Simple Transparent Adversarial Examples

Jaydeep Borkar*
MIT-IBM Watson AI Lab, IBM Research
jaijborkar@gmail.com

Pin-Yu Chen
MIT-IBM Watson AI Lab, IBM Research
pin-yu.chen@ibm.com

Abstract

There has been a rise in the use of Machine Learning as a Service (MLaaS) Vision APIs as they offer multiple services including pre-built models and algorithms, which otherwise take a huge amount of resources if built from scratch. As these APIs get deployed for high-stakes applications, it's very important that they are robust to different manipulations. Recent works have only focused on typical adversarial attacks while evaluating the robustness of the vision APIs. We propose two new aspects of adversarial image generation methods and evaluate them on the robustness of Google Cloud Vision API's optical character recognition service and object detection APIs deployed in real-world settings such as sightengine.com, picpurify.com, Google Cloud Vision API, and Microsoft Azure's computer vision API. Specifically, we go beyond the conventional "small-noise" adversarial attacks and introduce *secret embedding* and *transparent adversarial examples* as a simpler way to evaluate robustness. These methods are so straighforward that even non-specialists can craft such attacks. As a result, they pose a serious threat where APIs are used for high-stakes applications. 77% of transparent adversarial samples successfully fool the object detection APIs and 90% of images have secret text that fools time-limited humans but is detected by Google Vision API. We also present emerging trends on evaluating the robustness of neural networks and argue that the current research in this field has not focused on simple unconventional methods on evaluating the robustness.

1 Introduction

Deep neural networks have achieved state-of-the-art results on a variety of tasks such as Image recognition [1], speech recognition [2], natural language processing [3], and in games such as [4]. However, these neural networks have found to be vulnerable to adversarial attacks [5], where it is possible to add small imperceptible perturbations to the image that result in misclassification. Since these perturbations are unnoticeable to humans, the perturbed image still looks semantically similar to the original image. These perturbed samples are called as adversarial examples. There have been also adversarial attacks in other domains, such as speech [6], and natural language processing systems [7]. As neural networks get deployed for high-stakes applications, adversarial attacks pose a huge security risks. Such as perturbed stop sign which fools the object detection models[8], breaking copyright detection systems[9], attacking medical deep learning systems[10]. Therefore, it is very important that deep neural network models are evaluated against these attacks to learn if they are robust enough to get deployed in safety-critical areas.

In particular, the current methods of generating adversarial images require the attacker to have specific knowledge about the victim model (e.g. the input gradient used in white-box attack) or excessive number of prediction evaluations (e.g. gradient estimation or substitute model training with model queries in black-box attacks).

*Work done while Jaydeep was an external student at MIT-IBM Watson AI Lab, IBM Research.

In this work, we argue that though the current approaches of generating adversarial examples are important and well studied with respect to a defined threat model (e.g. ℓ_p norm bounded perturbation), they do not provide a complete spectrum of robustness evaluation using semantically similar adversarial examples. Specifically, we demonstrate a set of new and simple manipulations which can be made by anyone even without any machine learning expertise. We further propose a new type of simple attack called *Simple Transparent Adversarial Examples* where anyone without any machine learning expertise can very easily create simple prediction-evasive and semantically similar examples and successfully break neural networks.

As a result, these types of attacks are potentially more dangerous. We hope that this work will motivate the researchers to also consider such simple attacks while evaluating the robustness of deep neural networks.

Along with proposing a simple method of generating adversarial examples, we show that our method is adversarial in two folds: (1) it's prediction evasion (i.e. the adversarial examples evade the deep neural network based object detection) (2) The adversarial examples carry information that is only recognizable by deep neural networks based OCR service but not humans. We test our perturbed examples on Google Cloud Vision API's OCR service[2] where they fool time-limited humans but not deep neural networks based OCR, and on weapon detection APIs sightengine.com[3] and PicPurify[4], and on general object detection APIs Google Cloud Vision API[5], and Microsoft Azure's computer vision API[6] where the perturbed with transparent patches either evade object detection or get misclassified. Figure 1 gives a briefly illustrates *Simple Transparent Adversarial Examples*

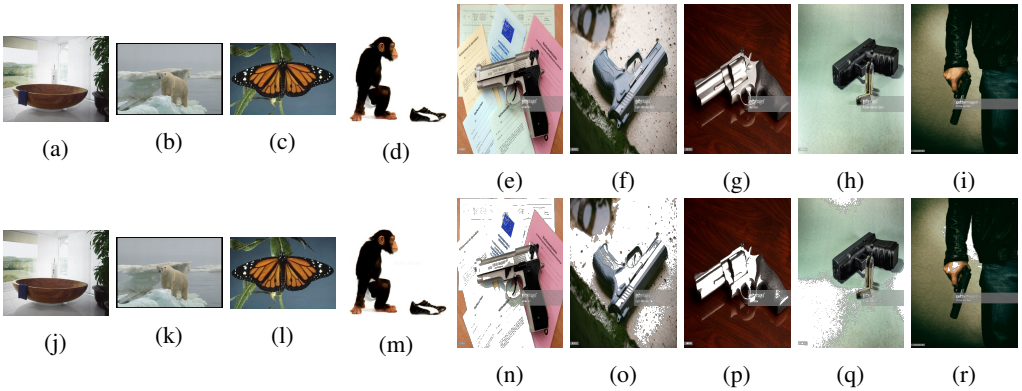


Figure 1: (a) to (i) are original unperturbed images. (j) to (m) are perturbed by adding secret embedding "Hello World" with varying font sizes and font colors. They evade vision of time-limited humans but are detected by Google Cloud Vision's OCR. (n) to (r) are perturbed by adding white transparent patches using an online tool. They fool object detection APIs such as sightengine.com, PicPurify, Google Cloud Vision, and Microsoft Azure's computer vision API either by evading detection or getting misclassified. (p), (q), and (o) are misclassified as packaged goods, camera, and grooming trimmer by Google Cloud Vision API, (n), (o), and (r) evade sightengine.com, (o) and (r) evade PicPurify.

2 Current approaches for Generating Adversarial Examples

In this section, we give a quick high level summary of the current approaches for generating adversarial examples. For the complete description, we encourage the reader to read the original paper.

[2] `https://cloud.google.com/vision/docs/ocr`
[3] `https://sightengine.com/detect-weapons-alcohol-drugs`
[4] `https://www.picpurify.com/demo-gun.html`
[5] `https://cloud.google.com/vision`
[6] `https://azure.microsoft.com/en-in/services/cognitive-services/computer-vision/`

2.1 Gradient-based attacks

2.1.1 L-BFGS

[5] generated adversarial examples using box-constrained L-BFGS. Let x_0 and x denote the original and adversarial examples, respectively, let ℓ denote the correct label. Given an image x_0, their method finds a different image x that is similar to x_0 under L_2 distance, yet is labeled differently by the classifier. They model the problem as a constrained minimization problem:

$$\begin{aligned} &\text{minimize}_x \ \ \|x_0 - x\|_2^2 \\ &\text{such that} \ \ F(x) = l, \ x \in [0,1]^p \end{aligned} \tag{1}$$

2.1.2 Fast Gradient Sign

The fast gradient sign method [11] is optimized for the L_∞ distance metric, produces fast instead of optimal adversarial perturbations. Given an image x_0 the fast gradient sign method sets:

$$x = x_0 - \epsilon \cdot \text{sign}(\nabla \text{loss}_{F,t}(x_0)) \tag{2}$$

where ϵ is chosen to be sufficiently small so as to be undetectable, and t is the target label. Intuitively, for each pixel, the fast gradient sign method uses the gradient of the loss function to determine in which direction the pixel's intensity should be changed (whether it should be increased or decreased) to minimize the loss function; then, it shifts all pixels simultaneously.

2.1.3 PGD attack

PGD attack [12] finds an adversarial example as a constrained optimization by maximizing the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount ϵ. This constraint is usually expressed in terms of L_2 or L_∞ norm of the perturbation and is added so that the perturbed example looks similar to unperturbed example.

2.1.4 JSMA

[13] introduced an attack optimized under L_0 distance known as the Jacobian-based Saliency Map Attack (JSMA). It can be viewed as a greedy attack algorithm that iteratively modifies the most influential pixel for crafting adversarial examples.

2.1.5 Adversarial Patch

This approach is different from the traditional methods to produce adversarial attacks. Here the attack is created by completely replacing a part of the image with a patch [14]. They mask the patch to allow it to take any shape, and then train over a variety of images, applying a random translation, scaling, and rotation on the patch in each image, optimizing using gradient descent. In particular for a given image x, patch p, patch location l, and patch transformations t (e.g. rotations or scaling), they define a *patch application operator A(p, x, l, t)* which first applies the transformations t to the patch p, and then applies the transformed patch p to the image x at location l.

To obtain the trained patch $\hat{p}$ they use a variant of the Expectation over Transformation (EOT) framework of [15]. In particular, the patch is trained to optimize the objective function

$$\hat{p} = \arg\max \mathbb{E}_{x \sim \mathcal{X},\ t \sim \mathcal{T},\ \ell \sim \mathcal{L}}[\log \text{Prob}(\hat{y} | A(p, x, \ell, t))] \tag{3}$$

where X is a training set of images, T is a distribution over transformations of the patch, and L is a distribution over locations in the image.

2.1.6 C&W Attack

Carlini and Wagner propose a new approach to generate adversarial examples [16]. They rely on the initial formulation of adversarial examples [5] and formally define the problem of finding an adversarial instance (with target label t) for an image x as follows:

$$\begin{aligned} &\text{minimize}_x \ \ c \cdot \|x - x_0\|_2^2 + (\max_{k \neq t} Z(x)_k - Z(x)_t)^+ \\ &\text{such that} \ \ x \in [0,1]^p \end{aligned} \tag{4}$$

$(\cdot)^+ = \max \cdot, 0$ and $Z(x)_k$ is the pre-softmax value of class k.

2.2 Black-box attacks

In this attack, the attacker doesn't have any access to the model's gradients or its internal architecture. The attacker relies on the labels given by the model to the input queries to design an adversarial attack. A successful black-box attack might require thousands of queries, so it's not a cost-efficient attack in terms of both time and money. (Papernot et al., 2017) proposed a query-based black box attack where the adversary can only observe the labels given by the DNN to the chosen inputs, train a surrogate model using the queried labels, and perform transfer attack from the surrogate model to the victim model [17]. (Chen et al. 2017) instead used model queries for gradient estimation in zeroth-order optimization and perform direct black-box attacks without training substitute models [18].

2.3 Semantic Adversarial Attacks

(Hosseini and Poovendran, 2018) introduce a new class of adversarial examples called as *semantic adversarial examples*, as images that are arbitrarily perturbed to fool the model, but semantically represent the original objects [19]. They formulate the problem as a constrained optimization problem, and propose a method for generating semantic adversarial images based on the shape bias property of human cognitive system. They first convert the images from RGB into HSV color space, composed of Hue, Saturation and Value color channels. Further, they randomly shift the hue and saturation components, while keeping the value same. This approach generates images that contain the original object with different colors and colorfulness.

3 Simple Transparent Adversarial Examples

In this section, we introduce a new type of adversarial examples called as *Simple Transparent Adversarial Examples*. We illustrate why they are important and how they differ from the current methods of generating adversarial examples.

The current methods of generating adversarial examples require the attacker to have a certain level of machine learning expertise (e.g. first-order or zeroth-order optimization for handling adversarial attacks with perturbation constraints) to craft these attacks. Thus, only the attacker who has the required domain knowledge can attack deep neural networks. Also, majority of the current methods generate adversarial samples that fool only deep neural networks but make no difference to humans, except [20] where the authors create adversarial examples that fool both computer vision and time-limited humans.

We propose a new type of simple attack that doesn't require the attacker to have any machine learning knowledge to attack deep neural network models. Further, the adversarial examples generated by our method fool deep neural networks but not humans (typical adversarial examples), fool time-limited humans but not deep neural networks, and fool both deep neural networks and time-limited humans. We call this type of adversarial examples as *Simple Transparent Adversarial Examples*.

To demonstrate this attack, we focus on attacking machine learning as a service (MLaaS) vision APIs deployed in real-world settings, which provide pre-built deep neural network models to the customers. We test our attack on two of the most popular services offered by the visions APIs: optical character recognition (OCR) and object detection.

Simple Transparent Adversarial Examples have dual definitions in our work:

1. Examples that are straightforward to craft and don't need any machine learning knowledge (i.e. simple and transparent to craft)
2. Examples perturbed with transparent white patches to fool deep neural networks and/or time-limited humans.

We make the following contributions:

1. We propose a simple method of generating adversarial examples called *Simple Transparent Adversarial Examples* which doesn't require any machine learning expertise.
2. We show that our method is adversarial in two folds:
    - It's prediction evasion (i.e. the adversarial samples evade object detection)

- The adversarial examples carry information that is only recognizable by deep neural networks based OCR service but not humans.

3. We show that our simple transparent adversarial examples:
   - Fool deep neural networks but not time-limited humans .
   - Fool time-limited humans but not deep neural networks.
   - Fool both humans and deep neural networks.

3.1 Simple Transparent Adversarial Examples for OCR

Simple Transparent Adversarial Examples for OCR refer to the examples that are simple to craft without any machine learning knowledge. We propose a new method called *secret embedding approach* to craft such type of examples. We evaluate this examples on Google Cloud Vision API's OCR feature. We find that the examples created by this method carry information that is only recognizable by deep neural networks through OCR whereas it evades time-limited human vision. The information is a text embedded in an image in such a way that it evades human vision but is still recognized by OCR. Thus, these examples are the type of adversarial examples that fool time-limited humans but not deep neural networks.

3.1.1 Attack creation

For an image x_0, we find an image x with an embedded text t, such that t evades human vision but is recognized by Google Cloud Vision API. To embed the text, we use an online tool imagecolorpicker.com [7] to find RGB value of the region in which we plan to embed the text, and then we embed the text using an online editing tool Photopea [8]. We use the publicly available online tools to show that such attacks can be easily desgined by anyone, even without any machine learning knowledge. We use Caltech 101 [9] and Caltech-256 [10] dataset for this attack. We propose the following algorithm for *secret embedding approach*:

1. Based on our experiments, we start with a font size of 15 px and a difference of 30 between the *RGB* values of the region in which we want to embed the text and *RGB* values of font color of the text we want to embed. We call this difference as *RGB* difference. Further, we evaluate the embedded image on Google Cloud Vision API's OCR service.
2. If OCR is able to recognize the embedded text, we further keep reducing the *RGB* difference or the font size or both up to the extent till which text can be recognized by OCR but evades the vision of time-limited humans.
3. If OCR is not able to recognize the text, we steadily increase the *RGB* difference keeping the font size as 15 px till the extent it gets recognized by OCR but evades the vision of time-limited humans.
4. If the embedded text in step 2 doesn't evade the vision of time-limited humans, then keeping the difference same, we experiment with font sizes smaller than 15 px. The goal is to get the embedding that is recognized by OCR but evades the vision of time-limited humans.

3.1.2 Observations

We find that in some cases there is a trade-off between the font size of the embedded text and the *RGB* difference. To elaborate, text with higher *RGB* difference and smaller font size (<=15 px) or text with smaller *RGB* difference and relatively larger font size (>15 px) is favorable to craft these adversarial examples. However, smaller *RGB* differnce and smaller font size are the ideal conditions. Figure 2 illustrates this attack. In our experiments, 90% of examples fooled the vision of time-limited humans and were recognized by OCR. However, we also observe that it's possible to achieve near 100% evasion rate in the case of examples that fool both time-limited humans and OCR with secret text of very small font size size and *RGB* difference set to 0.

We further find that this attack is adversarial in two folds:

[7] `imagecolorpicker.com/en/`

[8] `https://www.photopea.com/`

[9] `http://www.vision.caltech.edu/Image_Datasets/Caltech101/`

[10] `https://authors.library.caltech.edu/7694/`

1. It fools time-limited humans but not machines (Google Cloud Vision API's OCR).
2. It fools both time-limited humans and machines (Google Cloud Vision API's OCR) as shown in Figure 3.

(a) Original image (b) Perturbed image (c) Original image (d) Perturbed image

Figure 2: (b) & (d) are perturbed images that fool time-limited humans but not Google Cloud Vision API (i.e. they are recognized by OCR). (b) has an embedded text "Hello World" of font size 11 px inside a rectangular region formed by the x & y coordinates (11, 167), (86, 167), (11, 195), (86, 195) and *RGB* difference 30 and (d) has an embedded text "Hello World" of font size 9 px and *RGB* difference 30 inside a rectangular region formed by the x & y coordinates (90, 62), (147, 62), (90, 82), (147, 82). This adversarial examples fool time-limited humans but not machines.

(a) Original image (b) Perturbed image (c) Original image (d) Perturbed image

Figure 3: (a) and (c) are the original images. (b) and (d) are perturbed in such a way that they fool both time-limited humans as well as machines (Google Cloud Vision's OCR). (b) has a secret embedded text "Hello World" of font size 5 px and *RGB* difference 0 inside a rectangular region formed by the x & y coordinates (4, 13), (27, 13), (4, 28), (27, 28) and (d) has a secret embedded text "Hello World" of font size 5 px and *RGB* difference 0 inside a rectangular region formed by the x & y coordinates (18, 32), (77, 32), (18, 59), (77, 59).

3.1.3 Importance of the attack

The current work on evaluating the robustness of deep neural networks based OCR systems revolves mostly around traditional adversarial examples, such as [21] and [22]. We go beyond the traditional adversarial examples and introduce *Simple Transparent Adversarial Examples* to evaluate the robustness of OCR systems in which attacker introduces secret invisible text that fools time-limited humans but not deep neural networks. As vision APIs offering OCR services get deployed in high-stakes applications and safety-critical areas, such type of attacks can cause a tremendous loss. Moreover, as these type of adversarial examples can be crafted with just online tools that are publicly available to manipulate images by anyone, they pose more security risk than the current typical adversarial examples.

3.1.4 Possible Applications and Risks

We show some ways in which the attacker can use Simple Transparent Adversarial Examples to attack high-stakes applications.

1. **Breaking Blind Review**: anonymized submissions can be broken by adding author names to the figures such that they evade the vision of time-limited humans but are detected by OCR applications.
2. **Breaking Check Scanner Systems**: A lot of check scanner APIs are available in the market that use OCR to process the checks. The attacker can embed a secret text, such as manipulating the amount or the name, which can have a major security threat.

3. **Steganography**: Steganography is the method of hiding secret data within a file to avoid detection. The file is then sent and the secret data is extracted at its destination. *Secret embedding approach* can be used as a simple technique to perform steganography, where image with secret invisible text can sent safely to its destination without the actual content being getting revealed. The attacker can use this technique to send some malicious information or can use OCR to extract important sensitive information that is being sent.

3.2 Simple Transparent Adversarial Examples to Evade Object Detection

Recalling our definition, Simple Transparent Adversarial Examples for object detection refer to the set of examples that are perturbed by introducing transparent white regions in an image and the examples that are straightforward to craft without any machine learning knowledge. Recent adversarial attacks on the object detectors such as [23][24][25][8] require the attacker to have machine learning expertise to carefully design such attacks. We propose a simple attack method where the attacker can add transparent white patches in an image using any publicly available online transparency tool to successfully evade object detection.

3.2.1 Attack creation

For any image x_0, we perturb x_0 by introducing a white transparent patch p, such that the resultant perturbed image x is either misclassified or evades the classification entirely. To demonstrate our attack, we focus on attacking publicly available object detection APIs. To generate adversarial samples, we choose the images having weapons and use them to attack weapon detection APIs such as sightengine.com, PicPurify, and Google Cloud's and Microsoft Azure's computer vision APIs that detect the objects. We use publicly available onlinejpgtools.com's transparency maker tool [11] to introduce white transparent patches in the image. We use a readily available online tool to demonstrate that anyone can easily craft such attacks to attack high-stakes applications, hence these type of attacks pose a higher security risk than the typical adversarial attacks where the attacker is required to have machine learning expertise to design attacks. We use Kaggle's Weapons Dataset [12] to generate adversarial examples. We propose the following algorithm to craft such attacks that evade object detection APIs:

1. We choose a specific percentage of transparency to start with and get our first perturbed sample.
2. We test the perturbed sample against the object detection APIs.
3. If the perturbed sample evades object detection, we go for lesser transparency percentage levels and continue the process till we get the sample with the minimum transparency level that evades object detection.
4. If the perturbed sample doesn't evade object detection, we go for higher transparency level till we get the perturbed sample that evades object detection.

3.2.2 Observations

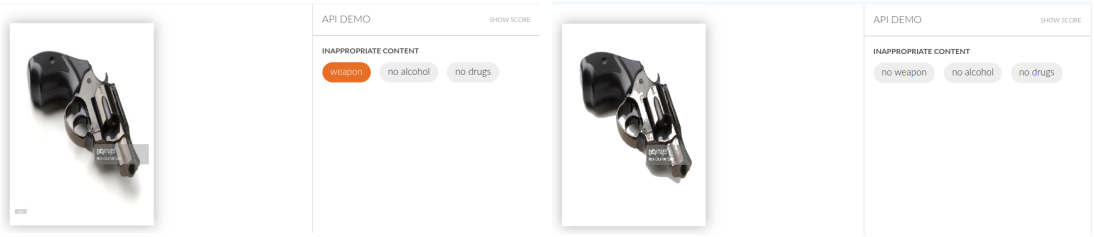
Based on our experiments, we find that our Simple Transparent Adversarial Examples successfully evade the weapon detection by sightengine.com and PicPurify, and object detection by Google Cloud Vision API and Microsoft Azure's computer vision API. Also, in some cases, the perturbed sample results in misclassification by Google Cloud Vision API. In our experiments, 77% of the perturbed examples successfully evade the object detection either by going undetected or getting misclassified. Figure 4 and Figure 5 illustrate this attack.

3.2.3 Importance of this attack

The current attack on Google Cloud Vision API [26][27] are majorly small noise adversarial attacks that require the attacker to have machine learning expertise. We show a simple attack where anyone without any machine learning expertise can generate adversarial examples that evade publicly

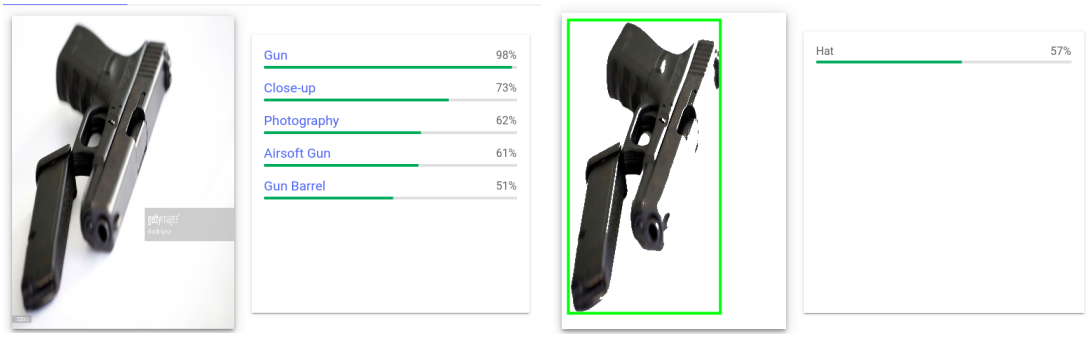
[11] `https://onlinejpgtools.com/make-jpg-transparent`

[12] `https://www.kaggle.com/ar5p1edy/weapons-datasets`



(a) Original image (b) Perturbed image

Figure 4: (a) is the original image having gun that is detected successfully by sightengine.com. (b) is the image perturbed by adding transparent patch, which evades the sightengine.com weapon detection API.



(a) Original image (b) Perturbed image

Figure 5: (a) is the original image that gets detected successfully by Google Cloud Vision API. (b) is perturbed image having 55 % transparency intensity and is misclassified as hat by Google Cloud Vision API.

available computer vision object detection APIs. As a result, they might pose a higher security risk than the current typical adversarial attacks.

Current black-box adversarial attacks [17] are not cost-efficient, both in terms of money and time. They require thousands of queries to design adversarial examples which can be very expensive. In our proposed attack, the attacker can successfully attack object detection APIs with just a couple of attempts by seeing the output and changing the transparency intensity accordingly. Hence, we argue that our black-box attack is cheap and query efficient with a different perturbation constraint.

4 Conclusion

In order to deploy deep neural networks in safety-critical areas and high-stakes applications, it's very important that the robustness evaluation is thorough, efficient, and covers the whole spectrum of semantically similar examples. Lately, there has been a huge surge in the amount of research in terms of creating adversarial attacks and defenses for deep neural networks, which has provided some great insights on evaluating the robustness. Though important, we argue that the current research in this field has not focused on simple unconventional methods on evaluating the robustness. This needs the attention because simple methods can be used any anyone to attack deep neural networks, whereas the current conventional methods can only be used by attacker having a machine learning expertise. Therefore, there's a high chance that attackers might adopt such simple and cheap methods to attack and thus posing a huge security concerns. Specifically, we propose simple transparent adversarial examples and illustrate their novel insights to complement current adversarial example generation pipeline on several image-based APIs. We hope that our unconventional route to highlight such simple attacks will motivate the research in this field to also consider the serious threats posed by such simple attacks to build more inclusive and broad robust defenses for machine learning systems in real-world settings.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally normalized transition-based neural networks," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 2442–2452, Association for Computational Linguistics, Aug. 2016.

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–, Oct. 2017.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[6] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," 2018.

[7] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 2021–2031, Association for Computational Linguistics, Sept. 2017.

[8] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," in *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, (Baltimore, MD), USENIX Association, Aug. 2018.

[9] P. Saadatpanah, A. Shafahi, and T. Goldstein, "Adversarial attacks on copyright detection systems," 2019.

[10] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, "Adversarial Attacks Against Medical Deep Learning Systems," *arXiv e-prints*, p. arXiv:1804.05296, Apr. 2018.

[11] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.

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

[14] T. Brown, D. Mane, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," 2017.

[15] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," vol. 80 of *Proceedings of Machine Learning Research*, (Stockholmsmässan, Stockholm Sweden), pp. 284–293, PMLR, 10–15 Jul 2018.

[16] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, pp. 39–57, 2017.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, (New York, NY, USA), p. 506–519, Association for Computing Machinery, 2017.

[18] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

[19] H. Hosseini and R. Poovendran, "Semantic adversarial examples," 2018.

[20] G. Elsayed, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial examples that fool both computer vision and time-limited humans," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 3910–3920, Curran Associates, Inc., 2018.

[21] C. Song and V. Shmatikov, "Fooling ocr systems with adversarial text images," 2018.

[22] L. Chen and W. Xu, "Attacking optical character recognition (ocr) systems with adversarial watermarks," 2020.

[23] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, "Dpatch: An adversarial patch attack on object detectors," 2019.

[24] Y. Li, X. Bian, M. ching Chang, and S. Lyu, "Exploring the vulnerability of single shot module in object detectors via imperceptible background patches," 2019.

[25] S.-T. Chen, C. Cornelius, J. Martin, and D. H. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," *Lecture Notes in Computer Science*, p. 52–68, 2019.

[26] H. Hosseini, B. Xiao, and R. Poovendran, "Google's cloud vision api is not robust to noise," 2017.

[27] D. Goodman, "Transferability of adversarial examples to attack cloud-based image classifier service," 2020.