images," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[12] A. Gangwar, E. Fidalgo, E. Alegre, and V. González-Castro, "Pornography and child sexual abuse detection in image and video: A comparative evaluation," in *8th International Conference on Imaging for Crime Detection and Prevention (ICDP 2017)*, Dec 2017, pp. 37–42.

[13] I. G. P. S. Wijaya, I. Widiartha, K. Uchimura, and G. Koutaki, "Phonographic image recognition using fusion of scale invariant descriptor," in *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*. IEEE, 2015, pp. 1–5.

[14] O. Surinta and T. Khamket, "Recognizing pornographic images using deep convolutional neural networks," in *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*. IEEE, 2019, pp. 150–154.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[16] Y. Wang, X. Jin, and X. Tan, "Pornographic image recognition by strongly-supervised deep multiple instance learning," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 4418–4422.

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

[18] S. E. F. de Avila, N. Thome, M. Cord, E. Valle, and A. de Albuquerque Araújo, "Pooling in image representation: The visual codeword point of view," *Computer Vision and Image Understanding*, vol. 117, pp. 453–465, 2013.

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

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

[21] K. Zhou, L. Zhuo, Z. Geng, J. Zhang, and X. G. Li, "Convolutional neural networks based pornographic image classification," in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*. IEEE, 2016, pp. 206–209.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

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

[24] Reddit. [Online]. Available: https://www.reddit.com/

[25] A. Ananthram, "Comparison of the best nsfw image moderation apis 2018," https://towardsdatascience.com/comparison-of-the-best-nsfw-image-moderation-apis-2018-84be8da65303, 2018, accessed in September 27th, 2018.

[26] S. P. Alvarez, *RedLight an efficient illicit image detection application for law enforcement*. University of Rhode Island, 2012.

[27] Amazon rekognition. [Online]. Available: https://aws.amazon.com/pt/rekognition/

[28] Clarifai. [Online]. Available: https://www.clarifai.com/models/nsfw-image-recognition-model-e9576d86d2004ed1a38ba0cf39ecb4b1

[29] Google vision. [Online]. Available: https://cloud.google.com/vision/

[30] Microsoft azure. [Online]. Available: https://azure.microsoft.com/en-in/services/cognitive-services/content-moderator/

[31] Y. Huang and A. W. K. Kong, "Using a cnn ensemble for detecting pornographic and upskirt images," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2016, pp. 1–7.

[32] X. Ou, H. Ling, H. Yu, P. Li, F. Zou, and S. Liu, "Adult image and video recognition by a deep multicontext network and fine-to-coarse strategy," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 5, p. 68, 2017.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[35] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[36] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *CoRR*, vol. abs/1611.05431, 2016. [Online]. Available: http://arxiv.org/abs/1611.05431

[37] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4148–4158.

[38] 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.

[39] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

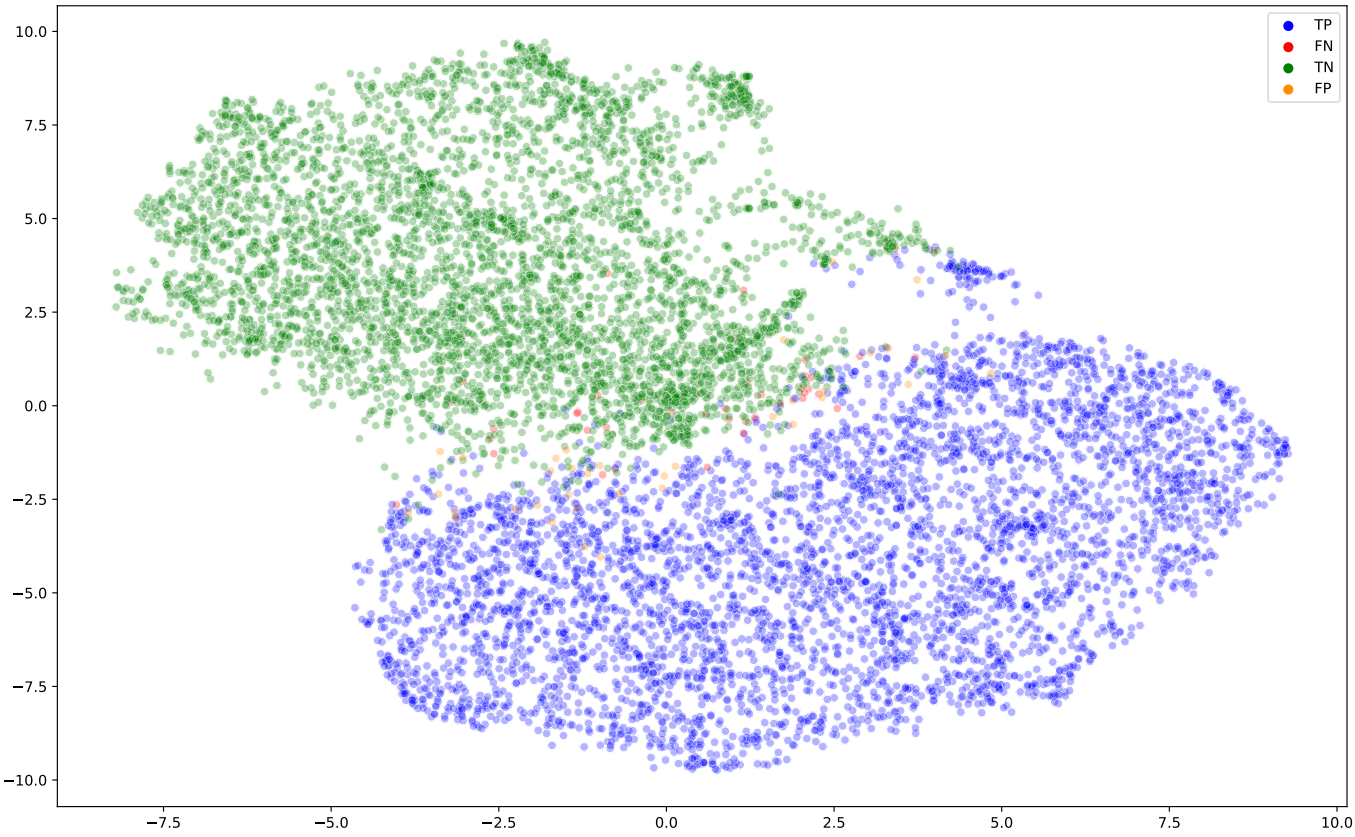

Fig. 4: The t-SNE projection showing the behavior of our proposed architecture over the PEDA 376K testset. The blue and green clusters are represented by the true positives and negatives, respectively. The few orange and red dots represent the misclassified images, respectively the false positives and negatives.

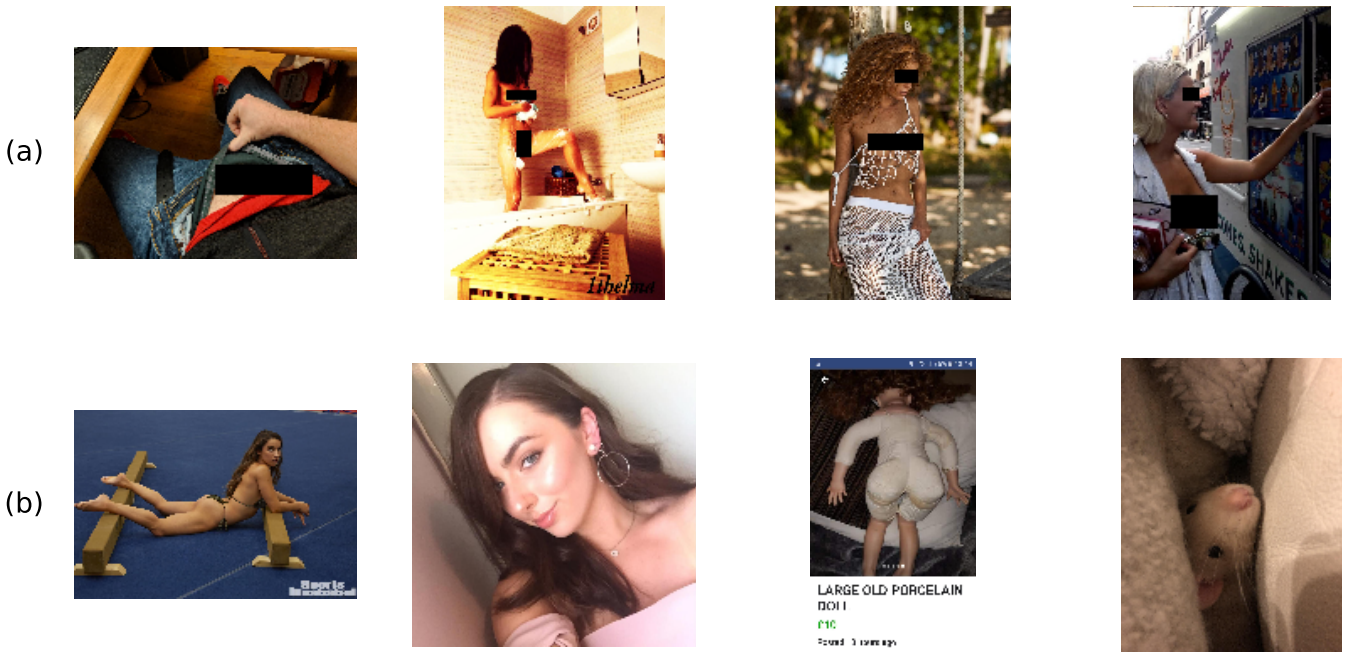

Fig. 5: Misclassified images by all optimized models that were correctly classified by our proposed model in evaluation using the PEDA 376K dataset. The first row (a) shows the pornographic images misclassified as non-pornographic (False Negatives - FN). The second row (b) exhibits the non-pornographic images misclassified as pornographic (False Positives - FP).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE