Combating the Elsagate Phenomenon: Deep Learning Architectures for Disturbing Cartoons

Akari Ishikawa, Edson Bollis, Sandra Avila

RECOD Lab., Institute of Computing, University of Campinas (UNICAMP), Brazil

Abstract—Watching cartoons can be useful for children's intellectual, social and emotional development. However, the most popular video sharing platform today provides many videos with Elsagate content. Elsagate is a phenomenon that depicts childhood characters in disturbing circumstances (e.g., gore, toilet humor, drinking urine, stealing). Even with this threat easily available for children, there is no work in the literature addressing the problem. As the first to explore disturbing content in cartoons, we proceed from the most recent pornography detection literature applying deep convolutional neural networks combined with static and motion information of the video. Our solution is compatible with mobile platforms and achieved 92.6% of accuracy. Our goal is not only to introduce the first solution but also to bring up the discussion around Elsagate.

Index Terms—Sensitive Content, Deep Learning, Elsagate

I. Introduction

Children today are part of a digital generation that has grown up in a world surrounded by technology (e.g., smartphones, tablets, electronic toys). They spend most of their time on the Internet, usually watching cartoons. Few children do not recognize the YouTube logo even from afar. Any smartphone or tablet is enough to navigate through the endless recommendation list generated by YouTube. Eventually, by following the suggested cartoons, it is unavoidable that one stumbles upon the Elsagate phenomenon [1].

In the Elsagate videos, Disney characters, superheroes and other popular childhood characters are depicted in disturbing scenes such as stealing alcohol, hurting each other, drinking from toilets, eating poop, drinking urine, smearing feces on people's faces, sexual and violent situations.

We claim that the Elsagate videos are a way of getting young children accustomed to sexual and disturbing content so pedophiles can groom them more easily. However, there is no reliable evidence on whether those claims are real, what the real motivation is, or even who are the people responsible for making these videos [2]–[4].

Elsagate channels have existed since 2014 [5]. In 2017, the term Elsagate (composed of Elsa, a character from the 2013 Disney animated film Frozen, and -gate, a suffix for scandals) became a popular hashtag on Twitter as users called attention to the presence of such material on YouTube and

YouTube Kids. On Reddit, an Elsagate subreddit (r/Elsagate) was created to discuss the phenomenon, attracting tens of thousands of users [1].

As far as we know, there is no previous work in the literature related to the Elsagate phenomenon. Also, despite the existence of good solutions towards pornography/violence detection, Elsagate videos are wildly different in several aspects. The situation is even direr not only due to the lack of solutions geared towards cartoons but also due to the similitude among them and non-sensitive cartoons. In other words, classifying a video as Elsagate or not is a challenge itself.

In this paper, we come up with solutions that take advantage of deep neural networks for disturbing Elsagate content detection in cartoons. In a nutshell, our contributions are three-fold: 1) we propose an end-to-end pipeline (features, neural network architecture, classification model) to detect Elsagate content in videos; 2) we evaluate several deep neural networks proposed for mobile platforms, and 3) we introduce the first Elsagate dataset, which comprises 285 hours (1,028,106 seconds) of 1,396 Elsagate and 1,898 non-sensitive videos.

We organize the remaining of this paper into five sections: In Section II, we review the works most related to Elsagate content detection. In Section III, we describe the proposed pipeline. In Section IV, we introduce the Elsagate dataset built and the evaluation metrics. In the Section V, we discuss the experimental results. In the Section VI, we conclude the paper proposing a solution for disturbing cartoons detection and suggesting approaches for future works.

II. Related Work

To the best of our knowledge, there are very few works related to sensitive content in cartoons. The most related ones are the following four works.

Alghowinem [6] proposed a multimodal approach to detect inappropriate content in videos from YouTube Kids. For that, one-second slices are extracted for analysis and classification. Image frames, audio signal, transcribed text and their respective features (e.g., temporal robust features (TRoF), mel-frequency cepstral coefficient (MFCC), bag-of-words) are extracted from each slice. These features are then fed into individual classifiers, which are combined using a threshold-based decision strategy. According to Alghowinem, the paper acted as a proof of concept. But, the pilot experiment is performed on three videos which are not even cartoons.

Kahlil et al. [7] tackled the problem of violence detection in cartoons. They exploited color features from seven color

channels (gray, red, green, blue, hue, saturation and value). The average dominant value, calculated from the color histograms, is used as a threshold-based classifier. As pointed by the authors, those features do not contain sufficient contextual information for content analysis. They performed the experiments on 504 clips (16,654 seconds), of which 112 have violent content. Their dataset is not publicly available.

Khan et al. [8] also explored violence detection in cartoons. They proposed a three-layered video classification framework: keyframe extraction, feature extraction using scale-invariant feature transform (SIFT), feature encoding using Fisher vector image representation and classification using spectral regression kernel discriminant analysis (SRKDA). They evaluated their approach on 100 videos, collected from various sources. The dataset (not publicly available) comprises nearly 2 hours (7,100 seconds) of 52 violent and 48 non-violent videos.

Papadamou et al. [9] studied the Elsagate phenomenon[1] using the video's titles, tags, thumbnails and general statistics (e.g., views, likes, dislikes). They proposed to process each type of feature using a different technique and to apply a fully-connected layer to combine their outputs. Despite the 82.8% accuracy achieved, their solution can be easily fooled by the uploaders since it does not take into account the video content (e.g., frames) and can disguise the features analyzed (e.g., title, thumbnail) to hide the sensitive content. Their dataset is not publicly available.

In the face of the related works, it is clear that there is a lack of research specifically for the Elsagate problem. As a matter of fact, we have in the literature plenty of solutions for sensitive content analysis but these works are focused on real-life videos with humans, regardless of the type of sensitive content (e.g., nudity [10], [11], pornography [12]–[14], child pornography [15], violence [16]–[18]).

III. METHODOLOGY

Our methodology is directly inspired by the work of Perez et al. [12], a deep learning-based solution for video pornography detection. The main reason is their flexible pipeline that allows modifications in any of its steps, letting us approach the Elsagate problem in different ways. Fig. 1 depicts a flowchart overview of the proposed method.

Although Perez et al. obtained their best results combining raw frames and optical flows, for the sake of efficiency we opted to combine raw frames and MPEG motion vectors, which are computationally very cheap. In this paper, we aim an effective solution for mobile platforms. Thus, we attempt to answer the following questions: 1) Is it better transferring knowledge (features) from a related dataset (e.g., Pornography) or an unrelated dataset (e.g., ImageNet)?, 2) Which deep learning architecture offers the best Elsagate classification performance regarding a mobile platform?

In the next subsections, we detail our pipeline. In Section III-A, we present the static/motion features that we used as input to the deep neural networks. In Section III-B, we
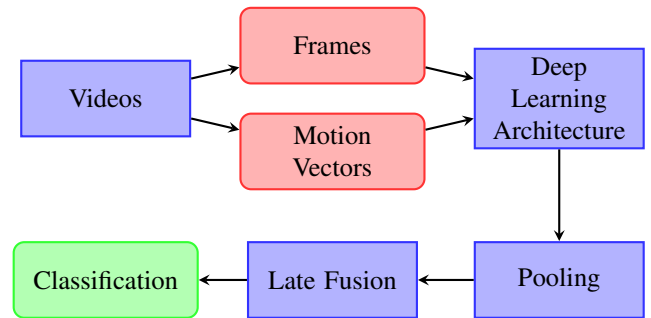
[1]This paper was not available before our submission.

Fig. 1. Overview of the proposed method for detecting Elsagate content.

overview the deep learning architectures that we evaluated: GoogLeNet [19], SqueezeNet [20], MobileNetV2 [21], and NASNet [22]. Finally, in Section III-C, we describe the fusion strategy adopted in this work.

A. Features

1) Static Information: Deep neural networks have been successfully used to directly process raw data as input [19], [21], [23]. To feed the video static information to the deep learning architectures, we extracted the raw frames from the videos using a frame sampling of one frame per second (1 fps). We then centered the frames, resized to 224×224 pixels (the input size of GoogLeNet, SqueezeNet, MobileNetV2, and NASNet) maintaining the aspect ratio, and cropped the remaining pixels in the largest dimension.

2) Motion Information: Incorporating motion information into local descriptors and deep neural networks leads to more effective sensitive videos classifiers [12], [17], [24]. In this paper, we aim to develop deep learning-based approaches for automatically extracting discriminative space-temporal information for filtering Elsagate content, with a good compromise between effectiveness and efficiency. For that, we opt to extract MPEG motion vectors [25], as suggested by Perez et al. [12].

Motion vectors can be decoded directly from standard video compressed files with very low computational cost. The decoding process includes many subprocesses such as motion compensation, inverse discrete cosine transform, and variable length decoding. The key process in decoding is the motion compensation by inter-frame prediction. In the motion compensation, a motion vector, which is a translation offset from the reference frame to the target frame, represents the movement of the macroblock (small regions in each frame).

We calculate the MPEG motion vectors that map the motion between the reference and the current frame, using macroblocks of size $M \times N$ (usually M and N equals to 8 or 16). Fig. 2 depicts a macroblock and its respective motion vector in the reference and the current frame.

B. Deep Learning Architectures

Since the success of the ImageNet challenge [23], deep learning-based methods have drawn a tremendous amount of attention from academia, industry, and media. Subsequent

Fig. 2. Example of macroblock and its respective motion vector in the reference frame (X, Y) and in the current frame (X', Y') [12].

works show that deep neural networks achieve the state-of-the-art results on many real-world tasks [26]. In this paper, we evaluate four different deep learning architectures (DLAs) on Elsagate classification problem.

1) GoogLeNet: GoogLeNet [19] won the 2014 ImageNet Challenge, with an error rate of 6.7%, an improvement of 55% compared to the AlexNet of Krizhevsky et al. [23].

The network architecture is quite different from previous ones (VGGNet, ZFNet, and AlexNet). It performs convolution on input with three different sizes of filters (1×1, 3×3, 5×5), and stacks all the outputs. This technique is called *inception module*. Before its inception idea, most popular DNNs just stacked convolution layers deeper and deeper.

GoogLeNet (or Inception v1) has nine inception modules stacked linearly, 22 layers deep, and 6.8 million parameters. Also, it uses a global average pooling at the end of the last inception module.

2) SqueezeNet: SqueezeNet [20] is an architecture that achieves AlexNet-level accuracy with $50\times$ fewer parameters.

The building block of SqueezeNet is called *fire module*, which contains two layers: a *squeeze* layer (convolution layer of only 1×1 filters) and an *expand* layer (convolution layer with a mix of 1×1 and 3×3 filters). The squeeze layer and expand layer keep the same feature map size: while the former reduces the depth to a smaller number, the latter increases it.

SqueezeNet contains eight fire modules, with a variable number of convolution filters, and uses 1.2 million parameters.

Its last layer is a fully-connected layer with 1000 neurons and a softmax activation. Differently from other architectures evaluated in this paper, where we used the last (or previous ones) as the feature vector, none of the SqueezeNet's layers were suitable for our purpose. Either because of its size (for example, the layer immediately before the last one has a $512\times13\times13$ dimension), either due to its sparsity. Thus, to make use of SqueezeNet, we added another fully-connected layer with two classes after the already existing one and considered the 1000 neurons layer as the feature vector.

3) MobileNetV2: MobileNetV2 [21] is an architecture developed to mobile platforms. It uses separable convolutions to reduce the number of parameters: depthwise separable convolutions and bottleneck blocks. In depthwise convolution [27], convolution is performed independently for each of the input channels. This significantly reduces the computational cost by omitting convolution in the channel domain. Bottlenecks [28]

reduce the amount of data that flows through the network. In the bottleneck block, the output of each block is a bottleneck.

The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers, and uses 2.3 million parameters.

4) NASNet: NASNet [22] architecture is constructed using the Neural Architecture Search (NAS) framework. The goal of NAS is to use data-driven for constructing the network architecture. Szegedy et al. [29] showed that a complex combination of filters in a 'cell' can significantly improve results. The NAS framework defines the construction of such a cell as an optimization process and then stacks the multiple copies of the best cell to construct a large network.

In this paper, we use the NASNet-A (4 @ 1056), where 4 indicate the number of cell repeats and 1056 the number of filters in the penultimate layer of the network. It uses 5.3 million parameters.

C. Late Fusion

In this fusion scheme, each information is processed by a separate decision-making approach (e.g., support vector machine (SVM) classifier), generating independent classification scores that are combined on a single score for the final classification. Similarly to Perez et al. [12], we employ late fusion taking the mean of the probabilities from static and motion information, making a more precise classification.

IV. EXPERIMENTAL SETUP

We describe here the general experimental setup, such as the first Elsagate dataset and the metrics we use to assess the performance of the Elsagate classifiers. All material related to this paper (dataset, code, models) is available at https://github.com/akariueda/DLAforElsagate.

To evaluate the results of our experiments, we apply a 1×2-fold cross-validation protocol. It consists of randomly splitting the dataset into two folds. Then, we switched training and testing sets and, consequently, we conducted two analyses for every model.

A. Elsagate Dataset

We introduce the first publicly available Elsagate video dataset. It comprises 285 hours (1,028,106 seconds) of 1,396 Elsagate and 1,898 non-sensitive videos. To put the number in perspective, the largest sensitive video dataset (Pornography-2k dataset [24]) contains 140 hours. It is worth mentioning the Elsagate dataset is composed of cartoons only.

Concerning the Elsagate class, we downloaded videos from YouTube channels reported by Reddit users in the thread "What is Elsagate?" [1]. With respect to non-sensitive content, we collected videos from official YouTube channels (e.g., Cartoon Network, Disney Channel).

On February 2018, we gathered a training/validation set with 1,567 non-sensitive and 1,118 Elsagate videos. On September 2018, we collected as a test set 331 non-sensitive and 278 Elsagate videos, totaling 1,898 non-sensitive and 1,396 Elsagates. The period between the two data gathering

(a) Non-sensitive

(b) Elsagate

Fig. 3. Example frames from the Elsagate dataset.

was purposeful to evaluate our model with new videos that could appear. The classes are imbalanced to offer a representative dataset. Fig. 3 depicts some frames from the dataset.

The Elsagate dataset is available free of charge to the scientific community but, due to the potential legal liabilities of distributing sensitive/copyrighted material, the request must be formal and a responsibility term must be signed.

B. Evaluation Metrics

To assess the performance of the Elsagate classifiers, we report the normalized accuracy and F_2-measure in all experiments.

- Normalized Accuracy (ACC): measures the classifier's success rate, independently from the classes' labels. Mathematically, it is expressed as:

$$ACC = (TPR + TNR)/2$$

where TPR is the True Positive Rate, and TNR is the True Negative Rate.

- F_2-measure (F_2): it is the harmonic mean of precision and recall where recall has double the precision's weight ($\beta = 2$). In this work, the F_2-measure is crucial since the false negatives are unacceptable as it means the sensitive material passed the filtering. It is considered less harmful to wrongly deny the access to non-sensitive material than to expose sensitive content by mistake. The F_2-measure can be defined as:

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

where the parameter β denotes the importance of recall in relation to precision.

V. RESULTS AND DISCUSSION

In this section, we evaluate the performance of different strategies on the Elsagate dataset.

A. Transfer Learning

Is it better transferring from a related dataset (e.g., Pornography) or an unrelated dataset (e.g., ImageNet)? For this analysis, we used the GoogLeNet model, pre-trained for pornography in real-life (with humans) videos. Perez et al. [12] kindly provided the weights trained for pornography.

Here, we aimed to evaluate the transfer learning exclusively. Note that in our pipeline (Fig. 1), the deep neural network is used only as a feature extractor. In the classification step, we trained SVM models using our data.

In Table I, we observe the accuracy obtained with GoogLeNet pre-trained on ImageNet and Pornography videos. The network trained on ImageNet showed better results. For that reason, in the next experiments, we used pre-trained models on ImageNet.

TABLE I
ACCURACY OF THE PEREZ ET AL. [12] MODEL IN ELSAGATE'S VIDEOS.

Features	ImageNet (%)	Pornography (%)
Frames	92.7	91.9
Motion Vectors	91.3	92.3
Late Fusion	96.1	94.8

B. Mobile Deep Learning Architectures

Which deep learning architecture offers the best Elsagate classification performance, regarding a mobile platform? We reproduced the experiment of the previous section now evaluating SqueezeNet, NASNet, and MobileNetV2. Fig. 4 compares the accuracy and F_2-measure of the three DLAs.

In the transfer learning (Fig. 4 (a) and (b)), SqueezeNet outperformed the other two architectures. But, it is worth mentioning that in preliminary experiments, using other SqueezeNet layers as feature vectors, it reported poor results. This indicates that SqueezeNet is not an adequate feature extractor for Elsagate content. The modifications we introduced to the architecture produced this curious outcome.

Although transfer learning provided a reasonable suggestion about the DLAs feature extraction capacity, it still does not give us a sense of their learning capability. In the finetuning experiments, we initialized the DNNs with the transferred weights and then retrained all layers. Some experimentation involving freezing initial layers did not report good results.

In Fig. 4 (c) and (d) we show the results for the finetuned models. Surprisingly, we noted that NASNet, which previously showed the worst performance, reached an accuracy of 95.3% and F_2 of 92.9%, surpassing both SqueezeNet and MobileNetV2 and competing closely to GoogLeNet, an architecture with the double of parameters. NASNet adaptation capability is comprehensible since it was developed through an architecture search method, as a joint of layers specialized for large datasets for classification (e.g., ImageNet) and detection tasks (e.g., COCO Object Detection).

In contrast with NASNet's improvements, we observed a reduction in SqueezeNet's results in all features (frames, motions and late fusion) and with MobileNetV2's frames. We
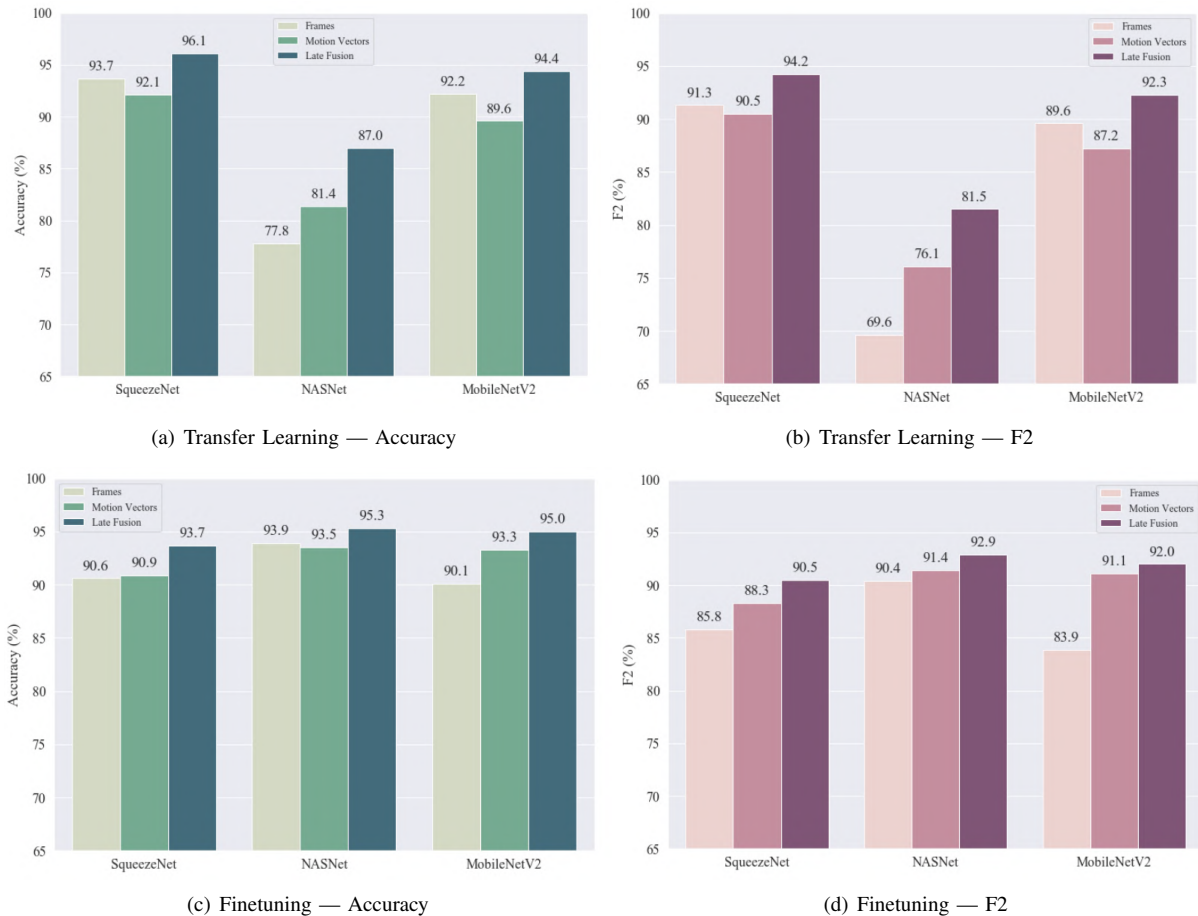
(a) Transfer Learning — Accuracy

(b) Transfer Learning — F2

(c) Finetuning — Accuracy

(d) Finetuning — F2

Fig. 4. Accuracy and F_2 of SqueezeNet, NASNet and MobileNetV2 in transfer-learning and finetuning.

hypothesize that the models that already report a fair performance (SqueezeNet achieved a 96.1% accuracy in transfer learning) are likely to stumble upon a "high accuracy barrier" and end up unlearning the task.

C. Testing

Despite the interesting improvement in the NASNet-based model, we note that SqueezeNet's results are slightly higher. Since it is expected that finetuning a model brings better results, we downloaded a test set to help us decide upon this dilemma and define our final model. As a decisive experiment, we also used the SVM models trained on the training set.

Although we had downloaded the test set around seven months later than the training set, a reliable model is supposed to be robust enough to deal with the unexpected. The results, shown in Table II, illustrate an impressive decline in SqueezeNet's performance. Thus, as our final model, we chose NASNet transferred from ImageNet and finetuned to Elsagate.

D. Model Interpretation

At the same time that accuracy and F_2 are proper metrics to compare the architectures, they do not tell much about the model interpretability. Until now, we have only assessed

TABLE II
SQUEEZENET TRANSFERRED MODEL AND NASNET FINETUNED MODEL
PREDICTING THE TEST SET.

	Train Set		Test Set	
	ACC (%)	F2 (%)	ACC (%)	F2 (%)
SqueezeNet (Transfer)	96.1	94.2	62.0	37.8
NASNet (Finetuned)	95.3	92.9	92.6	88.7

the models regarding the number of Elsagate videos they were able to filter. Thus, an interesting object of study is to understand the model's behavior towards the videos and to speculate possible reasons that led to errors. Here, we analyze and discuss the false negatives and false positives all the three architectures.

Although Elsagate is calling people's attention due to the bizarre topics depicted (e.g., gore or toilet humor), some softer topics such as nursery rhymes and counting numbers and colors are also considered Elsagate (accordingly to /r/Elsagate), leading to some controversy. Due to the subjectivity of the Elsagate content, we could say that the low-quality production of the video classifies it as an Elsagate, or also that the characters expressions, sounds, and groanings may cause a disturbance.

(a) Elsagate nursery rhyme   (b) Nursery rhyme

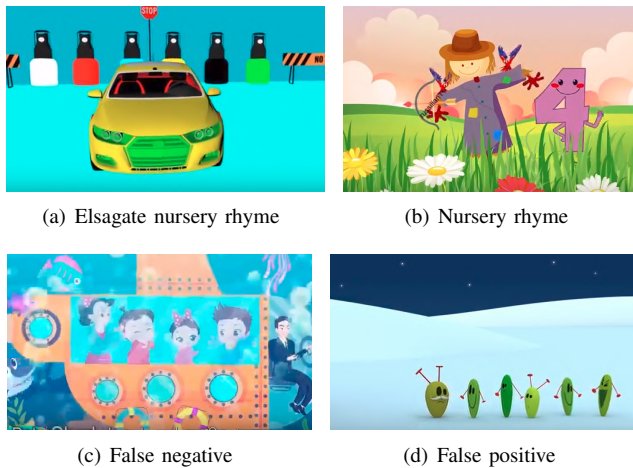(c) False negative   (d) False positive

Fig. 5. Sample frames of videos misclassified by all the architectures.

Not surprisingly, most of the misclassified videos were nursery or counting rhymes. We also noted the presence of this kind of video in both classes, although the ones in the non-sensitive class have a much higher quality (Fig. 5 (a) and (b)). Therefore, regarding those videos, even if considered Elsagate, they are much less harmful than the average. So, we believe this model's behavior is acceptable.

Concerning the other false positives and negatives, in some videos the same Elsagate characters were not in a grotesque circumstance (Fig. 5 (c)). Also, in other videos, the appearance (e.g., color-palette, motion patterns) resembles Elsagate (Fig. 5 (d)). Besides, we had many non-sensitive videos containing controversial scenes (e.g., fighting, naughtiness, accidents) that could be considered Elsagate content.

VI. CONCLUSION

In this paper, we brought up to the literature the discussion about the Elsagate phenomenon, and we proposed the first solution to solve the problem. Inspired by the most recent approaches in sensitive content detection, we evaluated popular DLAs such as GoogLeNet, SqueezeNet, NASNet and MobileNetV2 and both static and motion information of the video. Our experiments suggested that NASNet is an excellent feature extractor when applying transfer learning from ImageNet, followed by a finetuning to Elsagate. Even with the challenges imposed by the problem itself (e.g., the lack of a formal definition of Elsagate), we achieved a 92.6% of accuracy. As future work, we intend to embed the solution in a mobile application and to propose a more deep annotation for studying the phenomenon itself. We hope this work stimulates the development of better disturbing content filtering solutions.

REFERENCES

[1] "r/Elsagate: What is Elsagate?" https://www.reddit.com/r/ElsaGate/\comments/6o6baf/what_is_elsagate/.

[2] "Youtube's elsagate illuminates the unintended horrors of the digital age," https://www.forbes.com/sites/danidiplacido/2017/11/28/youtubes-elsagate-illuminates-the-unintended-horrors-of-the-digital-age/, 2017.

[3] "The disturbing videos that are tricking children," http://www.bbc.com/news/blogs-trending-39381889, 2017.

[4] "On youtube kids, startling videos slip past filters," https://www.nytimes.com/2017/11/04/business/media/youtube-kids-paw-patrol.html, 2017.

[5] K. J. Dailys, "Crude parodies of kids movies can't be stopped," http://koreajoongangdaily.joins.com/news/article/article.aspx?aid=3043884.

[6] S. Alghowinem, "A safer youtube kids: An extra layer of content filtering using automated multimodal analysis," in *IntelliSys*, 2018, pp. 294–308.

[7] T. Khalil, J. I. Bangash, Abdusalam, and A. Adnan, "Low level visio-temporal features for violence detection in cartoon videos," in *INTECH*, 2016, pp. 320–325.

[8] M. Khan, M. A. Tahir, and Z. Ahmed, "Detection of violent content in cartoon videos using multimedia content detection techniques," in *INMIC*, 2018.

[9] K. Papadamou, A. Papasavva, S. Zannettou, J. Blackburn, N. Kourtellis, I. Leontiadis *et al.*, "Disturbed youtube for kids: Characterizing and detecting disturbing content on youtube," *arXiv*, vol. 1901.07046, 2019.

[10] A. Lopes, S. Avila, A. Peixoto, R. Oliveira, and A. Araújo, "A bag-of-features approach based on hue-SIFT descriptor for nude detection," in *EUSIPCO*, 2009, pp. 1152–1156.

[11] A. Lopes, S. Avila, A. Peixoto, R. Oliveira, M. Coelho, and A. Araújo, "Nude detection in video using bag-of-visual-features," in *SIBGRAPI*, 2009, pp. 224–231.

[12] 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.

[13] C. Caetano, S. Avila, W. Schwartz, S. Guimarães, and A. Araújo, "A mid-level video representation based on binary descriptors: A case study for pornography detection," *Neurocomputing*, vol. 213, 2016.

[14] 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.

[15] P. Vitorino, S. Avila, M. Perez, and A. Rocha, "Leveraging deep neural networks to fight child pornography in the age of social media," *J Vis Commun Image Represent.*, vol. 50, pp. 303–313, 2018.

[16] S. Avila, D. Moreira, M. Perez, D. Moraes, I. Cota, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "RECOD at MediaEval 2014: Violent scenes detection task," in *MediaEval 2014 Workshop*, 2014.

[17] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Temporal robust features for violence detection," in *WACV*, 2017, pp. 392–399.

[18] B. M. Peixoto, S. Avila, Z. Dias, and A. Rocha, "Breaking down violence: A deep-learning strategy to model and classify violence in videos," in *ARES*, 2018, pp. 50:1–50:7.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[20] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," *arXiv*, vol. 1602.07360, 2016.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[22] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *CVPR*, 2018.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[24] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Pornography classification: The hidden clues in video space-time," *Forensic Sci Int*, vol. 268, pp. 46–61, 2016.

[25] ISO/IEC 14496-10:2014, "Information technology – coding of audio-visual objects – part 10: Advanced video coding," *Standard, 9*, 2018.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[27] L. Sifre and S. Mallat, "Rigid-motion scattering for image classification," Ph.D. dissertation, École Polytechnique, 2014.

[28] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, "SqueezeNext: Hardware-aware neural network design," in *CVPR*, 2018, pp. 1751–1760.

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI Conference*, 2017, pp. 4278–4284.