Adversarial Examples versus Cloud-based Detectors: A Black-box Empirical Study

Xurong Li, Shouling Ji, Meng Han, Juntao Ji, Zhenyu Ren, Yushan Liu, and Chunming Wu

Abstract—Deep learning has been broadly leveraged by major cloud providers such as Google, AWS, Baidu, to offer various computer vision related services including image auto-classification, object identification and illegal image detection. While many recent works demonstrated that deep learning classification models are vulnerable to adversarial examples, real-world cloud-based image detection services are more complex than classification and there is little literature about adversarial example attacks on detection services. In this paper, we mainly focus on studying the security of real-world cloud-based image detectors. Specifically, (1) based on effective semantic segmentation, we propose four different attacks to generate semantics-aware adversarial examples via only interacting with black-box APIs; and (2) we make the first attempt to conduct an extensive empirical study of black-box attacks against real-world cloud-based image detectors. Through evaluations on five popular cloud platforms including AWS, Azure, Google Cloud, Baidu Cloud and Alibaba Cloud, we demonstrate that our IP attack has a success rate of approximately 100%, and semantic segmentation based attacks (e.g., SP, SBLS, SBB) have a a success rate over 90% among different detection services, such as violence, politician and pornography detection. We discuss the possible defenses to address these security challenges in cloud-based detectors.

Index Terms—Cloud Vision API, Cloud-based Image Detection Service, Deep Learning, Adversarial Examples.

1 INTRODUCTION

Taking advantage of the availability of big data and the strong learning ability of neural networks, deep learning outperforms many other traditional approaches in various computer vision tasks such as image classification, object detection, and image segmentation. Since deep learning often requires massive training data and lengthy training time, many cloud service providers such as Google, AWS, Baidu, Alibaba, Azure, offer deep learning Applicant Program Interfaces (APIs) to clients such that clients can accomplish their computer vision tasks without the need to train their own models. These APIs can help the cloud service users check images for both commercial and non-commercial purposes. For example, the search engine giant Google\(^1\), and Baidu\(^2\) allow their APIs to identify the category of pictures (e.g., dog, cat); Alibaba Cloud\(^3\) and Azure\(^4\) provide APIs to check whether the images are illegal (e.g., pornographic, violent).

However, deep learning has been recently found extremely vulnerable to adversarial examples, which are carefully-constructed input samples that can trick the learning model into producing incorrect results. Hence, the study of adversarial examples in deep learning has drawn increasing attention among the security community\(^5\). In general, in terms of applications, research of adversarial example attacks against cloud vision services can be grouped into three main categories: self-trained classifier attacks, cloud-based classifier attacks, and cloud-based detector attacks, as shown in Table\(^6\). For self-trained classifiers, clients upload the training data themselves and attackers know the distribution of training data in advance\(^2\),\(^3\),\(^4\). For cloud-based classifiers, the cloud providers train the classifiers themselves (e.g., image classifiers on AWS), attackers do not have the prior knowledge and state-of-art attacks are achieved by hundreds of thousands of queries to successfully generate an adversarial example\(^5\),\(^6\). Different from classifiers, cloud-based detectors identify the bounding area in an input image, and then generate a label for the identified area. Despite a “classifier” is incorporated in the last step, a detector has to contain other modules, such as object detection, image segmentation, and even human judgment in some complicated situations. Attacking cloud-based image detectors is a challenging task since it is hard to bypass these complicated techniques simultaneously to launch a successful attack with limited queries.

Unfortunately, although cloud-based detectors are playing an increasingly important role, there is very limited work exploring the possibility of adversarial example attacks against the detection of cloud vision services. A few of the

TABLE 1

| Vision Model Service | Attack Method |
|----------------------|--------------|
| self-trained classifier | Substitution model\(^2\), MLaaS\(^3\), ZOO\(^4\) |
| cloud-based classifier | Boundary attack\(^2\), Ensemble models\(^5\) |
| cloud-based detector | Work in this paper |

1. https://cloud.google.com
2. https://ai.baidu.com
3. https://www.alibabacloud.com
4. https://azure.microsoft.com
most recent work that paid attention to fool image detectors focused on the standard object detection algorithm but cannot be readily applied to the cloud environment due to the complicity described above. In [38], Florian et al. succeed to steal a machine learning model via public APIs by an equation-solving method. However, it is impractical to attack a commercial model with hundreds of millions of parameters by a simple equation-solving method. Actually, [38] succeeds to steal a simple model trained by themselves and it belongs to self-trained classifiers attack.

To fill this emerging gap, in this work, we take the first step to present attacks on cloud-based detectors. In order to conduct a comprehensive study, we consider the image detector services on five popular cloud platforms worldwide including Baidu Cloud, Google Cloud, Alibaba Cloud, AWS and Azure. In the rest of this paper, we use “Google”, “Alibaba”, and “Baidu” for short, to present the corresponding cloud services provided by these companies.

In this study, by incorporating the image semantics segmentation, we propose four black-box attack methods on cloud-based detectors, which do not need prior knowledge of detectors and more importantly, can be achieved within very limited queries. Specifically, we present Image Processing (IP) attack, Single-Pixel (SP) attack, Subject-based Local-Search (SBLS) attack and Subject-based Boundary (SBB) attack. Our empirical study demonstrates that the proposed attacks can successfully fool the cloud-based detectors deployed on the major cloud platforms with a remarkable bypass rate even approaching 100% as shown in Table 2. We summarize our main contributions as follows:

- To the best of our knowledge, this is the first work to study the black-box attacks on cloud-based detectors without any access to the training data, model, or any other prior knowledge. Different from attacks on a classifier, we investigate the components of detectors and design four kinds of methods to fool the cloud-based detectors.
- We propose four attack methods by incorporating semantic segmentation to achieve a high bypass rate with a very limited number of queries. Instead of millions of queries in previous studies, our methods find the adversarial examples using only a few thousands of queries.
- We conduct extensive evaluations on the major cloud platforms worldwide. The experimental results demonstrate that all major cloud-based detectors are all bypassed successfully by one or multiple methods of our attacks. All the tests only rely on the APIs of cloud service providers. The results also verify the feasibility of our proposal.
- We discuss the potential defense solutions and the security issues. By revealing these vulnerabilities, we provide a valuable reference for academia and industry for developing an effective defense against these attacks. We reported the vulnerabilities to the involved cloud platforms and received very active positive acknowledgements from them.

Roadmap In the rest of the paper, we begin with the preliminary in Section II, followed by threat model and criterion in Section III. Section IV describes the details of our attack algorithms. Section V shows experimental results on the cloud-based detectors. The Effects of these attacks and potential defense methods are discussed in Section VI. Section VII summaries the related work of our paper. Finally, Section VIII concludes this paper and proposes further work.

2 PRELIMINARY

2.1 Neural Network and Adversarial Examples

A neural network is a function \( F(X) = Y \) that accepts \( X \in \mathbb{R}^n \) and outputs \( Y \in \mathbb{R}^m \), where the model is a \( m \)-class classifier, \( \mathbb{R} \) is the set of real numbers, \( n \) is the dimension of \( X \), and \( F \) is the combination of model parameters \( \theta \). In this paper, the parameters of models are unknown. The output \( Y \) is a \( m \)-dimensional vector \( y_i \) \((y_i)\) is the probability of each class), where \( 0 \leq y_i \leq 1 \), for \( 1 \leq i \leq m \) and \( \sum_{i=1}^{m} y_i = 1 \). We show the architecture of a Deep Neural Networks (DNN) model in Figure 1. The final label \( L(X) = \arg\max_i (y_i) \). Sometimes in response to a query, cloud models only return a confidence score instead of a probability distribution. Note that there is no correlation between scores in different classes. For instance, Alibaba Cloud only returns a confidence score from 0 to 100 when queried. Probability can leak more information than scores due to the strong correlation between the probabilities and the classes. Our algorithms could be adapted well in either of the cases (probabilities or scores).

![Fig. 1. A neural network example.](image-url)
Adversarial example attacks on neural networks were first proposed by Szegedy [8], wherein well-designed input samples called adversarial examples are constructed to fool the learning model. Specifically, adversarial examples are generated from benign samples by adding small perturbation that is imperceptible to human eyes, i.e., \( X^* = X + \delta_X \) and \( F(X^*) = Y^* \), where \( X^* \) is an adversarial example, \( \delta_X \) is a perturbation, and \( Y^* \) is adversarial label. Prior work [9], [10] has shown that adversarial examples are detrimental to many systems in the real-world. For instance, under adversarial example attacks, the automatic driving system may take a stop sign as an acceleration sign [9] and malware can evade the detection systems [10]. Depending on whether there is a specified target for misclassification, adversarial example attacks can be categorized into two types, i.e., targeted and untargeted attacks. Since we only intend to make the API service generate the incorrect label, in this paper, we focus on the untargeted attacks, which actually are the superset of targeted attacks due to more restricted access to the model.

### 2.2 Cloud Vision APIs Based on Neural Network

Due to the high cost of storing massive data and intensive computational resource usage of training a neural network for computer vision, it is common for individuals and small business to use cloud platforms to train and perform deep learning tasks in recent years. The cloud service providers normally own plenty of data and computing power, and they actively provide their users with multiple APIs of pre-trained neural networks as one of their services. By leveraging these cloud-based classification and detection services, a small fee could allow an application to complete a relatively complicated computer vision task. We list the computer vision services provided by several major cloud service providers in the market along with the fees they charge for 1000 queries in Table 3.

| Cloud Platform | Classifier | Detector | Fee/Query   |
|----------------|------------|----------|-------------|
| Baidu          | Y          | Y        | 1.5$/1000   |
| Google         | Y          | Y        | 1.5$/1000   |
| Alibaba        | Y          | Y        | 0.27$/1000  |
| Azure          | Y          | Y        | 1$/1000     |
| AWS            | Y          | Y        | 1$/1000     |

Both Classification and detection modules are provided by these computer vision APIs. The classification module accepts an image and produces a label, which has many applications such as logo recognition, celebrity recognition, animal recognition, plants recognition, and car identification, etc. The Detection module aims to find illegal images which violate the content security policy. A detector is more complex than a classifier, as it involves more components such as object detection and image segmentation. Warnings are generated by cloud-based detectors when the outputs of models exceed the threshold. In this paper, we select the most representative image detection topics, including Nausea, violence, politics, and pornography. All experiments are completed only using the free quota provided by these cloud service providers. This demonstrates that anyone can launch a successful attack by our models in the real world with very low cost.

#### 2.3 White-box and Black-box Attack

Security and privacy in deep learning models have been widely studied. Recent research has proposed several attacks on deep learning models. Based on the prior knowledge possessed by attackers, adversarial examples attacks can be classified to white-box and black-box two categories, as shown in Table 4. Architecture means the parameters of a model, training tools means the training methods used when training the model, and Oracle means whether the model gives an output when queried with an input.

| Attack types | Architecture | Training tools | Train data | Oracle |
|--------------|--------------|----------------|------------|--------|
| white-box    | ✓            | ✓              | ✓          |       |
| black-box    | ×            | ×              | ×          | ✓      |

In this paper, we only consider the black-box attacks against deep learning models, which is even more challenging due to the limited access to the model. In fact, based on the block-box attacks, it is straightforward to design the white-box attacks.

#### 2.4 Image Semantic Segmentation

Semantic segmentation of images includes dividing and recognizing the contents in the image automatically. Semantic segmentation processes an image at the pixel level, thus we can assign each pixel in the image to an object class. With the proposal of a full convolutional neural network [11], deep learning has been widely adopted in the field of semantic segmentation [12] [13] [14] [15].

Fig. 2. Image semantic segmentation example. (a) is the original image and (b) is the semantic segmentation of (a). There are two classes in the original picture: person and horse.

Through semantic segmentation techniques, we can focus on the key pixels of the images, where key pixels means the important pixels contributed the classification results. If we perturb the key pixels, the attacks will become easy. Therefore, the general idea of the attack is using the semantic segmentation model to identify the pixels of images, and then perturb the pixels of a particular class. In this paper, we choose FCN [11] as the semantic segmentation model. For an input \( X \), \( P(i,j) \) is the pixel of \( X \) at location \( (i,j) \), \( C(P(i,j)) \) is the class \( P(i,j) \) belongs to, and


S is the subject pixel set of the input. For instance, in Figure 2(a) we take person as the subject class. Thus, we have

\[ S = \text{subject}(X) = \{ P(i, j) : C(P(i, j)) = \text{person} \} \]

Similarly, we can get the set of animal images. The details of the image semantic segmentation are shown in the Table 5.

In our experiment, results show that perturbation based on semantic segmentation could speed up the generation of adversarial examples in the cases of violence, politician and pornography. Note that for political images, we choose the position of the face due to the importance of face in recognition of these images. In practice, due to the lack of a particular subject class, we are unable to semantically segment the disgusting images.

3 Threat Model and Criterion

3.1 Threat Model

In this paper, we assume that the attacker is just a client and s/he can only access the cloud-based computer vision APIs as a black box. Under the black-box attack, the attacker cannot access to the inner information of the model. The only data the attacker can collect is the feedback form the cloud APIs by the query. Moreover, the attacker can only access the APIs with a limited number of queries since it is inefficient and impractical to conduct a large number of queries in cloud platforms.

3.2 Criterion and Evaluation

The goal of adversarial example attacks in detection is to mislead the detector into misclassification. Nina [16] proposed the concept of top-k misclassification, which means the network ranks the true label below at least k other labels. However, the cloud-based detectors usually produce a label (e.g., violent, pornographic, etc.) after processing an image, which can be used by websites to judge the legitimacy of the image. Consequently, we choose top-1 misclassification as our criterion, which means that our attack is successful if the label with the highest probability generated by the neural networks differs from the correct label.

Evaluating the quality of adversarial images in detection is a challenge since a detector is largely different from a classifier and the quality of adversarial examples cannot be properly measured only based on the number of pixels changed. For the classifier, the objective is to perturb as few pixels as possible to generate adversarial images. For detectors, however, people can still easily recognize the politician in a political image, where many pixels have been perturbed. If the attacker performs some political activities, such as insulting slogans, on this disturbing image, the detection service may fail to block such misdeed. Instead, we consider three evaluation methods including L0, PSNR and SSIM. \( L_0 \) distance corresponds to the number of pixels that have been altered in an image. We assume the original input is O, the adversarial example is ADV. For an RGB image \((m \times n \times 3)\), \((x, y, b)\) is a coordinate of an image for channel \(b\) \((0 \leq b \leq 2)\) at location \((x, y)\). Thus,

\[
L_0 = \sum_{i=0}^{m} \sum_{j=0}^{n} \left| O(i, j) - ADV(i, j) \right|
\]

where \((i, j)\) represents any channel of the pixel at location \((i, j)\).

We also use Peak Signal to Noise Ratio (PSNR) [17] to measure the quality of images.

\[
\text{PSNR} = 10 \log_{10}(MAX^2/MSE)
\]

where \(MAX = 255\), \(MSE\) is the mean square error.

\[
MSE = \frac{1}{mn} \sum_{b=0}^{2} \sum_{i=1}^{m} \sum_{j=1}^{n} |ADV(i, j, b) - O(i, j, b)|^2
\]

Since we want to measure image similarity, structural similarity (SSIM) index is also adopted in this paper. The details of how to compute SSIM can be found in [18]. Therefore, \(L_0\) distance is used to measure how many pixels have been changed, while PSNR value is used to measure image quality, and SSIM value to measure structural similarity. In the following sections, the \(L_0\) distance is used to present the \(L_0\) distance between the original image and the adversarial image for short.

4 Black-box Attack Algorithms

In this section, we mainly talk about the black-box attacks used in our experiments. We list five frequently-used image processing skills, which can make image adversarial in section 4.1. In section 4.2-4.4, we analyze the flaws of previous literature on adversarial example attacks and design Single Pixel attack, Subject-based Local-Search attack, and Subject-based Boundary attack suited to the cloud platforms.

4.1 Image Processing

In the following, we explore the effect of five different image processing techniques on both the classifier and the detector, including Gaussian Noise, Grayscale Image, Image Binarization, Salt-and-Pepper Noise, and Brightness Control. Prior work such as Hossein et al. only discussed Salt-and-Pepper noise on Google vision APIs [19]. All these image processing techniques are implemented with Python libraries, such as sciimage [5], OpenCV [6], and PIL [7].

4.1.1 Gaussian Noise

Gaussian noise is statistical noise with a probability density function (PDF) equal to the normal distribution as shown in Equation 1.

\[
f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)
\]

5. https://scikit-image.org
6. https://opencv.org
7. http://www.pythonware.com/products/pil
where $\sigma$ and $\mu$ represent the mean and variance, respectively. Thus

$$ADV = O + \text{Noise}$$  \hspace{1cm} (5)

Note that, clipping $ADV$ is necessary to maintain a reasonable RGB value, which is $[0,255]$.

### 4.1.2 Grayscale Image

A grayscale image is one in which the value of each pixel is a single sample representing only the amount of light, i.e., it only carries intensity information. In the computer vision field, the black and white image contains only black and white pixels, while the grayscale image has many levels of color depth between black and white. RGB means three channel values of pixels, namely, red, green and blue. To obtain the $ADV$, a common equation is used:

$$ADV = R \times 0.299 + G \times 0.587 + B \times 0.114$$  \hspace{1cm} (6)

This function was also implemented in Python library (PIL). Clipping the $ADV$ is necessary to maintain a reasonable RGB value.

### 4.1.3 Image Binarization

A binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white. Floyd-Steinberg dithering is used to approximate the original image luminosity levels in the implementation of PIL.

### 4.1.4 Salt-and-Pepper Noise

Salt-and-Pepper noise is also known as impulse noise. This noise can result in sharp and sudden disturbances in the image signal. For a pixel $(i,j,k)$ of an RGB image, the noise image of Salt-and-Pepper is calculated by $\text{Noise}(i,j,k;p)$, where $\text{Noise}$ is the noise image, and $p$ is the noise density. The other pixels remain original.

8. https://en.wikipedia.org/wiki/Grayscale

### 4.1.5 Brightness Control

We hypothesize that the image brightness may affect classification results. Thus we iteratively adjust the brightness of the image to observe the change of the result. A constant value is added to all pixels of the image at the same time. Then we clip it to a reasonable range, namely, $[0,255]$.

$$ADV = O + \varepsilon \times 255$$  \hspace{1cm} (7)

where $\varepsilon$ is the parameter to control brightness, which is in the range of $[0-1]$.

Examples of the above five image processing techniques on a political image are shown in Figure 3.

### 4.2 Single-Pixel Attack

The single-pixel attack proposed by Nina et al. [16], is an attack where perturbation of a single pixel would make the classifier generate a wrong label. However, it suffers from several limitations. First, when processing high-resolution images, a single pixel is not enough to cause the misclassification. Second, Nina’s experiment was completely offline and the data were fed to the classifiers directly. Given the online classifier in the physical world, attacks are more difficult to succeed.

To address these problems, we design a new single-pixel attack by gradually increase the number of modified pixels and integrating the idea of image semantic segmentation. In order to verify the validity of a semantic segmentation, we implement the attack in three areas of the image, namely, subject region, non-subject region, and random region. The random region is chosen as a baseline to compare with the other two regions. These three regions are defined as follows:

- **subject region**: a region composed of all pixels which belong to a subject class.
- **non-subject region**: a region composed of all pixels which do not belong to any subject class.
- **random region**: a region composed of all pixels chosen from the image randomly.

Other types of image segmentation are shown in Table VII. In the rest of this paper, we name Single-Pixel attack as SP attack since inspired by single-pixel perturbation.
4.3 Subject-based Local-Search Attack

Nina et al. [16] proposed local greedy algorithm, where it searches the optimal perturbation of local area based on iteration. However, there are many flaws that makes this attack ineffective on cloud classifiers or detectors. Firstly, they only conducted the test offline on the VGG model [21]. Secondly, they fed the image data to classifier directly. Kurakin [22] pointed out that the transformations applied to images by the process of printing them may have a negative effect on adversarial examples. If we save the RGB value of an image using lossy compression, the resistance may disappear. Our experiment also shows that JPEG format hinders the adversarial examples to a certain extent, while PNG format does not. The reason is that PNG format uses lossless compression coding. If Nina sent the perturbed RGB values to the VGG model and generate the adversarial example, it may be not adversarial to online cloud classifiers using JPEG format. Thirdly, Nina et al. initialized a very large perturbation (more than 500) to detect the probability changes of the classifier, which is impractical for cloud classifier. However, one can only send an image to it and the RGB value of the image is between 0 and 255. Namely, we cannot send an image with an RGB value of 500 to the cloud. Finally, if the initial disturbance region is very large, it is easy to fall into a local optimum, making it difficult to find adversarial examples. Based on the above reasons, local greedy algorithm [16] does not apply to cloud classifiers or detectors.

In this paper, we propose Subject-based Local-Search (SBLS) Attack by incorporating semantic segmentation to speed up the attack and saving all images with PNG format to retain their original features. Considering the online models, the initial modified pixel values is 0 or 255 and it is within the range of RGB value. The main steps of our algorithm are summarized as follows.

1) Firstly, we obtain the subject region of the image by semantic segmentation techniques.
2) Secondly, 50 pixels are selected from the subject region randomly and the image is perturbed on each pixel one by one. The perturbed images are fed to the cloud-based detectors and 50 predictions are produced. The first N (10 in this paper) pixels for which the probability drops the most are picked.
3) Thirdly, the image is perturbed on the N pixels and the algorithm will record whether the prediction result changes from illegal to normal.
4) Finally, the perturbed image is taken as the initial image of the next round and cycle through the above steps until the label becomes normal.

The pseudocode for the algorithm is shown in Algorithm 1.

![Algorithm 1](image)

4.4 Subject-based Boundary Attack

Boundary Attack solely relies on the final model decision [6], which is also called decision-based attack. A decision-based attack starts from a large adversarial perturbation and then seeks to reduce the perturbation while staying adversarial. This method works in theory, but is inefficient. About 1.2 million predictions were used in the Boundary Attack to find an adversarial image for ResNet-50 [6], which is a huge overhead for the cloud service APIs. If we want to generate hundreds of adversarial images, the required time and expense will be unbearable.

To make the attack practical, we design a new Subject-based Boundary (SBB) Attack by incorporating semantic segmentation and greedy algorithm. By semantic segmentation, the subject region is first perturbed with the average RGB value of the non-subject region, since the background color has great influence on the recognition of a subject class based on the previous experiments. Through greedy algorithm, the attack is able to recover as many of the perturbed pixels as possible, making the probability of correct classification as small as possible. The main steps of the algorithm are summarized as follows.

1) First, all pixels in the subject region are perturbed, which makes the image normal and the probability close to zero.
2) Then, the distance between the current perturbed image and the original image is computed. A certain percentage of pixels, which are selected from the different pixels between the original image and the previous one, are recovered randomly.
3) Next, the recovery process is repeated to choose the best recovering, which leads to the slowest increase
in probability or score. The perturbed images are still recognized as normal by the cloud APIs.

4) Finally, steps 2-3 are repeated to minimize the distance of the perturbed image until the image is correctly classified. If the perturbed image is recognized as illegal, the iteration will be stopped and the last perturbed image will be returned as the adversarial image.

The pseudo-code of the algorithm is shown in Algorithm 2.

**Algorithm 2 Subject-based Boundary (SBB) Attack.**

**Input:** Origin image I, round: R, detections: D.

**Output:** adversarial examples;

1. S = subject(I),
2. Non-S = I - subject(I),
3. averpixel = getaverpixel(Non-S),
4. ADV = perturb(I, S, averpixel)
5. while r < R do
6.     Step = \( L_0(I, adv)/10 + 100 \)
7.     for \( j = 0 \) to \( j < detections \) do:
8.     Advtemp = recover(step, I, ADV)
9.     Advcandidate.append(advtemp)
10. Label, Prob = cloud.predict(Advtemp)
11. if Label == normal then
12.     Probs.append(Prob)
13.     end if
14. end for
15. if len(Probs) == 0 then
16.     break
17. end if
18. Index = argmax(probs)
19. ADV = Advcandidate[Index]
20. end while
21. return ADV;

the getaverpixel(Non-S) means getting the average pixel value of Non-S region, \( L_0(I,ADV) \) means getting the \( L_0 \) norm distance between I and ADV, recover(step, I, ADV) means recovering step pixels according to the difference of I and ADV.

5 EXPERIMENTAL EVALUATION

5.1 Validation of Semantic Segmentation

Since cloud-based detectors are based on classifiers and attacking detectors is more difficult than classifiers, we first conduct the SP attack on classifiers to better understand the validity of semantic segmentation. The results of classifiers can help us adjust our attack algorithms. We choose the SP attack because it belongs to coarse-grained perturbation. If the SP attack works well with semantic segmentation techniques, SBLS and SBB attacks should have better performance since both of them belong to fine-grained perturbation. Several local models and a cloud classifier are used in experiments. The local models we are using are VGG16 [20], Resnet50 [21] and InceptionV3 [22]. In this paper, we leverage the Keras framework and pre-trained deep learning models [9] to conduct experiments. These pre-trained models are trained with ImageNet datasets since ImageNet as a standard dataset is widely used in deep learning classifiers. Due to the easier usage of Baidu APIs, we choose Baidu animal classifier as an example.

5.1.1 Datasets

100 animal images are selected from the ImageNet set. Because VGG16 and Resnet50 both accept input images of size \( 224 \times 224 \times 3 \), every input image is clipped to the size of \( 224 \times 224 \times 3 \), where 3 is the number of RGB channels. Only animal images are selected for simplicity. The attack strategy mainly consists of two parts: perturbation methods and perturbation regions.

- Different methods of perturbation have different effects on the prediction results. Three types of perturbation are considered: \( P=0 \), \( P=255 \), and \( P=2 \), where \( P \) is a perturbation parameter. For instance, \( P=0 \) or 255 means setting the RGB value of the pixel to 0 or 255, where 0 represents black and 255 represents white. \( P=2 \) means multiplying the pixel value by 2 and clipping it to a reasonable range.
- In order to verify the effectiveness of semantic segmentation in perturbation, three perturbation regions described in section IV are chosen to test.

5.1.2 Results and Analysis

First, we carry out a precursor experiment by 1) perturbing pixels that do not belong to subject class using different perturbation methods; 2) recording the changes in prediction. The examples are shown in Figure 4. The overall result is shown in Table 6. It is evident from the results that perturbation of non-subject region can cause a misclassification rate of up to 0.8. VGG16 model is the least resilient against all three perturbation methods, while InceptionV3 and Online are the two most robust models. Besides, the results of \( P=2 \) has the lowest successful misclassification rate among three perturbation methods since it is only slight perturbation. From the experimental results, it can be seen that about 60% image classifications of InceptionV3 are not affected with perturbation on the background pixels. Thus, it is necessary

![Figure 4. The pixels of background are set as different P values. Here we just change the pixels of non-subject region.](Image 320x309 to 391x380)

![Figure 4. The pixels of background are set as different P values. Here we just change the pixels of non-subject region.](Image 403x309 to 473x380)

![Figure 4. The pixels of background are set as different P values. Here we just change the pixels of non-subject region.](Image 485x309 to 556x380)

**TABLE 6**

Successful misclassification rate. We perturb the pixels of non-subject region and record the misclassification.

| Background | VGG16 | Resnet50 | InceptionV3 | Online (Baidu) |
|------------|-------|----------|-------------|---------------|
| \( P=255 \) | 0.8   | 0.58     | 0.38        | 0.40          |
| \( P=2 \)  | 0.53  | 0.41     | 0.25        | 0             |
| \( P=0 \)  | 0.68  | 0.56     | 0.39        | 0.40          |

9. https://github.com/fchollet/deep-learning-models/releases
to perturb the subject pixels to improve the accuracy of the attack.

We conduct a number of experiments to understand the choice of perturbation parameters and regions on different models, as shown in Figure 5. Firstly, we adjust the number of perturbed pixels and record the effects on four classifiers. As shown in Figure 5(a), the number of successful attacks increases as the number of perturbed pixels increases. After reaching a large perturbation, the success rate increases very slowly. Compared to three other local classifiers, attacking online classifier (Baidu) is more difficult.

Secondly, for a single classifier, we select the perturbation regions in three different ways to observe the changes in the prediction results. We take the online classifier (Baidu) for example. Figure 5(b) shows that it is very sensitive to perturbation on the pixels in the subject region, resulting in a high misclassification rate. Figure 5(b) also demonstrates the results when we use \( P=2 \) to perturb the pixels. However, surprisingly, perturbation in random regions performs even better than subject regions. Our conjecture is that slight perturbation in the non-subject region makes the pixels after perturbation close to those in the subject region. In order to verify our conjecture, we perturb the non-subject pixels with the average RGB value of the subject region. For all pixels in the subject region, the average RGB value in each channel is computed and the non-subject region is perturbed with the derived average value. The results in Figure 5(c) demonstrate that perturbation with the average value performs better than that with \( P=2 \). When perturbing 2000 pixels, perturbation with the average value performs best.

**Conclusions.** We draw the following conclusions based on the results from classifiers:

- Online models have **good robustness** and attacking online models is more difficult than attacking local models under the same number of pixels of perturbation.
- Perturbations in **subject regions** are more effective than other regions.
- The prediction of the model is sensitive to the magnitude of the disturbance value, especially the value close to the pixels of subject region.
- Larger disturbance values are more destructive to image recognition than smaller disturbance.

### 5.2 Attacking Cloud-based Detectors

In this paper, we explore the security issues in cloud-based detectors, thus we conduct four kinds of attack described in section IV on real-world cloud platforms.

#### 5.2.1 Datasets and Preprocessing

To test cloud-based detectors, 400 images are selected from Google Images or Baidu Images. For the four areas to be detected, 100 images for each area are manually selected and labeled. All images are resized to a fixed size: \( 224 \times 224 \times 3 \). Note that these images properly followed the requirements of the tested APIs, including input format, size and resolution and these images were collected legally. Because the images are labeled by us, the illegal images may not be identified by detectors. We first filter these images by calling the detectors. Only images labeled as illegal by the detectors are retained. The results are shown in Table 2 where “-” means the platform does not provide the API service.

| Platforms | Pornography | Violence | Politician | Disgust |
|-----------|-------------|----------|------------|---------|
| Baidu     | 95/100      | 32/100   | 45/100     | 98/100  |
| Google    | 90/100      | 30/100   | -          | -       |
| Alibaba   | 67/100      | 67/100   | 49/100     | -       |
| Azure     | 54/100      | -        | -          | -       |
| AWS       | 84/100      | -        | -          | 57/100  |

According to Table 2, we can learn that Baidu, which label 95% of the pornography correctly, has done a better job than other cloud platforms. To our surprise, 46% of the pornographic images have not been identified by Azure’s detector. For violent images, Alibaba’s detector has the best performance since it label 67% of the images correctly. The detectors of Google and Baidu only recognize 30% and 32% of the images, respectively. One reason can be that the scenes of violent images are more complex and the detectors do not consider a variety of scenarios.

The images to be tested are carefully selected and filtered, as shown in Table 2. For instance, Alibaba will offer suggestions for manual review to suspicious images and thus we exclude such images. About 30 pornographic images are discarded. For Google’s detector, we can only obtain a single-world result, e.g., **POSSIBLE, LIKELY** in Table 2. Only the images whose labels are **LIKELY and VERY LIKELY** are considered. About 93% pornographic images are labeled as **VERY LIKELY** and 7% are labeled as **LIKELY** by Google’s detector. For violent images, 83% are labeled as **VERY LIKELY** by Google’s detector, and the others are labeled as **LIKELY**.

To better understand the quality of these images, we record the probabilities or scores when the API is called to detect these images. The detailed information can be found in Figure 6. As shown in Figure 6, the majority of images are labeled by the APIs with very high confidence. For Azure’s detector, 80% of the probability labels are over 0.7. In the following subsections, we only test the images which are identified correctly by cloud APIs. Using only 400 images for detection, we clearly demonstrate the ability of our attack algorithms.

#### 5.2.2 Detectors of Pornographic Images

The Internet is flooded with pornographic images, which is a serious problem for website regulators. Websites often leverage detectors to detect these illegal images. Evasion attacks on these detectors can result in huge content security risks. The four cloud platforms all provide pornographic image detection services. The results returned by these detectors are different, as shown in Table 3.

**Image Processing.** The success rates of IP attacks are shown in Figure 7(a). From Figure 7(a), we know that the detectors of Azure, AWS and Google on pornographic images are vulnerable to the Gaussian noise attack. The Grayscale attack has a slight effect on Google and Azure, and no effect on Baidu and Alibaba. For Binarization, Salt-and-Pepper and Brightness attacks, the success rates all increase as the parameters increase.
Fig. 5. In (a), we randomly select pixels on the image to perturb and increase the number of perturbed pixels. The figure records the number of misclassifications under different numbers of perturbed pixels. In (b), we test the different regions of image with $P = 255$ and $P = 2$ and record the number of successful misclassification. The horizontal axis is the number of the perturbed pixels, and the vertical axis is the number of successful misclassification. In (c), we perturb the pixels in the non-subject region with different $P$ values. $P = \text{average}$ means that we perturb the non-subject region with average RGB value of subject region.

Fig. 6. We show the probabilities and scores of test images. Most points are concentrated around high probabilities. Besides, many points are overlapped where the probability is 1 or the score is 100.

Fig. 7. For pornographic images, we use five Image Processing techniques to test these cloud platforms and show the success rates and PSNR values of successful adversarial images.
In order to evaluate these successful adversarial images, PSNR value [17] is used. The average PSNR values of all successful adversarial images are shown in Figure 7(b). As shown in Figure 7(b), we can learn that the results of Gaussian noise and Grayscale attacks are very promising. Usually, values for the PSNR are considered between 20 and 40 dB, (higher is better) [17]. The Binarization attack makes pornographic images contain only black and white pixels, which affects the image quality greatly and leads to the misclassification problem of detectors. On the other hand, the Binarization attack also significantly reduces the content visibility to humans in pornographic images. Thus, the PSNR values are very small. For Salt-and-Pepper and Brightness attacks, we keep increasing the attack parameters to achieve high success rates, which affects the PSNR values.

Table 8. Forms of prediction. We record the returned information and labels for every platform.

| Cloud | Feedback | Label Category |
|-------|----------|----------------|
| Baidu | Probability | Porn, Sexy, Normal |
| Google | N | UNKNOWN, VERY_UNLIKELY, UNLIKELY, POSSIBLE, LIKELY, VERYLIKELY |
| Alibaba | Score | porn, sexy, normal |
| Azure | Probability | True, False |
| AWS | Score | Explicit Nudity, Nudity, Graphic Female/Male Nudity, Sexual Activity, Suggestive, Revealing Clothes, No label (normal) |

Figure 8. Cumulative distribution function. (a) is the CDF figure of success rates for Salt-and-Pepper attack and (b) is the CDF figure of success rates for Brightness attack. The horizontal axis is the corresponding parameter.

Single-Pixel Attack. We conduct the Single-Pixel (SP) attack on the cloud platforms with perturbation in different regions of the images, as shown in Figure 9(a). For instance, Baidu-s means the perturbation of subject region on Baidu, and Baidu-r means the perturbation of random region. As shown in Figure 9(a), the effect of the perturbation in the subject region is much more significant than that in the random region in the case of perturbing the same number of pixels, which verifies the validity of semantic segmentation. Besides, we find that the success rate can be reduced by perturbing more pixels on random regions. Moreover, even if 2000 pixels are perturbed, the success rates of attacks on Baidu and Alibaba still remain low, whereas 91% of pornographic images can bypass Azures detectors. This demonstrates that the detectors of Baidu and Alibaba are more robust to SP attack than others.

In order to evaluate the picture quality, we choose to perturb subject regions. The PSNR values are shown in Figure 9(b). The majority of PSNR values are larger than 10, which means the values of most successful adversarial images are within the acceptable range.

Subject-based Local-Search Attack. We set the maximum number of cycles $R$ to 30 in the Subject-based Local-Search (SBLs) attack. Besides, we set $P$ to 255 and $D$ to 10, which correspond to the RGB value and expanding 10 unit pixels in the next loop as described in Algorithm 1 respectively. Previous experiments have demonstrated the effectiveness of semantic segmentation. Thus, the evaluation of subject-based adversarial images is statistically analyzed in following subsections. We skip Google’s detector for SBLs attack since Google’s does not return probability or score and SBLs attack must rely on them. We do not conduct SBLs attack or SBB attack on AWS’s detector since AWS only provides 5000 queries each month for free, which is not enough for us to test hundreds of images with SBLs attack or SBB attack.

The results of this attack are shown in Figure 10. Since small $L_0$ distance means fewer queries, we can obtain the queries based on $L_0$ distance. According to Figure 10, Azure’s detector is the weakest of all as 54% adversarial im-
ages on Azure’s detector cannot be detected. The minimum number of queries is 50 and the minimum $L_0$ distance is 10, which means that one round in an SBL attack is enough to generate an adversarial image. The average PSNR values are all over 20 and the average SSIM values are all over 0.9. In other words, good adversarial images are obtained through only a few queries. Note that the prediction is normal, not a similar illegal category. The success rates of attacking Baidu and Alibaba are 34% and 19%, respectively. Although the success rates on Alibaba’s detector are the lowest, the average number of queries is only 101, which means two rounds of iterations is enough to achieve the attack. Among the successful adversarial images, all SSIM values are over 0.9 and all PSNR values are over 20. Based on the $L_0$ distance, we modify about 0.01%-0.6% pixels of the whole image (224 × 224 pixels). These data suggest that the quality of these adversarial images is very high. It is easy for people to observe pornographic information from these adversarial images.

Subject-based Boundary (SBB) attack as described in Algorithm 2. Initially, we set $D = 30$, $Step = L_0(I, adv)/10 + 100$. The step size is set based on the tradeoff between accuracy and efficiency. The results are shown in Table 9.

As shown in Table 9 the successful adversarial images have lower similarity than SBL attack. However, good adversarial examples still exist in our experiments. For Baidu’s detector, the minimum $L_0$ distance is 84 and the maximum SSIM value is 0.97. For Azure’s detector, the minimum $L_0$ distance is 2 and the maximum SSIM value is 0.98. The majority of PSNR values are over 20, which means good image quality. Moreover, thousands of queries are sufficient to conduct SBB attacks.

Subject-based Boundary Attack. Finally, we conduct...
5.2.3 Detectors of Violent Images

Similarly, violent images are segmented with semantic segmentation model. An example of segmentation is shown in Figure 13. The pixels of the subject region (persons) are painted white. Noted that we only consider the violent images which contain persons. In fact, person is a subject class and plays an important role in the identification of violent images. Besides, subsequent experiments all focus on perturbation in the subject regions except IP attacks.

**Image Processing.** The success rates of IP attacks are shown in Figure 12(a). We find that the success rates of IP attacks are extremely high and violent image detectors are easier to attack than pornographic. Similar to pornographic images, Gaussian noise and Grayscale attacks can generate adversarial violent images which have high-quality grade according to PSNR values.

We show successful adversarial images in Figure 15. Figure 15(a) and Figure 15(d) are labeled as violent by the detectors of Alibaba and Google, respectively. However, Figure 15(b) and Figure 15(e) which have been added with Gaussian noise are labeled as normal. Besides, grayscale images of Figure 15(a) and Figure 15(d) are also labeled as normal. Although the colors have been changed, we can still easily identify guns and terrorists from the images. The detailed PSNR values of IP attack on violent images can be found in Figure 12(b).

**Single-Pixel Attack.** The success rates of SP attack are shown in Figure 14. To our surprise, the detectors of Baidu and Alibaba are not resistant against SP attack on violent images due to the high attack success rate, which is very different from their performance on the pornographic images. Google’s detector is not vulnerable to SP attack on violent images since we cannot launch a successful SP attack on it. We speculate that different companies have different content security priorities. For instance, Google in the United States may focus more on images filled with violence and terrorism, while Baidu and Alibaba in China are faced with stricter censorship on pornographic images.

**Subject-based Local-search Attack.** Similarly, the SBLS attacks are conducted on violent images. We set P to 255 or 0 for different platforms and we choose the best results. The results are shown in Figure 16. The success rates of attacking Baidu and Alibaba are 100% and 72%, respectively. In other words, we can make all images adversarial on the detector of Baidu with SBLS attack within a limited number of queries. The average value of queries is 200 for Baidu’s detector, and the average value of $L_0$ distance for Baidu’s is 38, which means only the modification of 38 pixels on average could make the violent images adversarial. Besides, the average SSIM value is 0.99 for Baidu’s detector, which reveals a high degree of similarity. Although Alibaba’s detector is more robust than Baidu’s on violent images, the quality of adversarial images for Alibaba’s is also sufficiently good. The average SSIM value for Alibaba’s detector is also 0.9.

The scatter diagram of various evaluation indicators is plotted in Figure 17. As indicated in Figure 17, most SSIM values of the adversarial images are over 0.98 and most $L_0$ distances are below 100. Additionally, the quality of adversarial images on Alibaba’s detector is better than Baidu’s, since the majority of Alibaba’s points are above Baidu’s. Examples of the adversarial images can be found in Figure 18.

**Subject-based Boundary Attack.** We also carry out Subject-based Boundary attack on violent images. The re-
A successful adversarial image is shown in Figure 19.

TABLE 10
SBB attack on violent images.

|       | Average | Baidu | Google | Alibaba |
|-------|---------|-------|--------|---------|
| Success Rate | 0.91     | 0.84  | 0.78   | 0.67    |
| Query    | 724      | 17    | 696    | 592     |
| \(L_0\)  | 3515     | 461   | 461    | 461     |
| PSNR     | 34       | 32    | 32     | 32      |
| SSIM     | 0.93     | 0.73  | 0.87   | 0.79    |
pixel until the detector recognizes it as illegal. In round 22, the detector marked the image as violent, so the image of round 21 is chosen as the best adversarial image.

![Image](image1.png)

**Fig. 19.** This is a successful example of SBB attack on Baidu’s detector. In round 21, the image is still adversarial and the $L_0$ distance between it and original image is just 10, which means we just perturb 10 pixels.

### 5.2.4 Detectors of Political Images.

Both Baidu and Alibaba provide the cloud services to detect whether a picture contains politicians, since sensitive political images can be insulted on the Internet. We are interested in exploring the adversarial attacks that aim to interfere the results from detectors of such political images. The political images come from several countries, such as China, United States, Japan, South Korea, etc. The subject of political images is defined as person’s face since faces determine the identification of politicians. In order to find the location of the face in the image, we adopt face detection service in open cloud platform. In this paper, we choose the face detection API from Baidu since it is free and easy to use. In Figure 20, (b) is the extreme case of an adversarial image, where the whole face area is perturbed. Generating Figure 20.(b) is to verify the validity of the face area, which can speed up subsequent attacks except for IP attack. We only need to perturb the face area rather than the entire persons pixels.

![Image](image2.png)

**Fig. 20.** Face detection of a political image: (a) is a original image and (b) is the adversarial image where we perturb all face area to make detector misclassified.

**Image Processing.** Firstly, IP attacks are used to test political images. We find that the Gaussian noise attacks make about 76% images on Baidu’s detector and 57% images on Alibaba’s adversarial. For AWS’s detector, the success rate is 45%. 20% images on Baidu’s detector and 33% images on Alibaba’s can evade detection using the Grayscale attack. To our surprise, Binarization attacks perform very well. Unlike pornographic and violent images, low-quality political images do not affect human judgment. Several successful adversarial images are shown in Figure 21. Although the images quality are damaged through binarization, we can still recognize these politicians easily. For Salt-and-Pepper noise, the average values of $p$ are all $0.05$ on the detectors of these platforms. Among the successful adversarial images of Brightness attack, the average values of $\varepsilon$ are about 0.5. These results imply that the quality of the images is still good.

![Image](image3.png)

**Fig. 21.** (a) and (b) are both binary images. They are labeled normal, while their original images are all labeled politician.

**Single-Pixel Attack.** Then, the SP attacks of subject region on political images are considered. The success rates are shown in Figure 14(b). Based on the figure, political images are vulnerable to SP attack. Even hundreds of pixels are perturbed, the quality of images is still acceptable.

**Subject-based Local-Search Attack.** Next we conduct SBLS attacks on political images. The success rates of Baidu’s detectors and Alibaba’s are 60% and 46%, respectively. Only 222 and 382 queries are needed on average. Here we show an example in Figure 22. The $L_0$ distance between Figure 22(a) and Figure 22(b) is 10, which means we only perturb 10 pixels to make the image adversarial. Among all successful examples on Baidu’s detector, the average $L_0$ distance is 54, the average SSIM value is 0.98. Figure 22(c) is adversarial image misclassified by Alibaba’s detector. Even noticeable perturbation have been added in this image, people can still recognize Kim Jong-un easily through the adversarial image. In other words, if we upload the image to the cloud, the detector will not give any warning about the politician.

![Image](image4.png)

**Fig. 22.** Minimum adversarial political image: (b) is the adversarial image of (a) through SBLS attack, and (c) is the adversarial images of Figure 20.(a). Among adversarial images, the $L_0$ distance of these two are the least for the detectors of Baidu and Alibaba, respectively.

The scatter plot of various evaluation indicators is drawn in Figure 23. As illustrated in Figure 23, most SSIM values...
of adversarial images are near 1 and most PSNR values of adversarial images are within an acceptable range. Additionally, when the SBLS attack is conducted on the detectors of political images, all $L_0$ distances are no more than 150.

![Graph](image)

Fig. 23. We conduct the SBLS attack on political images and record the evaluation indicators among successful adversarial images. (a) is the scatter plot between SSIM values and $L_0$ distances, and (b) is the scatter plot between PSNR values and $L_0$ distances.

**Subject-based Boundary Attack.** Finally, SBB attacks are used to test the robustness of models. Through SBB attacks, we can make 82% images adversarial on Baidu’s detector and 98% images adversarial on Alibaba’s. Moreover, the attack only needs 601 and 375 queries on average, depending on the platform. All SSIM values are over 0.9, which represents a very high similarity between original images and adversarial images. The median value of $L_0$ distance on Alibaba’s detector is just 334. A successful example is shown in Figure 24, where only 17 iterations are required. We first perturb the whole face area to make the image adversarial. Then the original pixels are recovered step by step until the image is labeled political. In Figure 24 the loop stops in round 17. Thus the image in round 16 is considered the best adversarial image.

![Images](image)

Fig. 24. This is a successful example of SBB attack on political images. We keep restoring the original pixels until the image is recognized as illegal. In round 16, the image is still adversarial. In round 16, the image is recognized as a politician.

## 6 DISCUSSION

### 6.1 Effect of Attacks

Unlike previous attacks, our attacks do not require massive queries. For some attack methods (e.g., IP attack), the detectors can be bypassed without any queries. For an iteration-based attack, only less than two thousand queries are needed to generate a good adversarial image, which can be done with the **free quota**. Based on our investigation, almost all cloud service APIs have free invocation quotas and service providers that allow a trial period for registered users are shown in Table 11. Thanks to the effective attack design, all of our experiments are completed within the free trial or quotas.

![Table](image)

**Table 11** Cloud services free quotas. *apply means customers can apply for free quotas from Baidu staffs.

| Baidu | Google | Alibaba | Azure | AWS  |
|-------|--------|---------|-------|------|
| 3000/day | a year | fee ≤ 300$ | 3000/day | fee ≤ 250$ | 5000/month |
| +20000(apply) | a month | a month | a month | a year |

Secondly, different from any of previous work, our attacks can affect cloud services to a certain extent. In our experiment, we find that the detectors are easily tricked. Although noticeable perturbation can be found in the adversarial examples, they are close to original images or do not affect people’s judgment. For instance, people can still recognize perturbed political images easily while detectors fail to do so. Such attacks can help spread illegal images, which breaks the content security of the Internet. Moreover, attackers can utilize defects of these services to make money.

### 6.2 Defenses

Since these attacks pose a significant threat to cloud services, it is crucial to design the defense mechanisms against these attacks. In [19], the authors claimed that **noise filter** can counter the perturbation. However, noise filter can only defend against noise attacks, like Salt and pepper noise, Gaussian noise, and Brightness attacks. For Grayscale and Binarization attacks, there is still no effective defense approach developed yet, since they have changed the style of images. Besides, in [26] [27] [28], **adversarial training** was proposed to improve the robustness of deep learning model. They iteratively create a set of adversarial examples and include them into the training data. Retraining the model with new training data may be very helpful. For instance, cloud service providers can collect plenty of grayscale and binarization images or adversarial examples which are generated from the SBLS attack and SBB attack. In other words, a new deep model is trained to detect these illegal images. Although adversarial training can defend against these attacks to a certain extent, obtaining adversarial examples is costly. In [29] [30], the researchers proposed a mechanism where a detection is needed before feeding data to models. Certainly, **detecting inputs** can avoid some strange adversarial examples but also increase false positives, since the input in the real world is similar to an adversarial example. Besides, in [31], the authors showed that detection mechanism can also be bypassed. **Limiting the queries** is a straightforward strategy but can be impractical. Since many websites which rely on the cloud services may call the APIs multiple times within a short period to review the content of their webpages. In [32] [33], **randomization** was proposed for mitigating adversarial effects. We believe this
might be a good direction in deploying defense mechanism. The models make decisions through multiple classifiers, which can increase the robustness of the model. In addition, since our attacks (SBLS,SBB) rely on confidence, rounding confidence scores of output or giving label only to some fixed precisions is also a good defense.

7 Related Work

Previous works mainly study the security and privacy in deep neural network via white-box mode [8] [14] [25] [35] [36] [37]. In the white-box model, the attacker can obtain the adversarial examples quickly and accurately. Besides, the perturbation is too small to be perceived by people. However, it is difficult for the attacker to know the inner parameters of models in the real world. For instance, the architecture and parameters of deep models on cloud platforms cannot be obtained by the attack. The attacker can only access the APIs opened by cloud platforms. Thus, the black-box attack of neural networks is more threatening.

The researchers have launched some black-box attacks on deep neural networks recently. In [2], Papernot et al. proposed that the attacker can train a substituted model, which approaches the target model, and then generate adversarial examples on the substituted model. Their experiments showed that good transferability exists in adversarial examples, but the attack is not totally black-box. They have knowledge of the training data and test the attack with the same distributed data. In [5], Liu et al. adopted an ensemble-based model to improve transferability and successfully attack Clarifai.com. However, the classifiers are greatly different from detectors. Several existing classifiers models, like VGG16, Resnet50, and InceptionV3 perform well on classifying the test images, which can contribute to the generation of adversarial examples. It is difficult for attackers to get open-source models which are used to detect illegal images. Especially for complicated detectors, transferability is not very well.

The query-based black-box attack has also been explored by researchers. In [38] [39] [40], researchers can get inner information of models through lots of queries, but it is impractical in commercial detectors. In [41], thousands of queries are required for low-resolution images. For high-resolution images, it still takes tens of thousands of times. In [4] [42], both of them proposed gradient estimation black-box attacks for adversaries with query access to the target model’s class probabilities, nevertheless, faced with high-resolution images, millions of queries are required, which is very inefficient and impractical in the real world. In [6], Brendel proposed a decision-based attack that starts from a large adversarial perturbation and then seeks to reduce the perturbation while staying adversarial, but they need 1,200,000 queries on average to generate a good adversarial example for high-resolution images. In this paper, we utilize the semantic segmentation technique to speed up the process. Besides, we choose the important pixels according to the returned probability. Finally, only thousands of queries are required to generate pretty good adversarial examples. In [16], Nina et al. proposed a greedy local search algorithm to attack models via black-box mode. The SBLS attack in this paper is originated from [16], but we have modified it to adapt to cloud platforms. They do not attack the models on the real world. For instance, setting the RGB value to a value bigger than 255 is impractical in real images.

Several other forms of black-box attacks existed in recent works. In [43], Anish et al. constructed real-world 3D objects that consistently fool a neural network across a wide distribution of angles and viewpoints. They show that adversarial examples are a practical concern for real-world systems, but they focus on classifiers only. In [19], Hossein et al. found that the API generates completely different outputs for the noisy image, while a human observer would perceive its original content. We expand the experiment and use five image processing skills to attack models of four cloud platforms. When faced with detectors, we find Salt and pepper noise is not valid enough and other image processing methods perform well. In [44], Seong et al. inferred the inner information of models through multiple queries and the revealed internal information helps generate more effective adversarial examples against the black-box model. Even thousands of models are trained to infer useful information. However, this is impractical and there are no multiple candidate models to infer when handling detectors.

8 Conclusion and Future Work

In this study, we conduct a comprehensive study of security issues in the cloud-based detectors. We design four kinds of attack and verify them on major cloud service platforms. According to our experimental results, we find that cloud-based detectors are easily bypassed. In particular, Azure’s detector is the weakest on pornographic images since these attacks can be achieved within hundreds of queries, which makes the attacks practical in the real world. Moreover, we find that Baidu and Alibaba performs better on the detection of pornographic images than Google and Azure, while Google in the United States has higher success rates at the detection of violent images. For political images and disgusting images, the detectors are easier to be attacked than pornographic images. We reported our findings to the tested cloud platforms and received positive feedbacks from them.

In the future, we aim to explore the space of adversarial examples with less perturbation in black-box. We are interested in the optimization of queries to further improve the effectiveness of our algorithms. A general method to attack all platforms with a small number of queries would be another meaningful topic. It is also important to design the defense mechanisms for these cloud services against adversarial example attacks+. For instance, the cloud platforms can perform effective detection before outputting a label and distinguish between malicious tests and normal samples. We hope our work can help cloud platforms to design secure services and provide some inspiration to researchers who study the security in deep learning models.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. arXiv preprint arXiv:1801.00553, 2018.
[2] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pages 506–519. ACM, 2017.

[3] Jamie Hayes and George Danezis. Machine learning as an adversarial service: Learning black-box adversarial examples. arXiv preprint arXiv:1708.05207, 2017.

[4] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pages 15–26. ACM, 2017.

[5] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770, 2016.

[6] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decisional properties of neural networks. arXiv preprint arXiv:1704.02448, 2017.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[8] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks, 32:323–332, 2012.

[9] I Rosenberg, A Shabtai, L Rokach, and Y Elowici. generic black-box end-to-end attack against state of the art api call based malware classifiers, 2017.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[11] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561, 2015.

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1606.02147, 2016.

[13] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D Reid. Refinedet: Multi-path refinement networks for high-resolution semantic segmentation. In Cog, volume 1, page 5, 2017.

[14] Chao Peng, Xiangyu Zhang, Gang Yu, Guoming Luo, and Jian Sun. Large kernel mattersimprove semantic segmentation by global convolutional network. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 1743–1751. IEEE, 2017.

[15] Nina Shiva Kassanathan et al. Simple black-box adversarial attacks on deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, pages 6–14, 2017.

[16] Aishy Amer, Amar Mitiche, and Eric Dubois. Reliable and fast structure-oriented video noise estimation. In Image Processing, 2002. Proceedings. 2002 International Conference on, volume 1, pages I–I. IEEE, 2002.

[17] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004.

[18] Hossein Hosseini, Baicen Xiao, and Radha Poovendran. Google’s cloud vision api is not robust to noise. In Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on, pages 101–105. IEEE, 2017.

[19] R. W Floyd. An adaptive algorithm for spatial grey scale. Sid Digest, 17:75–77, 1975.

[20] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531, 2014.

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[25] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[26] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of neural networks through robust optimization. arXiv preprint arXiv:1511.05432, 2015.

[27] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204, 2017.

[28] Xiaoyu Cao and Neil Zhengjiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In Proceedings of the 33rd Annual Computer Security Applications Conference, pages 278–287. ACM, 2017.

[29] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155, 2017.

[30] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pages 3–14. ACM, 2017.

[31] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991, 2017.

[32] Yeongjeong Vorobeychik and Bo Li. Optimal randomized classification in adversarial settings. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, pages 485–492. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[33] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In Security and Privacy (EuroS&P), 2016 IEEE European Symposium on, pages 372–387. IEEE, 2016.

[34] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2574–2582, 2016.

[35] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. arXiv preprint arXiv:1608.04644, 2016.

[36] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. arXiv preprint arXiv:1511.04508, 2015.

[37] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In USENIX Security Symposium, pages 601–618, 2016.

[38] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In Security and Privacy (SP), 2017 IEEE Symposium on, pages 3–18. IEEE, 2017.

[39] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pages 1322–1333. ACM, 2015.

[40] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jesse Lin. Query-efficient black-box adversarial examples. arXiv preprint arXiv:1712.07113, 2017.

[41] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Exploring the space of black-box attacks on deep neural networks. arXiv preprint arXiv:1712.09491, 2017.

[42] Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07379, 2017.

[43] Seong Joon Oh, Max Augustin, Bernt Schiele, and Mario Fritz. Whitening black-box neural networks. arXiv preprint arXiv:1711.01768, 2017.
Xurong Li is currently a Ph.D. student in the College of Computer Science and Technology at Zhejiang University. He received his Bachelor's degree in Information Security from Nanjing University of Posts and Telecommunications. His current research interests include Deep Learning and AI Security.

Shouling Ji is a ZJU 100-Young Professor in the College of Computer Science and Technology at Zhejiang University and a Research Faculty in the School of Electrical and Computer Engineering at Georgia Institute of Technology. He received a Ph.D. in Electrical and Computer Engineering from Georgia Institute of Technology, a Ph.D. in Computer Science from Georgia State University. His current research interests include AI Security, Data-driven Security, Privacy and Data Analytics. He is a member of IEEE and ACM and was the Membership Chair of the IEEE Student Branch at Georgia State (2012-2013).

Meng Han is an assistant professor in College of Computing and Software Engineering at Kennesaw State University. He got his Ph.D. in Computer Science from Georgia State University. His research interests include Big Social Data Mining, Cyber Data Security and Privacy, and Data-driven Intelligence. He is currently an ACM member, an IEEE member, and an IEEE COMSOC member.

Juntao Ji is an Undergraduate Student of Zhejiang University. His current research interests include Adversarial Machine Learning and Security.

Zhenyu Ren is a Postgraduate Student in the College of Computer Science and Technology at Zhejiang University. He received his Bachelor's degree from Central South University. His current research interests include Deep Learning and Security.

Yushan Liu is a Ph.D. student in Department of Electrical Engineering at Princeton University. She received her Bachelor’s degree from Shanghai Jiao Tong University. Her research interests are Network Privacy and Security, Trustworthy Social Networks, and Machine Learning.

Chunming Wu is a Professor in the College of Computer Science and Technology, Zhejiang University. His research interests include Computer Networks and Network Security.