Stealthy Porn: Understanding Real-World Adversarial Images for Illicit Online Promotion

Kan Yuan*, Di Tang†, Xiaojing Liao*, XiaoFeng Wang*,
Xuan Feng*‡, Yi Chen*‡, Menghan Sun†, Haoran Lu*, Kehuan Zhang†
*Indiana University Bloomington, †Chinese University of Hong Kong, ‡Chinese Academy of Sciences
*{kanyuan, xliao, xw7, haorlu}@indiana.edu, †{td016, sm017, khzhang}@ie.cuhk.edu.hk,
‡{fengxuan, chenyi}@iie.ac.cn

***Abstract*—Recent years have witnessed the rapid progress in deep learning (DL), which also brings their potential weaknesses to the spotlights of security and machine learning studies. With important discoveries made by adversarial learning research, surprisingly little attention, however, has been paid to the real-world adversarial techniques deployed by the cybercriminal to evade image-based detection. Unlike the adversarial examples that induce misclassification using nearly imperceivable perturbation, real-world adversarial images tend to be less optimal yet equally effective. As a first step to understand the threat, we report in the paper a study on adversarial promotional porn images (APPIs) that are extensively used in underground advertising. We show that the adversary today's strategically constructs the APPIs to evade explicit content detection while still preserving their sexual appeal, even though the distortions and noise introduced are clearly observable to humans.**

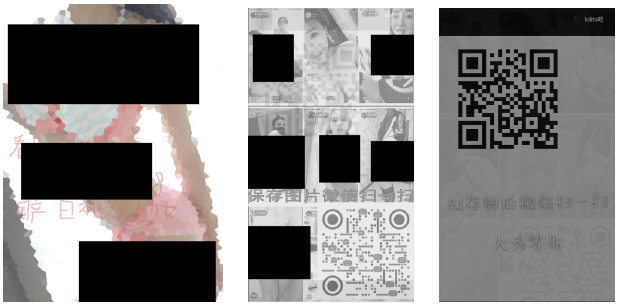
To understand such real-world adversarial images and the underground business behind them, we develop a novel DL-based methodology called Malèna, which focuses on the regions of an image where sexual content is least obfuscated and therefore visible to the target audience of a promotion. Using this technique, we have discovered over 4,000 APPIs from 4,042,690 images crawled from popular social media, and further brought to light the unique techniques they use to evade popular explicit content detectors (e.g., Google Cloud Vision API, Yahoo Open NSFW model), and the reason that these techniques work. Also studied are the ecosystem of such illicit promotions, including the obfuscated contacts advertised through those images, compromised accounts used to disseminate them, and large APPI campaigns involving thousands of images. Another interesting finding is the apparent attempt made by cybercriminals to steal others' images for their advertising. The study highlights the importance of the research on real-world adversarial learning and makes the first step towards mitigating the threats it poses.

I. Introduction

Adversarial learning aims at understanding the weaknesses of machine learning in the adversarial environment and developing protection against potential threats. Research along this line can be traced back a decade ago, to evasive attacks on intrusion detection systems [32] and spam filters [59], and to data contamination risks in classifiers [60]. More recently, the rapid progress of deep neural networks (DNN) and their wide adoption in image processing have moved the focus of adversarial learning to these models' vulnerabilities towards *adversarial examples*: it has been found that a small amount of noise, once added to an image, could cause a DNN to misclassify the image, even when the modified image looks almost indistinguishable from the original one to humans. Given the security-critical applications of the DNN-based image classification, like self-driving cars, face recognition based authentication, etc., such risks have aroused a great deal of interest from the security community as well as the industry, even though no evidence has yet been found that related attacks have ever taken place in the real life. In the meantime, surprisingly little attention has been paid to the adversarial techniques actually employed by real-world cybercriminals, particularly those against image classification systems, which turn out to be quite different from those intensively studied by the aforementioned research [37], [49], [55].

Adversarial explicit content. More specifically, anecdotes have it that obfuscated images have been extensively used by the underground businesses for illicit online advertising (Ad), phishing and other insidious purposes. Unlike the old style image spam, where spam messages are directly embedded in pictures, today's promotional images include explicit sexual and violence content to attract audience and various obfuscation tricks to hide them from automatic content checkers, such as Google Cloud Vision API, Baidu AipImageCensor API, Clarifai NSFW API. Examples of such images (with proper masking) are presented in Figure 1. Compared with the adversarial examples studied by the ongoing adversarial learning, such *adversarial explicit content* does not need to be optimized in a sense that the perturbation introduced to an image remains less perceivable to humans. Instead, all the adversary wants is just to get the semantics (e.g., pornography) through without triggering the alarm (e.g., Google SafeSearch filter). This lowers the bar to constructing the attack instances and raises the challenges for finding them. Indeed, so far, little has been done to systematically discover and analyze the adversarial explicit content, not to mention any effort to understand the underground ecosystem behind these images.

Malèna: finding stealthy porn. In this paper, we report the first systematic study on the adversarial explicit content, focusing on *adversarial promotional porn images* (APPIs), based upon a novel methodology for a large scale discovery of such images. More specifically, we developed a Malicious Explicit Content Analyzer, called Malèna (Section III), leveraging two key observations about these obfuscated images: they need to include promotional information (links, phone



(a) APPI 1 (b) APPI 2 (open with Firefox) (c) APPI 2 (open with Chrome)

Fig. 1: Examples of APPIs. (b) and (c) show a special APPI that displays different content on white and black background.

number, etc.) for follow-up and they cannot obfuscate all the obscene image content or risk losing interest from the target audience. Exploiting these observations, our approach first identifies the pictures carrying text or numbers or QR code and then performs a Region-of-Interest (ROI) processing to find the persons in each image. After that, we run a DNN-based explicit content detector on each identified region (including the mask of the person detected from the ROI) to discover pornographic content. In this way, we are able to significantly reduce the image regions to which noise can be injected for inducing misclassification, and achieve a precision and a recall of 91% and 85%.

Measurement and discoveries. Running Malèna on the data crawled from 2 forums, including Baidu Tieba [1], the largest Chinese largest Chinese communication platform provided by the Chinese search engine company, Baidu, and Sina Weibo [9], a Chinese microblogging website, we were able to detect over 4,000 confirmed APPIs from totally 4,042,698 images downloaded. Analyzing these images, we discovered interesting obfuscation techniques deployed in underground advertising, such as adding high-frequency signals (e.g., texturing and noising) or filter effects (e.g., blurring) to an image. Of particular interest is the observation that some images have been converted from the RGB color space into grayscale to evade skin-related features widely utilized in explicit content detection.

Those APPIs turn out to be quite effective in evading state-of-the-art explicit content detectors such as Google Cloud Vision API, Baidu AipImageCensor API, Yahoo Open NSFW model, and Clarifai NSFW API: we observed that 35.6% of the APPIs circumvented all four detectors. Further, we looked into the open-source Yahoo NSFW model, a convolutional neural network model, to find out how it missed those APPIs. Particularly, we examined the output of its first convolution layer, which is used to extract image edge features, and found that the obfuscations performed on the APPIs significantly degrade the qualities of these features.

Further using the links or the WeChat numbers promoted by those images, we were able to analyze the ecosystem behind such obfuscated pornographic pictures. More specifically, from the images, we discovered 31 URLs, 76 QQ IDs, 245 WeChat IDs, and 45 QR code and 266 other contacts. Such information was obfuscated in some cases, using jargons, emojis or homophonic characters, apparently in an attempt to evade the OCR based text detection. Further we studied the ways such APPIs are disseminated: for example, 3,080 accounts on Baidu Tieba and 472 accounts on Sina Weibo were found to be involved in the distribution of APPIs, where 1,676 Baidu Tieba and Sina Weibo account were compromised legitimate accounts. Also discovered in our research was a huge APPI spam campaign, which used 1,325 APPIs to promote more than 7 illicit mobile sexual apps. Interestingly, we observed that APPIs were being reused, with their original promotional content erased, which indicates possible competitions among cybercriminals.

Contributions. The contributions of the paper are outlined as follows:

• *New understanding about the use of adversarial images in the cybercrime*. We report the first systematic study on the real-world adversarial images and their use in online illicit promotions. Our study sheds light on how evasive techniques are deployed by cybercriminals to evade image detection systems and identifies the gap between these techniques and what have been studied in the ongoing adversarial learning. Further our study brings to light the ecosystem behind such illicit promotions, which is important for finding technical and policy means to address such new security challenges.

• *New techniques for finding real-world adversarial images*. We developed a novel methodology to identify those adversarial images, which demonstrates to be highly effective on today's APPIs. Although explicit content detection likely continues to be in an arm race with cybercriminals, our approach raises the bar to the evasive attacks, and makes a first step toward more effective control of this emerging threat.

Roadmap. The rest of the paper is organized as follows: Section II presents the background of our study; Section III elaborates the design of Malèna and Section IV presents its implementation and evaluation; Section V describes our large-scale measurement study on APPIs using Malèna; Section VI continues to unravel the underground ecosystem behind those images; Section VII discusses the limitation of our current research and ethical issues; Section VIII surveys the related prior research and Section IX concludes the paper.

II. Background

A. Promotional Explicit Content

Promotional explicit content aims to utilize sex appeal images (e.g., explicit displays of sexual acts and seductive behavior) for advertising, typically through injecting promotional URLs, QR codes or instant message app IDs into a pornographic image. Such content has been used to serve various purposes, such as phishing and promotion of counterfeit products, illicit online pharmacy, gambling or porn sites, etc.

Dissemination of explicit content has been controlled in many countries. For example, in US, the Child Online Protection Act restricts the exposure of such content to minors, though the legal status of Internet pornography is still less clear for adults; in China, explicit content has been forbidden by its Cybersecurity law [4]. Also, regulations on sexual materials have been put in place by the industry. As an example, Twitter does not allow adult content to be used as a user profile or header image [11], Google provides its SafeSearch lock to protect minors and all mainstream Chinese social networks (e.g., Sina Weibo [9]) and online forums (e.g., Baidu Tieba [1]) prohibit explicit content [10] [2]. Further Google, Yahoo, Microsoft, Baidu and others all provide their inappropriate content detection services. Examples include Google Cloud Vision API [7] and Baidu AIP ImageCensor API [18]. In response to such control and censorship, the underground advertiser starts to utilize adversarial images to evade the detection, as observed in our research.

B. Image Processing

Center to the arm race between the underground advertiser and the explicit content regulator are image processing techniques, which we briefly introduce below.

Object recognition. Object recognition is a computer vision task for detecting and recognizing the instances of semantic objects in a certain class from images or videos. Although this ability comes naturally to humans, it is actually fairly challenging for computers. Many solutions have been proposed in the past several decades, from traditional feature-based approaches to deep learning.

Among the most influential object recognition techniques today is the Regions with Convolutional Neural Networks (R-CNN) [34], [35], [39], [53]. More specifically, in R-CNNs, a manageable number of "regions of interest" or "ROIs", that may contain object instances, are first identified. Then a convolutional neural network (CNN) [42] is applied on each region candidates to extract features independently for classification. Particularly, Mask R-CNN [39] is a cutting-edge R-CNN technique. Besides reporting the object type and the corresponding bounding box, Mask R-CNN also segments the object from the bounding box, which is achieved by adding a Fully Convolutional Network (FCN) [44] on top of the feature map extracted by the CNN. The FCN predicts a binary mask indicating whether or not a given pixel is part of the object. Hence, the recognized object can be segmented in pixel-level with high quality. In our research, we utilize the segmentation mask generated by Mask R-CNN to degrade the interference of obfuscation techniques in the image and to recognize the explicit content effectively.

Scene text detection. Scene text is the text content that appears in an image, which may vary in shape, font, color, orientation and position across images. In our research, we utilized an off-the-shelf scene text detection tool, PixelLink [31], to capture it for analyzing the promotional content it advertises. More specifically, PixelLink uses a neural network to perform a pixel-level text/non-text prediction, first to find out how likely a pixel is part of a text instance, and then to determine whether adjacent pixel pairs can be linked together and related to the same instance. After that, it performs text instance segmentation through joining these linked text pixels to detect the content of the text. According to the prior research [31], PixelLink can achieve a precision of 87.5%, recall of 88.6%, and F-score of 88.1%.

Explicit content detection. In our research, we ran popular explicit content detectors on APPIs to understand the effectiveness of real-world adversarial images in evading these machine learning models. Such tools include Google Cloud Vision API [7], Yahoo "Not Suitable for Work" (NSFW) Image detector [17], Baidu AIP ImageCensor API [18], and Clarifai NSFW API [5].

Google Cloud Vision API is capable of detecting faces, objects and text content from an image. It uses the machine learning models that also power SafeSearch [8] to capture five categories of inappropriate content, including adult, spoof, medical, violence, and racy. For each category, the API returns one of five possible likelihood values: "VERY_UNLIKELY", "UNLIKELY", "POSSIBLE", "LIKELY", or "VERY_LIKELY".

Baidu AIP ImageCensor API provides a series of image recognition interfaces such as pornography recognition, terrorism identification, etc. Given an input image, the pornography recognition API rates it with a porno level, which can be "NORMAL", "SEXY", or "PORN", as well as a list of probabilities indicating whether the image belongs to a certain category of pornography.

In 2017 Yahoo open-sourced its deep learning model for NSFW detection. The model rates an image using a score between $0-1$: a score below 0.2 indicates that the image is likely to be safe with high confidence, while the images rated above 0.8 are considered to be NSFW; those in-between could be binned based upon their different NSFW levels. Also a similar NSFW API is provided by Clarifai [5].

C. Threat Model

In our research, we consider an adversary who tries to use the adversarial promotional explicit images to evade inappropriate image detectors for promoting illicit products (e.g., sexual products, gamble sites, illicit online pharmacies, etc.). For this purpose, the adversary can obfuscate the image, using various distortion techniques (such as noise, blur and occlusion). However, we assume that such adversarial promotional explicit images, even evasive and distorted, should still be correctly recognized by humans.

III. Adversarial Explicit Image Identification

Here we elaborate the technique we used to identify *adversarial promotional porn images* (APPI), starting with an overview of the idea behind our detection tool, Malèna, which is followed by the design details of each component.

A. Overview

To make their images less detectable and therefore less likely to be removed from high-profile forums, the adversary increasingly introduces strong distortions to obfuscate explicit image content. Finding such images in a large scale is challenging due to the stealthy nature of APPIs, which circumvent at least one existing detector, as observed in our research (Section V-B). To detect such images, we leverage two unique features of the APPIs. First, to promote illicit products, these images must contain promotional content such as text or QR codes. So we can use a scene text detection tool to capture the images with such content. More importantly, to preserve some level of sexual appeal, some explicit content of an APPI needs to be less obfuscated, which makes it easier to identify by ROI-based detection. More specifically, our Malicious Explicit Content Analyzer (Malèna) runs an R-CNN to locate all *regions of interest* from an image and then checks the presence of explicit content within individual ROIs. Once found, such an image is scanned by four mainstream detectors (Google Cloud Vision API, Baidu AipImageCensor API, Yahoo Open NSFW model, and Clarifai NSFW API). Only those successfully evade at least one detectors but flagged by Malèna are reported as APPIs.

Architecture. As illustrated in Figure 2, Malèna consists of four components: *preprocessor*, *promotional content identifier*, *regional explicit content detector* and *evasiveness checker*. The preprocessor identifies the format of an input image and unifies its color space (Section III-B). Then, the promotional content identifier is run to seek scene text and QR codes from the image, and drops it if neither can be found (Section III-C). Otherwise, the image is further analyzed by the regional explicit content detector (Section III-D), which locates all ROIs and inspects each region for explicit content. Once discovered, the image is sent to the evasiveness checker to find out whether it can be detected by any of the four mainstream detectors (Section III-E) and flagged when it cannot.

B. Preprocessing

Online forums often receive images in various formats, such as JPG, PNG, GIF etc. For simplicity, some forums change the extensions of all such image files to the same one (e.g., .JPG) without actually altering their formats. For example, all images scrapped from Baidu Tieba [1], the largest online forum in China, have the ".JPG" extension, though their true formats can be not only JPG but also PNG and GIF, which can all be displayed by the same interpretor.

Format recognition. To analyze these images for explicit content, the preprocessor first identifies their real formats, which is necessary for properly processing animated images such as GIFs, and filtering out non-image files with image file extensions. For this purpose, we utilize *Libmagic* [15], a library for recognizing image formats from their magic numbers.

Animation processing. Once the correct format is discovered, the preprocessor throws away non-image files, keeps static images and breaks an animated image such as GIF into a set of pictures using Python Imaging Library [23]. Note that animated images are very common amongst APPIs since they are not only eye-catching but also hard to detect. In our research, we process such an image based upon the relations among its consecutive frames: when each frame looks similar to its subsequent one (e.g., video), our approach just picks out the first frame as the image's representative for the follow-up analysis; when every frame turns out to be quite different from the next one (e.g., slide show), we keep all frames for explicit content detection.

Specifically, our preprocessor runs a uniformity check on all the frames of an animated image, based upon *perceptual hash* (pHash) [12]. Perceptual hashing summarizes an image into a short bit string, 128 bits used in our research. Two similar images have a small Hamming distance between their strings, while dissimilar ones are distance away. Our preprocessor measures the similarity between two consecutive frames according to the ratio of their Hamming distance (the distance divided by the string length): the ratio is 0 if the two frames are identical, 0.5 if totally different, and 1 if one is the exact inversion of the other. We consider that an animated image is not "uniform" if the average similarity score across all its consecutive frame pairs is between 0.4 and 0.6. In this case, all frames are kept for the follow-up content analysis. Otherwise, the image is considered to be uniform and only the first frame is used for the analysis.

C. Promotional Content Identification

The next step is to identify promotional content in the images. In our research, we manually collected 250 APPIs from Baidu Tieba and found that all of them carry the contacts for the products and services being promoted. So Malèna uses the presence of such content as a necessary condition to filter out non-APPI images. More specifically, promotional content is typically in the form of text (URL, QQ ID etc.) or QR code, which is sought by the promotional content identifier in an image to determine whether it needs to go through the follow-up content analysis. Here the text content is captured using PixelLink [31], a state-of-the-art scene text detection tool. Recognizing QR code, however, is more complicated, as elaborated below.

Finding QR code. As a machine-readable matrix barcode, QR code is supposed to be easily recognizable by bar code readers. However, we found that some spammers (the adversary posting APPIs) apparently do not want their code to be identified by popular scanners (such as ZBar [13], ZXing [14], and BoofCV [3]) and instead only accessible to some specific ones (e.g., the scanner used by WeChat, a popular Chinese social network app). Examples of such QR code are shown in Figure 3. Unlike standard QR codes, these codes have position patterns less conspicuous and in a nonstandard shape (circle) and alignment patterns even less identifiable. In our study, we found that scanner software including ZBar, ZXing and BoofCV could not detect them. However, WeChat can always pick them up.

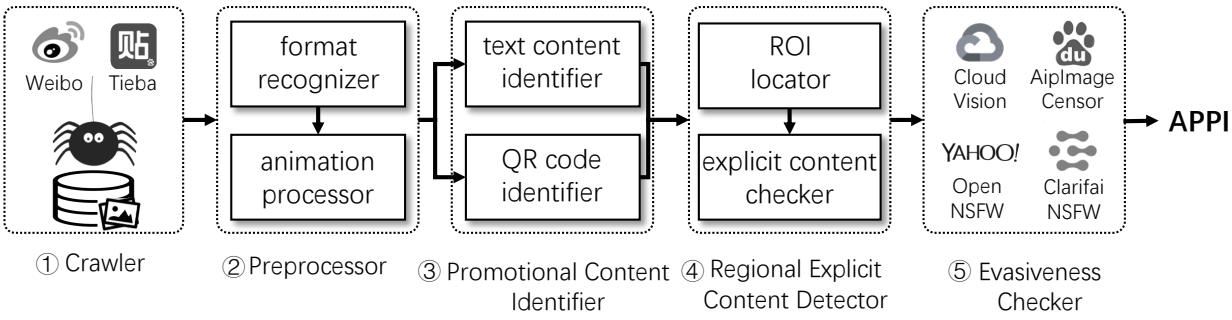


Fig. 2: System architecture.

All we want to do here is just to confirm the presence of these codes. For this purpose, we come up with a simple algorithm, which attempts to capture the three *position patterns* (i.e., three big squares on the three corners) and the *alignment pattern* (i.e., the other smaller one on the fourth corner). Specifically, QR code recognition is based upon separation of the dark and the relatively bright components from the image before the position and the alignment can be found. This can be done using the `threshold` within OpenCV [22], which coverts each pixel on a standard QR code image into one of the two values: 0 if the pixel's grayscale is above a given threshold and 1 if not. For a standard QR code, such a threshold is typically set to 127. For those obfuscated ones, however, we can no longer rely on a single threshold 127, because of the ability of spammers to manipulate the QR code, making (part of) the dark components' grayscale larger than the threshold, and causing false separation (as shown in the upper left figure in Figure 3). Nevertheless, the spammers have to a maintain large enough contrast for the QR code to make feasible the separation of the dark and bright components. Therefore, our approach utilizes multiple thresholds (31, 63, 95, 127, 159, 191, and 223 for our implementation) and for each threshold generates a binary image (with each pixel either 0 or 1). The idea is to analyze all such images to determine whether a QR code is indeed present. To this end, for each image, our approach runs the OpenCV function `findContours` to search for contours (i.e., the boundaries of continuous non-zero pixels). After dropping the contours that are too small or too large, if there is a QR code, we should be able to find three contours under the following constraints: 1) they have similar size and shape (approximately square or circle) and 2) the centers of these contours form an isosceles right triangle. Note that the shape of a contour can be determined by looking at whether the square of the contour's perimeter comes close to 16 (if it is a square) or 4π (if it is a circle) times its area. Finally, we check whether there exists another square-or-circle like contour near the fourth corner to confirm the presence of the QR code.

We found that this algorithm is capable of identifying most unconventional, oriented, and distorted QR codes like those in Figure 3. It was evaluated in our research using 50 images with obfuscated QR codes and 50 images without the code, and found to achieve 100% accuracy.

Fig. 3: Examples of unconventional, oriented, and distorted QR codes.

D. Regional Explicit Content Detection

From the images carrying text or QR codes, Malèna further detects whether they also contain explicit content. As mentioned earlier, even though an APPI attempts to hide such content from detection, it is constrained by the need to preserve the sexual appeal of the image. To exploit this observation, our approach is designed to search for the explicit content from some regions of the image, so as to avoid the interference of the noise introduced to other regions that may cause the whole image to be misclassified. Specifically, Maléna adopts object recognition algorithms [34], [35], [39], [53], [57] to first find a few *regions of interest* where explicit adult content possibly resides, and then runs R-CNN to extract features from each ROI for further detection. Unlike typical region proposal algorithms for object recognition [34], [35], [39], [53], [57], our approach locates the image regions involving humans for finding adult content. Serving this purpose is Mask R-CNN [39], an edge-cutting object recognition framework that applies region proposal networks (RPN) [39], [53] to find ROIs (a bounding box for the object classified as "person") and further

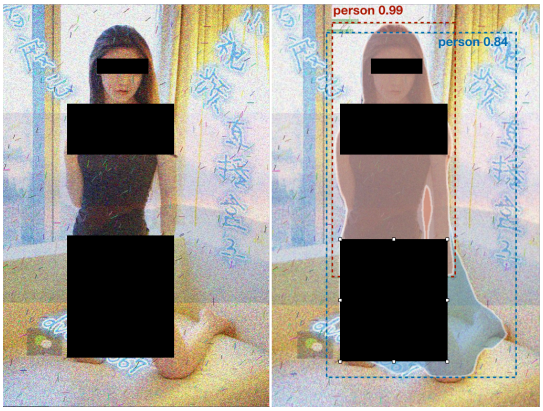


Fig. 4: Mask R-CNN's output on an APPI with ROIs and segmentation masks.

generate their segmentation masks to highlight the objects in these ROIs. An example can be found in Figure 4.

Using the bounding box, our approach crops each ROI from the original image and feeds it together with its segmentation mask to a ResNet-50 [40] model for explicit content detection. Note here that we do not use existing detection models but train a new CNN-based model, since the traditional models may not take advantage of all the information recovered by our R-CNN: for example, our analysis shows that the Yahoo open NSFW does not leverage the segmentation masks, which are critical for locating the target object (i.e., person) and removing noise. The training of the ResNet-50 model and the dataset used for this purpose is elaborated in Section IV-A. Running the trained model on the ROIs, if any of them is found to contain explicit content, we label the input image as a candidate APPI. In this way, we are able to identify the APPIs once their least obfuscated explicit content is found, forcing the adversary to perturb not only some part of the picture but also every region on the picture that contains the adult content to evade detection.

E. Evasion Checker

Finally, to find out whether an identified image is indeed an APPI, which is expected to evade existing detection, we scan the image using four mainstream commercial inappropriate image detectors: Google Cloud Vision API, Baidu AipImageCensor API, Yahoo Open NSFW model, and Clarifai NSFW API. It is flagged as an APPI if it bypasses at least one detector.

IV. Implementation and Evaluation

A. Implementation

Datasets. In our research, we use four datasets for model training and evaluation: the social media image set, the porn picture set, the non-porn picture set, and the groundtruth set.

• *Social media image set*. The social media image set is the dataset from which we want to find APPIs. It includes the images from two major Chinese social media: Baidu Tieba and Sina Weibo. From these two sources, we develop two spiders to collect images.

More specifically, for Baidu Tieba, we gather images from top 63 most active "bars" (a forum on a certain topic). Considering the relatively short lifespan of porn images, our crawler is designed to iteratively visit the 150 latest posts at each "bar" without sleep. This strategy gives our crawler a better chance to discover APPIs once they emerge before they are reported and deleted. In this way, we totally obtain 3,813,888 unique images from 648,621 posts on Tieba from 03/15/2018 to 07/15/2018.

On Sina Weibo, microblog comments are the spammers' favorite channel to distributing APPIs to the large number of blog fans. Therefore, our crawler focuses on 76,763 microblogs with more than 10K fans. For each microblog, we retrieve 100 latest posts, and crawl the images from their comments. In total, we discover 228,810 images on Weibo from 07/01/2018 to 08/25/2018.

• *Porn and non-porn picture sets*. These datasets are used for training the ResNet-50 model (see Section III-D). The non-porn picture set serves as the negative samples, which comes from three sources: (1) 50k images from Microsoft's Celeb-1M [16], (2) 17k images of females with casual wear gathered from the results of querying the Google image search engine using the keyword "female with casual wear", and (3) 20k images of the athletes and 15k images containing single body part or apparels. We use (2) since most females in the Celeb-1M dataset wear very revealing or shiny clothes, which do not provide sufficient information about the regular female clothing. Also, dataset (3) is found to be necessary for differentiating sexual behaviors from sport activities, where athletes also wear tiny clothes and expose large areas of their skin, which are close to porn patterns.

The porn picture dataset, the positive samples for our model, includes 85k images known to contain sexually explicit content. They are scrapped from two pornographic websites (t66y.com and vulvapornpics.com).

• *Groundtruth set*. The groundtruth dataset is used to evaluate our methodology (see Section IV-B), which includes 250 APPIs and 250 non-APPIs randomly sampled from the social media image set. All these images have been manually labeled.

System implementation. We implemented Malèna with 2,400 lines of Python code. Three Python libraries (Pillow [23], skimage [24] and OpenCV [22]) are used for the aforementioned image processing tasks. Also, we build our deep learning models over Tensorflow using two deep neural network architectures and two pre-trained models: a pre-trained PixelLink model (over the VGG16 backbone [21] and the ICDAR2015 dataset [6]) serves as the promotional content identifier; a pre-trained Mask R-CNN model (over the ResNet-50 backbone [25] and the MS COCO dataset [20]) are used for ROI identification; also another ResNet-50 model are trained in our research for detecting explicit content in each ROI.

Model training. As mentioned earlier, our regional explicit content detector includes the ROI locator and the explicit content checker. The ROI locator is a pre-trained Mask R-

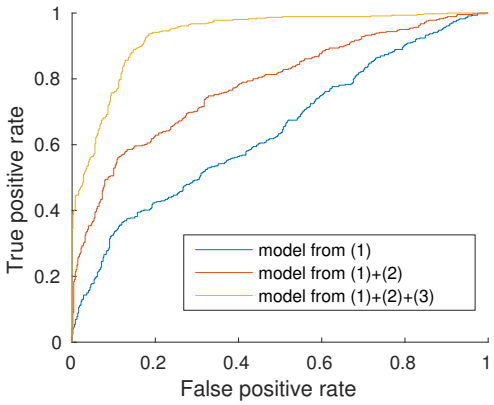


Fig. 5: ROCs of models trained on three levels of picture sets.

CNN model on the MS COCO dataset [20] and the explicit content checker has been trained in our research using the porn/non-porn picture sets (see Section IV-A). To train the checker, we first run the ROI locator on the porn/non-port sets. For each image, the locator reports its ROIs discovered, together with each region's object type, confidence score for the identification, bounding box and segmentation mask for highlighting its object. Then we select all the ROIs whose object types are "person" and confidence scores above a threshold, and crop these regions along their bounding box from their original images. The threshold is chosen empirically through an experiment to identify the "person" objects from 400 random sampled images from the aforementioned social media image dataset. In the implementation of Malèna, the threshold is set to 0.8, as it achieves the best results in the experiment. For each such regions, our approach further applies its segmentation mask to produce a 4-channel image (the standard RGB channels together with the mask channel). All these images are then used as the training inputs for the explicit content checker.

To evaluate the effectiveness of the regional explicit content detector, we use the groundtruth set (see Section IV-A) to build a testing dataset for model evaluation. We also compare models trained on our three levels of porn and non-porn picture sets, (1), (1)+(2) and (1)+(2)+(3). Figure 5 shows their ROCs. We observe that dataset (3) significantly improved the performance. Specifically, the AUC of three models are, 0.64, 0.72 and 0.94 respectively.

B. Evaluation

Precision and recall. To understand the effectiveness of Malèna, we run the system on the groundtruth dataset. Table I shows the precision and recall at each stage of Malèna. Overall, it achieves an overall precision and recall of 91% and 85% respectively on the groundtruth set. Among the 500 images in the set, 233 are reported as APPIs, where 212 are true APPIs, and the other 38 true APPIs are not detected by our system.

TABLE I: Precision and recall at different stages.

stage	precision	recall
promotional content identification	98%	90%
ROI locator	89%	96%
explicit content detection	80%	93%
overall	91%	85%

TABLE II: Running time at different stages.

stage	running time	images per minute
promotional content identification	79.78 min	125.35
regional explicit content detection	140.90 min	70.97
evasiveness checker	163.67 min	61.10
overall	384.35 min	26.02

We investigate the false positives and false negatives. Among the 21 false positives, there are 3 photos of human hands and feet, these body parts are entirely of skin color and there are less other information helping the model to classify them right; 6 of comic books; 4 of athletes and the other 8 of women. As for false negatives, 26 false negatives are because the promotional content identifier failed to detect the text on the image. Two false negatives are due to the misclassification of the ROI locator and the rest are the results of the failure of the explicit content checker.

Performance. To understand the performance of Malèna, we measure the time it takes to process 10,000 images from the social media image set at each individual analysis stage, the promotional content identification, image quality assessment and regional explicit content detection. The experiment is run on a server equipped with a 4-core Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, and a Nvidia GeForce GTX 1070 graphic card with 8 GB memory. We instruct the GPU to process 8 images in parallel.

Table II shows the running time at each stage of of Malèna. Overall, it takes 384.35 minutes to finish processing the 10,000 images. The results provide strong evidence that our system is efficient and can be easily scaled to a desirable level to handle the massive amount of images from online forums every day.

V. Understand Adversarial Images in the Wild

A. Landscape

Running Malèna on the social media image set, our approach automatically detects 4,353 APPIs among the 4,042,698 images collected from 76,752 hot posts/microblogs on Baidu Tieba and Sina Weibo. By comparison, Baidu Tieba hosts more APPIs (3,395 out of 4,353, 78%), while images from Sina Weibo are more likely to contain explicit content (958 out of 228,810 collected from microblogs, 0.419%). Note that both sources prohibit displaying explicit content [2], [10]. Baidu also provides an image censorship API capable of detecting pornographic content.

Overall, 3,080 and 472 Tieba and Weibo accounts are found to post APPIs. Figure 7 presents the distribution of APPI instances over these accounts. We observe that 34% of the Tieba accounts and 16% of the Weibo accounts post more than 5 APPI instances. Meanwhile, 5,060 and 1,103 posts/microblogs are found to be the targets of APPI spammers. Apparently,

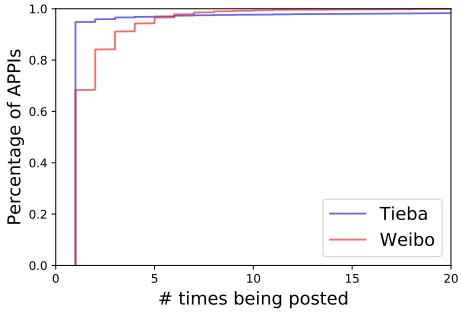


Fig. 6: Distribution of # APPIs per account.

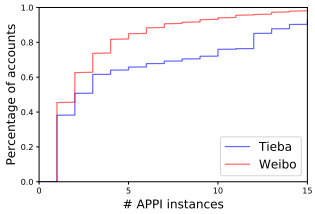


Fig. 7: Distribution of # APPI instances per account.

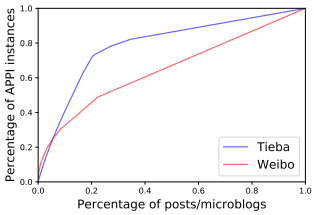


Fig. 8: APPI instance coverage of post/microblog.

their strategy is to select a set of posts/microblogs to post multiple APPIs to each of them, instead of disseminating the images to many different posts. Figure 8 shows the distribution of the number of APPIs per post/microblog. As we can see here, 75% of the Tieba APPIs are discovered under 20% of the posts, while 50% of the Weibo APPIs are associated with 23% of the microblogs. Meanwhile, we find that the spammers continuously post the same APPIs. Figure 6 shows the distribution of duplicated instances per APPI image. 176 (5%) of the Tieba APPIs and 346 (32%) of Weibo APPIs have been found more than once. Particularly, one Tieba APPI has been posted 4,171 times by 251 different users on 298 posts crossing 48 different "bars".

B. Model Evasion

Obfuscation techniques. To understand the obfuscation tricks employed by APPIs, we look into the APPIs detected by our system and categorize their techniques into 7 major categories. Examples of the APPIs in each category are shown in Figure 9.

• *Color manipulation*. The adversary often changes the color of the original image, which is effective since skin-related features are widely utilized in explicit content detection. Color manipulation approaches such as grayscaling, monochromatization, and hue-rotation remove or obfuscate color information of the original image, thereby rendering the skin-color based detection less effective.

• *Rotation*. Rotation involves a linear transformation on the coordinates of each pixel according to a *rotation matrix*. The technique works on the detector that utilizes the features not rotation-invariant. For instance, without data augmentation, many CNN architectures cannot learn rotation invariants.

TABLE III: The usage of 7 obfuscation techniques.

obfuscation technique	# APPI (%)
color manipulation	160 (3.7%)
rotation	1,083 (24.9%)
noising	2,130 (48.9%)
texturing	132 (3.0%)
blurring	829 (19.0%)
occlusion	1,517 (34.8%)
transparentization & overlap	46 (1.0%)

• *Noising*. A common obfuscation trick is adding random perturbations to images. Such perturbations introduce new high-frequency signals to the image, making it harder to recover the high-frequency signals of the original image. High-frequency signals are important in image processing because they contain important structural information (e.g., edges).

• *Texturing*. Texturing is a technique that applies a certain texture (e.g. leather, paper, or marble) to the surface of an image. For example, Figure 9c shows an APPI with a brick-wall texture. Texturing is typically done by overlaying an image with that of a texture material, which often contains rich high-frequency signals. So similar to noising, this technique makes image structural information difficult to extract.

• *Blurring*. Blurring (or smoothing) spreads each pixel's RGB color value to those of nearby pixels, typically by applying a filter to an image. Popular blur filters include mean filter, weighted average filter, and Gaussian filter. Unlike noising and texturing, blurring weakens the high-frequency signals on the original image, by making the transition from one color to the other smoother, which results in less obvious edge features.

• *Occlusion*. A common way to obfuscate explicit content is to apply occlusions to sensitive areas (e.g., posing lip stickers on the breast area as shown in Figure 9g). Occlusion hides the critical area of a pornographic image, which could hide the key features to identify its explicit content, though visually the sexual semantics still gets through to the viewer.

• *Transparentization and overlay*. Transparentization is a process to make an image semi-transparent, which lowers the contrast of the image and therefore weakens its edge features. Semi-transparent images are usually overlaid on top of other images, causing an effect similar to texturing that makes structural information harder to extract.

Table III shows how these obfuscation techniques are deployed across all detected APPIs. Noising appears to be the most common approach. Moreover, we observe that spammers may combine several techniques together: 816 APPIs (19.8%) contain at least 3 different types of obfuscations.

Evading state-of-the-art detection. Our study shows that these APPIs are indeed effective on existing explicit content detection. We run Google Cloud Vision API, Baidu AipImageCensor API, Yahoo Open NSFW model, and Clarifai NSFW API on all the APPIs discovered in our research. It turns out that 35.6% of them *cannot* be detected by *any* of these detectors. Table IV presents their detection rates. Among them, Yahoo Open NSFW model achieves the highest detection rate. Still, more than 2,600 APPIs are missed by this detector.

(a) monochromatization (b) noise (c) texture (d) transparentization (e) rotation (f) blur (g) occlusion

Fig. 9: Examples of APPIs.

TABLE IV: Detection rates of the 4 explicit content detectors.

detector	# detected APPIs	detection rate
Google Cloud Vision API	1,310	30.9%
Baidu AipImageCensor API	1,546	35.5%
Yahoo Open NSFW model	1,728	39.7%
Clarifai NSFW API	1,242	28.5%

TABLE V: Parameters used in 5 types of distortions.

Noise Type	L1	L2	L3	L4
Gaussian noise (var)	0.083	0.107	0.131	0.155
Box blur (ksize)	(3,3)	(5,5)	(7,7)	(9,9)
Transparentization (alpha)	0.8	0.6	0.4	0.2
Rotation (degree)	45	90	145	180
Color manipulation	gray	blue	green	red

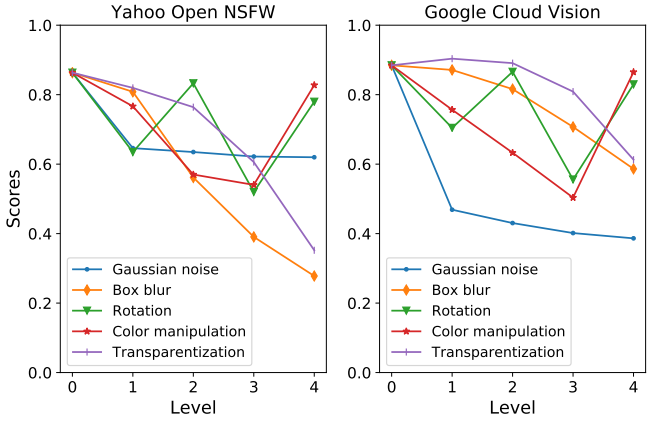


Fig. 10: Explicit content detection results on the distorted images.

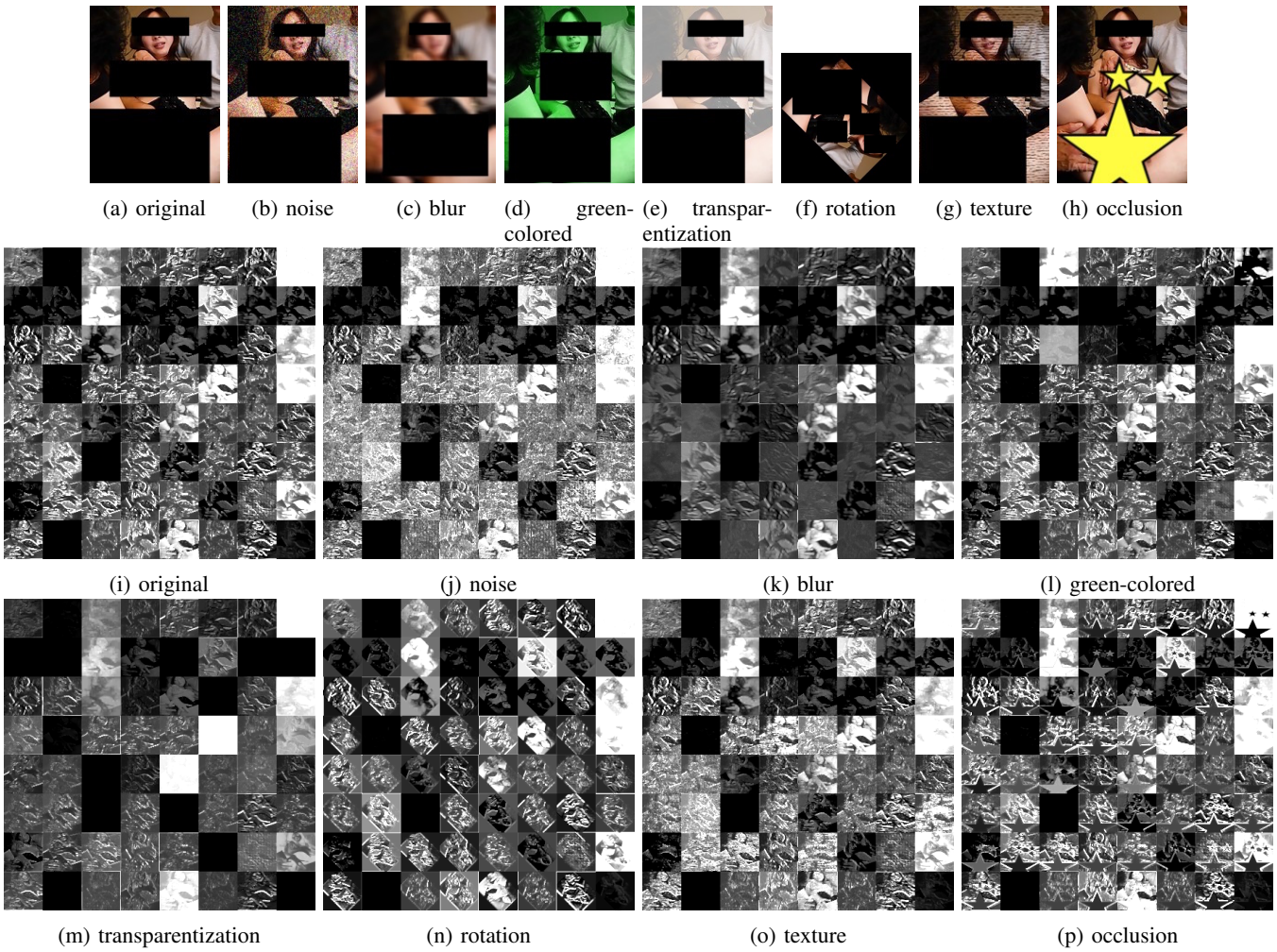
To further understand the effectiveness of the aforementioned obfuscation techniques, we sample 250 (non-adversarial) porn images from the porn picture set (for training Malèna), and apply 5 obfuscation techniques (noising, blurring, transparentization, rotation, and color manipulation) each with 4 different settings (such as angle for rotation, density of noise, etc., see Table V). In this way, we get a total of 5,250 images (including the original images), on which we run Google Cloud Vision API and Yahoo Open NSFW model. The results are presented in Figure 10. As we can see from the figure, with proper settings, these 5 obfuscation techniques are able to effectively degrade the performance of start-of-the-art explicit content detectors. Compared with Yahoo Open NSFW, Google Cloud Vision seems to be more robust against blurring and transparentization (the two processes have similar influence on the image), but more sensitive to noise.

Also we find that the effects of these obfuscation techniques are quite consistent under different settings in both detectors, given their differences. Particularly, the "L" shape curves of noising indicate that the technique is effective when the noise intensity is low but its performance cannot be improved significantly when the intensity increases. Transparentization and blurring, however, perform much better at a higher intensity level than a lower level. Of particular interest are the "W" shape rotation curves: rotating an image by 90 or 180 degree does not make a porn image much harder to detect; while rotating it by 45 or 135 does. This indicates that both Google and Yahoo models seem to use rotation invariant features, but such features only work on right-angle rotations. This could be caused by the absence of other rotated pictures in their training sets. Lastly, color manipulation exhibits a "V" shape curve, indicating that the detectors react differently to different colors: red images are easiest to detect as the color comes closet to the skin color compared with other colors tested, while green ones turn out to be hardest to find. This also implies that both detectors use an image's color features (probably skin color).

Furthermore, we study how the obfuscation techniques impact neural network-based detection systems, i.e., Yahoo Open NSFW model. In particular, we compare the original images' 64 features learned by the first convolutional layer of Yahoo's open NSFW model with those of the images obfuscated by the 7 aforementioned obfuscation techniques. Figure 11 shows the input images and the extracted features (each displayed in a small square), illustrating how each obfuscation technique affects the initial layer of the neural network.

C. Adversarial Examples

As mentioned earlier, recent studies on adversarial learning in image processing focus on finding the adversarial examples that use almost imperceivable perturbations to induce misclassification [37], [49], [55]. To find out whether there is evidence that the real-world adversary indeed utilizes these techniques to seek adversarial examples, we inspect all the APPIs found in our research, looking for high-quality images (with almost imperceivable perturbations) that also circumvent



(a) original (b) noise (c) blur (d) green-colored (e) transparentization (f) rotation (g) texture (h) occlusion

(i) original (j) noise (k) blur (l) green-colored

(m) transparentization (n) rotation (o) texture (p) occlusion

Fig. 11: Features of distorted images extracted by Yahoo NSFW's first convolutional layer (after max pooling).

all explicit content detectors. In the end, we fail to find any such adversarial examples, even when we look at the images without promotional information.

To understand whether the absence of such adversarial examples is actually caused by Maléna's limited capability to find them, we generate 200 adversarial examples for the Yahoo Open NSFW model using the state-of-the-art C&W approach [29], and then pass these examples to our implementation. Maléna successfully detects 196 of them, while Yahoo classified all of them as non-explicit. This indicates that today's cybercriminals likely still rely on a set of predetermined obfuscation techniques (see Section V-B) to generate APPIs, not gradient descent.

VI. Promotional Explicit Content Campaign

The discovery of APPIs and their promotional information and distribution channels enables us to investigate the ecosystem of such illicit promotions. In our research, we look into the content the APPI spammer promotes, the correlation among different APPIs and the way such images are disseminated. Also, we analyze a large APPI campaign as a case study.

A. Promotional Content Analysis

As mentioned earlier, we find that APPIs carry two types of promotional content: text and QR codes. To extract such promotional information, we leverage Google Cloud Vision API's OCR function to extract text for further manual validation. For all the QR codes detected by our preprocessor (Section III-C), we first attempt to use ZBar [13] to automatically decode each of them. If ZBar fails, we then manually scan it using WeChat, which almost always works on these codes, even in the presence of some obfuscation. In this way, 612 unique promotional content pieces are discovered in the form of URLs, QQ and WeChat (popular instant message apps) IDs, Weibo IDs, QR codes etc. Table VI shows the number of promotional content pieces in each type. We observe that QR code is the most prevalent one (1,430 out of 3,432, 41.7%), because it is convenient for the target viewers to extract the promotion information directly from images using their Wechat apps.

Interestingly, in addition to the contact information for illicit products, text items in APPIs sometimes include trending Internet buzzwords. For example, we find "skr" (a popular

TABLE VI: Statistics of promotional content.

Type	Weibo	Weibo (unique)	Tieba	Tieba (unique)
QQ ID	17	7	186	69
Weibo ID	375	261	8	5
WeChat ID	239	110	1092	135
QR code	0	0	1430	45
URL	0	0	85	31

TABLE VII: Examples of sensitive text replacement.

Examples	Type	Meaning	Num
v♥	emoji	WeChat	12
“刊片”	homophonic	porn movie	10
“企鹅”	jargon	QQ	18
“呦呦”	jargon	child porn	8
vx	homophonic+initial	WeChat	39

Internet buzzword trending in the late July) was used in 13 APPIs posted on Weibo during that particular time period, even though the meaning of the word is totally irrelevant to the products being promoted (adult videos). Apparently, APPI spammers try to leverage such eye-catchy words to draw attention from their potential buyers.

Evasive techniques on promotional content. In addition to the obfuscation techniques applied to the explicit content in APPIs (Section V-B), cybercriminals also use other approaches to protect their promotional content from detection. In particular, we observe that special text styles are used (e.g., semi-transparent text, hollow text, or even handwritten texts) to prevent the texts from being recognized by OCR tools. Also, we find that some keywords (such as the name of instant message app) in the promotional content are replaced with the characters of similar meanings, similar shapes or jargons. Specifically, in our dataset, more than 200 APPIs are found to include such keyword replacements, using jargons, homophonic words, romanization of Chinese words, etc. For example, as shown in Table VII, the sensitive word “QQ” (an instant message app) is often replaced with a jargon “企鹅” (penguin) because the app uses a penguin as its icon, and “微信” (WeChat) is replaced with an English letter “v” and an emoji “♥”, where “v” is homophonic to “微”, and “♥” is “心” in Chinese, which makes them sound like WeChat in Chinese.

Cybercriminals also protect QR code on APPIs. As mentioned in Section III-C, unconventional, disoriented, and distorted QR codes are observed in APPIs (see Figure 3). Although standard QR codes are designed as machine-readable, most open-source QR code scanners, such as ZBar, ZXing, and BoofCV, are incapable of recognizing or even detecting the QR codes after the adversarial processing, while they are still recognizable by the professional QR Code Scanner in WeChat. Among 1,430 QR codes detected from the APPIs, 595 (41.6%) evade ZBar but are still readable by WeChat.

Promoted products. It is not surprising that the overwhelming majority of APPIs are used to promote sexual products, because such products are in line with the explicit content displayed by the images. In addition, APPIs that promote gamble websites, drugs, and virtual merchandise in video games are also found in our dataset. We even observe one APPI with two different types of promotional content (a sexual product and a game app).

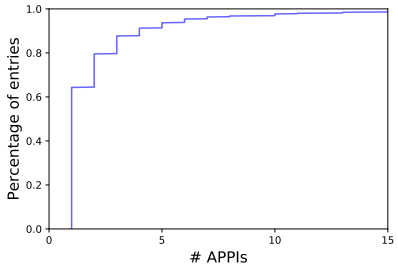


Fig. 12: Shared promotional content.

Among the sex-related products, porn videos, comics, and porn websites are most prevalent. We even observe child porn and bestiality porn being promoted via APPIs at Baidu Tieba. Besides, sexual apps are common products advertised by APPIs. Such illicit apps are the platforms for both online and offline transactional sex, such as live sex webcam and one-night stand.

B. Campaign Discovery

To reveal the criminal campaigns behind the APPIs, we study the correlation among different APPIs from two perspectives: shared promotional content and explicit content reuse.

Shared promotional content. We observe the prevalence of shared promotional content in APPIs. Figure 12 illustrates the distribution of promotional content volume per APPI. As we can see here, 113 out of 285 (39.6%) promotional content pieces appear in more than one APPI from Tieba, and 119 out of 378 (31.5%) from Weibo. The most prevalent one is a QR code leading to a porn app download link (http://i8cv.com/ /index.php/S/eOs5EVtc), which is embedded in 669 different APPIs.

Explicit content reuse. Also interestingly, it is very common for cybercriminals to reuse the same set of explicit images to craft APPIs. To find reused explicit content, we first locate the explicit content in APPIs and then compare the similarity among those explicit content pieces. Specifically, we again leverage Maléna to locate the ROIs with explicit content (see Section III), and perform object matching using the SIFT algorithm [46] on each ROI pair. In this way, we find that 3,981 out of 4,353 APPIs share explicit content with at least one other image.

APPI campaign discovery. With the information for promotional and explicit content reuse, we are able to recover APPI campaigns, which share explicit content or have same promotional information. The APPI campaigns can be discovered using a graph algorithm, where each image is a node in the graph, and for each pair of images sharing promotional or explicit content, we connect the corresponding nodes with an undirected edge. The APPI campaigns can be recognized by finding connected components in the graph. In this way, we recover 19 APPI campaigns including more than 10 APPIs.

TABLE VIII: Top 5 APPI campaigns.

Campaign	# APPIs	Source
1	1,325	Tieba
2	786	Tieba
3	347	Weibo
4	39	Weibo&Tieba
5	25	Tieba

Table VIII shows the top 5 APPI campaigns, where the largest one consists of 1,325 APPIs. We elaborate this campaign in Section VI-D.

C. Distribution Channels

From the 4,353 APPIs we discovered, 3,080 Tieba accounts, and 472 Weibo accounts are identified to distribute at least one APPI. Among the 3,080 accounts on Baidu Tieba, 2,748 were banned or deleted by Baidu by Aug 30 2018, while most of the Weibo accounts (399 out of 472) were still alive then.

To investigate whether those APPI distribution accounts are dedicated APPI distribution accounts or legitimate ones but compromised by the cybercriminals to post APPIs, we use a set of criteria for filtering and manual validation. Specifically, for the live accounts, we crawl their profiles and social relations. Then, for Baidu Tieba accounts, we utilize two criteria: the average document frequency of each character in the username and the number of the subscribed bars. This is because dedicated accounts often use auto-generated usernames consisting of uncommon words rarely appearing in Chinese documents such as Chinese Wiki [19]. Also, normal users usually subscribe several bars. In this way, 82 out of 332 live Tieba accounts are found to be dedicated APPI distribution accounts, and we manually investigate and validate 166 as compromised accounts. Similarly, we identify 211 compromised Weibo accounts. We notice that the activities of compromised Weibo accounts are only to comment on hot microblog using APPIs. They did not actively post any microblogs themselves for months or even years. On the other hand, for the dedicated accounts, they constantly post microblogs (without illicit content but sometimes meaningless sentences) at a very high frequency (more than 30 posts per day). We suspect those dedicated promotional accounts are maintained by bots.

D. Case Study

In Section VI-B, we discover a huge APPI spamming campaign containing 1,325 APPIs on Baidu Tieba. The campaign was active from 04/16/2018 to 07/15/2018, covering 1,515 posts across 50 bars, involving 1,314 accounts. All promotional content pieces in APPIs are QR codes, from which we extract 19 URLs over 8 different domains. They were used to redirect visitors to download at least 7 mobile sexual apps.

We observe that the cybercriminals heavily reuse the explicit content: 1,238 APPIs in the campaign are variants of the same image but protected with a variety of obfuscation techniques including noising, blurring, occlusion, transparentization and color manipulation (grayscaling).

After tracking the QR codes in APPIs, we observe that six QR codes demonstrate interesting redirection behavior: all of them can be decoded as URLs under t.cn, the domain of Sina's URL shortener service. Such shortened URLs lead to a redirector controlled by SoHu (https://passport.zhan.sohu.com/passport/sohu/login-jumpto?callback={redirected_url}), which then redirects the visitors to 4 different landing domains under ".top" top-level domain: dannh.top, 000internet17.top, 000cangzhouu.top, and sj87.top

Also, three URLs are under iamh5.cn, an online HTML5 web app developing and hosting platform, and three URLs under i8cv.com, a website providing alpha test service for mobile apps. Unfortunately, all of the 6 apps were removed by the time we studied the campaign. However, another URL in this campaign, http://bilibilibilibilibili.cn/1 leads to a website that was still alive. The website was used as a doorway redirecting visitors to http://cl.lgubn.cn, which hosted the promoted app, "COLOR直播", or "Sexual Streaming" in English. As the app name suggests, it is a video streaming platform that focuses on sexual content. Interestingly, we find out that the images used to craft APPIs in this campaign are actually the screenshots of the app, and hence we suspect that the rest URLs in this campaign are used to promote the same sexual apps of different versions and platforms.

VII. Discussion

Chinese social media. In our study, we investigate APPIs on two Chinese social media platforms: Baidu Tieba and Sina Weibo. We should acknowledge that the relatively limited vantage points may limit the findings made by our research.

Specifically, our data source is limited to Chinese social media, so our research is insufficient to confirm whether the risk of APPI is a regional or global problem. To the best of our knowledge, adversarial sexual images for the promotional purpose are more prevalent in Chinese social media than English ones. The reason is that China enforces a more strict content censorship policy, where explicit content is strictly prohibited in the Chinese Internet [4]. Therefore, Chinese social media service providers regularly clean up explicit content, which motivates the cybercriminals to aggressively apply obfuscation techniques to protect their promotional porn images. Meanwhile, we perform a relatively small-scale study on Twitter and found 6 instances of APPIs from 80k images among hot tweets. This finding indicates that APPIs exist, although not that prevalent, in English social media.

Method limitation. Despite our system's success in finding more than 4,000 images with obfuscation, we admit that although it is nontrivial, Malèna is possible to be evaded if the adversary has the knowledge of the system design or is able to access its model query API. One way to evade Malèna is to fully obfuscate the promotional content or explicit content in the image. However, it could make the audience less interested in the image. Alternatively, it is possible to craft adversarial examples against the neural network models adopted in Malèna. For example, one could attack the Mask

R-CNN model to cause the misclassification of the "person" object on the explicit image, and Malèna would reject the image as no ROI. Although it is not a robust detection system, Malèna is still effective as a measurement tool to help us better understand the real-world APPIs today.

Considering the performance of Malèna, our implementation on the test environment with a single Nvidia GeForce GTX 1070 graphic card of 8 GB memory reports the processing speed of 26.02 images per minutes, or, equivalently, a 2.3-second processing time for each image (Section IV-B). Such performance is not enough for real time tasks, as the user may perceive noticeable delays. However, most Malèna's time-consuming steps are deep learning jobs which greatly favor parallelism and can be accelerated by introducing more graphic cards with larger memory to make it feasible in a production environment.

Mitigation. Our study reveals the problem of the distribution of real-world adversarial images with an underground business behind promoting illicit products. We suggest all services that allow user-generated content to adopt a dedicated detector for real-world adversarial images such as Malèna in their content security pipeline. Detecting such adversarial images is still challenge because the illicit content on such images is typically obfuscated and thus is hard to identify. However, our measurement study shows that the adversary heavily relies on the promotional content on the images to advertise their products. So the semantic-aware image processing approaches targeting such promotional content would be helpful in the mitigation of the problem of real-world adversarial images. Further, the key to mitigate such a problem is the takedown of underground business to generate such adversarial images. In our future work, we will systematically explore underground business and key actors enabling those images.

Adversarial example defenses. Previous researchers have achieved limited success in defense against adversarial examples [50], [54]. Based on this, we believe previous defensive approaches will be pale or even worse when handling APPIs, a harder problem than defeating adversarial examples. This is because APPIs are distorted further away from the original images than adversarial examples. From the neural network's perspective, classifying two such far way images into the same category will not provide any bonus for better performance on the usual training set, and there are no rules encouraging the neural network to do that: even in the min-max defense [54] trying to increase the minimal distance between two categories, let alone the distillation defense [50] that hide the gradients. As for our approach, Malèna, its success is ascribed to using the mask (provided by Mask R-CNN, see Section III), which lowers the weights in occlusion areas or no-body areas, and therefore neutralizes some parts of interference introduced by the adversary.

In the meanwhile, we still have to acknowledge that our protection could be evaded by carefully designed and targeted attacks from the adversary who has the full knowledge of our system design and parameters. However the appliance of Malèna would raise in the bar of such attacks, making them more costly especially to the adversary who want to launch black-box attacks. This is because the classification results of our *explicit region proposal network*, which are used only internally in the following Malèna detection pipeline and hence are transparent to the adversaries, can not be easily inferred. Specifically, to attack Malèna, the adversary can either add perturbations to attack the ROI locator or regional explicit content detector (see Figure 2). The attack to the ROI locator (i.e., explicit region proposal network) is non-trivial due to the missing classification results: those classification parameters of our proposal network are orthogonal from the gradient propagation process given the final detection output. Meanwhile, attacking regional explicit content detector faces two challenges: first, our proposal network restricts the available region where adversaries can add the perturbations; second, the mutual effects between the mask and the input image will significantly hinder the searching process of adversarial perturbations. While adversaries can justly find perturbations simultaneously to bypass our proposal network and detection network, it, as we expected, would be much harder than evading normal networks.

Ethical issue. Before we started this research, we consulted with IRB and confirmed that this research would not require the approval of IRB because all of the images that we reviewed were pre-existing (collected from the Internet). Therefore only a secondary analysis of already published materials was involved, and thus did not constitute human subjects research. In addition, in this paper, we attach a few APPIs to help the readers better understand our paper and the problem we study. We have applied masks to cover most exposed skin areas and actors' eyes in all the attached explicit images. We are not intended to distribute any explicit content and leak actors' privacy.

VIII. Related Work

Explicit content detection. Numerous studies have looked into the detection of nudity or pornography in color images or videos. The two traditional elements for explicit content detection are skin detection and text detection. Platzer et al. [52] proposed an explicit image detection algorithm which accurately detects skin and skin position using a collection of shapes. geometric rules. Chan et al. [30] proposed a pornographic website based on skin-derived features and text analysis. Lopes et al. [45] investigated a nudity detection in videos based on a bag-of-visual-features representation for frames. Recent years, deep learning techniques have been used in explicit content detection. Wehrmann et al. [58] comprised both convolutional neural networks and LSTM recurrent network for adult content detection in videos. Perez et al. [51] classified pornographic videos using convolutional neural networks along with static and motion information. In contrast to previous works, which all detect plain explicit contents, we proposed a unique technique to detect adversarial explicit contents with multiple evasive techniques (such as blur, occlusion) applied.

Adversarial image detection. Recent years have seen rapid growth in the area of adversarial example detection. Previous works on adversarial example detection mainly fall into three categories [28]: the first category of detection scheme is secondary classification based detection, which builds a second classifier to detect adversarial example. Grosse et al. [38] propose a variant on adversarial re-training, which introduces a new class solely for adversarial examples. Gong et al. [36] construct a binary classifier to learn to partition the natural images from adversarial examples. The second category of detection scheme is principal component analysis (PCA) detection, which transforms points from high dimensional space to low dimensional space. Hendrycks et al. [41] observe adversarial examples placing a higher weight on the larger principal components than natural images. Li et al. [43] apply PCA to the values after inner convolutional layers of the neural network, and use a cascade classifier to detect adversarial examples. Another category of detection schemes detect adversarial examples by comparing the distribution of natural image to the distribution of adversarial example. As an example of this category, Feinman et al. [33] investigate model confidence on adversarial samples by looking at Bayesian uncertainty estimates and other features. Different from previous works, our paper proposed a technique to detect the real-world adversarial explicit content generated by the attackers and used for cybercrime.

Image processing for security. Recent years, image processing technique has been actively used for security and privacy research. Borgolte et al. introduced Meerkat [27], a computer vision approach to website defacement detection. The technique is capable of identifying malicious content changes from screenshots of the website. Medvet et al. [47] propose a system to detect a potential phishing page leveraging features such as parts of the visible text, the images embedded in the website, and the overall appearance of the website as rendered by the browser for detection. Anderson et al. [26] introduce image shingling, a technique similar to w-shingling, to cluster screenshots of scams into campaigns. Nappa et al. [48] leverage perceptual hashing to group visually similar icons of malicious executables under the assumption that a similar icon suggests that the two executables are part of the same malware distribution campaign. Templeman et al. [56] introduced a technique for owners of first-person cameras to 'blacklist' sensitive spaces (like bathrooms and bedrooms), which performs novel image analysis to classify where a photo was taken. Zannettou et al. [61] develop a processing pipeline based on image processing technique to detect and to track memes across multiple Web communities To the best of our knowledge, no prior work applies image-based methods to detect promotional adversarial explicit contents.

IX. Conclusion

In this paper, we report our study on adversarial promotional porn images, which promote the illicit business (e.g., porn app or gambling site) using adversarial porn image aiming at evading explicit content detectors. To capture such stealthy image, our advanced explicit content detector, Maléna, utilizes a set of DNN based techniques to automatically identify the promotional information and capture the relatively less obfuscated region in the image. Our study shows that Maléna achieves a low false detection rate (about 9%) with 85% coverage. Running on 4,042,698 images from 725,384 hottest posts/microblog across two social media platforms Baidu Tieba and Sina Weibo, Maléna automatically detects 4,353 APPIs, which brings to light the real-world image obfuscation techniques used by the cybercriminals to evade state-of-art explicit content detector (e.g., Google Cloud Vision API and Yahoo Open NSFW model). Such obfuscation techniques include adding high-frequency signals (e.g., texturing and noising) or filter effects (e.g., blurring) to an image. Our research further demonstrates the effectiveness of such obfuscation techniques and the bar our technique raises for the attacks. Moving forward, our study reveals the ecosystem of such illicit promotion from the distribution channel, APPI campaigns to the promoted illicit businesses. It helps to get a more comprehensive view of the illicit promotion and develop effective solutions to mitigate such security risks.

X. Acknowledgements

We are grateful to our shepherd Gianluca Stringhini and the anonymous reviewers for their insightful comments. We thank Xiaoran Peng, David Crandall and Dmitry Evtyushkin for their valuable feedback. This work is supported in part by NSF CNS-1527141, 1618493, 1801432, 1838083, 1801365 and ARO W911NF1610127.

References

[1] "Baidu tieba," https://tieba.baidu.com/.
[2] "Baidu tieba policy," http://static.tieba.baidu.com/tb/eula.html.
[3] "Boofcv," https://boofcv.org/index.php?title=Main_Page.
[4] "China cybersecurity law," http://www.cac.gov.cn/2016-11/07/c_1119867116.htm/.
[5] "Clarifai," https://clarifai.com/.
[6] "Downloads - incidental scene text," http://rrc.cvc.uab.es/?ch=4&com=downloads.
[7] "Google cloud vision api," https://cloud.google.com/vision/.
[8] "Safesearch - wikipedia," https://en.wikipedia.org/wiki/SafeSearch.
[9] "Sina weibo," https://www.weibo.com/.
[10] "Sina weibo service usage agreement," https://www.weibo.com/signup/v5/protocol.
[11] "Twitter media policy," https://help.twitter.com/en/rules-and-policies/media-policy/.
[12] "phash: The open source perceptual hash library," http://www.phash.org, 2008.
[13] "Zbar bar code reader," http://zbar.sourceforge.net, 2011.
[14] "Zxing ("zebra crossing") barcode scanning library for java, android," https://github.com/zxing/zxing/, 2011.
[15] "Libmagic," https://github.com/threatstack/libmagic, 2014.
[16] "Ms-celeb-1m: Challenge of recognizing one million celebrities in the real world," https://www.microsoft.com/en-us/research/project/ms-celeb-1m-challenge-recognizing-one-million-celebrities-real-world/, 2016.
[17] "Tensorflow implementation of yahoo's open nsfw model," https://github.com/mdietrichstein/tensorflow-open_nsfw, 2017.
[18] "Baidu aipimagecensor github," https://github.com/Baidu-AIP/php-sdk/blob/master/AipImageCensor.php, 2018.
[19] "维基百科," https://zh.wikipedia.org, 2018.
[20] "Coco - common objects in context," http://cocodataset.org/, 2018.

[21] "Implementation of our paper 'pixellink: Detecting scene text via instance segmentation' in aaai2018," https://github.com/ZJULearning/pixel_link, 2018.

[22] "Opencv: Opencv modules," https://docs.opencv.org/3.4/index.html, 2018.

[23] "Pillow," https://pillow.readthedocs.io/en/latest/, 2018.

[24] "Scikit-image," https://scikit-image.org, 2018.

[25] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[26] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker, "Spamscatter: Characterizing internet scam hosting infrastructure," Ph.D. dissertation, University of California, San Diego, 2007.

[27] K. Borgolte, C. Kruegel, and G. Vigna, "Meerkat: Detecting website defacements through image-based object recognition."

[28] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 3–14.

[29] ——, "Towards evaluating the robustness of neural networks," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 39–57.

[30] Y. Chan, R. Harvey, and D. Smith, "Building systems to block pornography."

[31] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," *arXiv preprint arXiv:1801.01315*, 2018.

[32] R. Di Pietro and L. V. Mancini, *Intrusion detection systems*. Springer Science & Business Media, 2008, vol. 38.

[33] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[34] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[36] Z. Gong, W. Wang, and W.-S. Ku, "Adversarial and clean data are not twins," *arXiv preprint arXiv:1704.04960*, 2017.

[37] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *stat*, vol. 1050, p. 20, 2015.

[38] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[41] D. Hendrycks and K. Gimpel, "Early methods for detecting adversarial images," *arXiv preprint arXiv:1608.00530*, 2016.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[43] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics." in *ICCV*, 2017, pp. 5775–5783.

[44] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[45] A. P. B. Lopes, S. E. de 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 *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on*. IEEE, 2009, pp. 224–231.

[46] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[47] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*. ACM, 2008, p. 22.

[48] A. Nappa, M. Z. Rafique, and J. Caballero, "Driving in the cloud: An analysis of drive-by download operations and abuse reporting," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2013, pp. 1–20.

[49] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 372–387.

[50] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 582–597.

[51] 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.

[52] 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*. ACM, 2014, pp. 45–56.

[53] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[54] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.06605*, 2018.

[55] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[56] R. Templeman, M. Korayem, D. J. Crandall, and A. Kapadia, "Placeavoider: Steering first-person cameras away from sensitive spaces." 2014.

[57] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[58] 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.

[59] G. L. Wittel and S. F. Wu, "On attacking statistical spam filters." in *CEAS*, 2004.

[60] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?" in *International Conference on Machine Learning*, 2015, pp. 1689–1698.

[61] S. Zannettou, T. Caulfield, J. Blackburn, E. D. Cristofaro, M. Sirivianos, G. Stringhini, and G. Suarez-Tangil, "On the origins of memes by means of fringe web communities," *CoRR*, vol. abs/1805.12512, 2018. [Online]. Available: http://arxiv.org/abs/1805.12512