DO NOT OVERESTIMATE BLACK-BOX ATTACKS

Han Wu
Department of Physics and Astronomy
University of Southampton
{han.wu}@soton.ac.uk

Sareh Rowlands & Johan Wahlström
Department of Computer Science
University of Exeter
{s.rowlands, j.wahlstrom}@exeter.ac.uk

ABSTRACT

As cloud computing becomes pervasive, deep learning models are deployed on cloud servers and then provided as APIs to end users. However, black-box adversarial attacks can fool image classification models without access to model structure and weights. Recent studies have reported attack success rates of over 95% with fewer than 1,000 queries. Then the question arises: whether black-box attacks have become a real threat against cloud APIs? To shed some light on this, our research indicates that black-box attacks against cloud APIs are not as effective as proposed in research papers due to several common mistakes that overestimate the efficiency of black-box attacks. To avoid similar mistakes, we conduct black-box attacks directly on cloud APIs rather than local models.

1 INTRODUCTION

Image classification models are widely used in real-world applications , often achieving top-5 accuracy exceeding 90%. Cloud-based image classification services, such as Google Cloud Vision, offer pre-trained models as APIs, allowing users to classify images by sending requests to cloud servers. This is useful for IoT devices that lack the computational power to run deep learning models locally.

However, image classification cloud services are vulnerable to black-box adversarial attacks, which generate imperceptible perturbations on input images to mislead classification models. Although prior research has shown that black-box attacks can achieve higher than 95% success rates with only 1,000 queries without access to model structure and weights (Bhambri et al., 2019), most research generates adversarial images **offline** on local models (see Fig.1) rather than **online** on cloud APIs (see Fig. 2), thereby inadvertently exploits information that is unavailable for cloud-based black-box models. The actual efficiency of online black-box attacks against cloud services remains unclear.

Black-box attacks generate adversarial images by sending queries to the target model. Implementing **online** black-box attacks is more challenging because cloud APIs generally have slower response times compared to **offline** attacks against local models. While local models with GPU acceleration can respond to more than 100 queries per second, the typical response time from an API server is 0.5 - 2s per query. Online black-box attacks pose a limited practical threat because generating multiple adversarial images could take several hours. Therefore, online attacks must be both time-efficient and can achieve high success rates. However, previous research often underestimates the time consumption and overestimates the attack success rate.

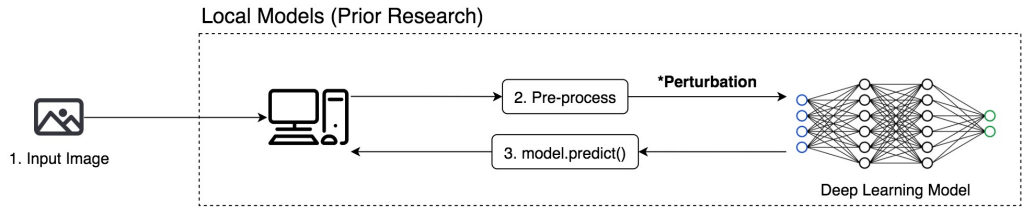


Figure 1: Most prior research tests black-box attacks on local models, where the adversarial perturbation is applied after pre-processing and just before the input is fed into deep learning models, assuming access to the input of a black-box model.

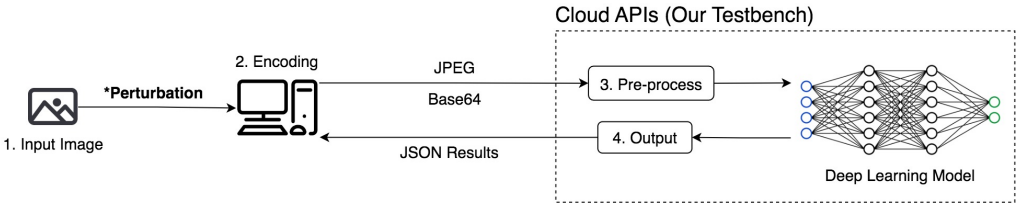


Figure 2: We initiate black-box attacks directly against cloud APIs, applying the adversarial perturbation before image encoding and pre-processing. This approach assumes no access to the internal workflow of cloud-based black-box models.

2 METHODOLOGY

2.1 PROBLEM FORMULATION

Given an input image x and the true label y, the objective of the adversary is to add a small perturbation δ to the original image, and generate an adversarial image $x' = x + \delta$ that can fool a black-box image classifier $C(x)$, such that $C(x') \neq C(x)$. Typically, the perturbation δ is bounded by the l_2 or l_∞ norm with user-defined constants ϵ (Bhambri et al., 2019).

The adversary does not know the model structure and weights. Further, the attacker has limited access to model outputs for cloud-based black-box models. For example, in the partial-information setting, the adversary only has access to the prediction probabilities of the top k classes $\{y_1, ..., y_k\}$. In the label-only setting, the adversary can only access the prediction label without any knowledge of prediction probabilities (Ilyas et al., 2018a).

2.2 COMMON MISTAKES

We observed that some previous research made similar mistakes in the query process, which provided their attacks with an unfair advantage. This advantage led these methods to outperform state-of-the-art black-box attacks, but it was based on the assumption of accessing information that is not available in black-box attacks. These issues are present in several widely-used black-box attacks, including Bandits Attack (Ilyas et al., 2018a;b), SimBA Attack (Guo et al., 2019), Parsimonious Attack (Moon et al., 2019), Square Attack (Andriushchenko et al., 2020) and some recently published research (Liu et al., 2024; Park et al., 2024; Ran et al., 2025). It is important to raise awareness within the research community and prevent similar mistakes in future publications.

2.2.1 IMAGE ENCODING

In real-world scenarios, images are encoded before being sent to cloud services to reduce the amount of data transmitted and save bandwidth. However, prior research, as mentioned above, assumes that perturbations can be added directly to the raw input of deep neural networks (see Fig. 1).

Cloud services such as Google Cloud Vision and Imagga accept raw binary and base64-encoded JPEG images as input. Since JPEG compression is lossy, it may discard some of the perturbations during encoding, thereby reducing the success rate of attacks Dziugaite et al. (2016). Therefore, evaluating black-box attacks on local models without considering image encoding could lead to an overestimation of their effectiveness against real-world cloud APIs.

2.2.2 IMAGE PRE-PROCESSING

Papers listed above apply perturbations after image resizing, implicitly assuming knowledge of the input shape of the black-box model. Moreover, note that original input images are typically larger than the model input shape. Resizing high-resolution images to a lower resolution reduces the sampling space, thereby making it less computationally intensive to generate perturbations.

Besides, image classification cloud services do not accept images with invalid pixel values (pixel value > 255 or < 0). For example, the Bandits Attack does not clip the pixel value of adversarial images, and thus overestimate the attack success rate by sending invalid pixel values to the model.

2.3 POSSIBLE CAUSES

Most prior research tests their attacks on local models rather than on online models because it is both faster and less costly. Sending queries to real-world cloud APIs typically costs around $1 for every 1,000 requests (for example, using Google Cloud Vision). In other words, an experiment attacking 1,000 images and maintaining a query budget of 1,000 queries per image would require 1,000,000 queries and cost $1000.

As a result, most prior research evaluates their attacks on local models and relies on themselves to restrain access to extra model information. However, either intentionally or unintentionally, they exploit extra information that enhances their attacks for the following reasons:

- The PyTorch prediction function only accepts input images x as an array with the same shape. Thus, it is tempting to resize all input images to match the model's required input size, thereby exploiting extra information about the model input shape.
- The PyTorch prediction function accepts input images as floating-point numbers and does not produce an error even if the input image x contains negative values. Consequently, prior research, without considering image encoding, can sent invalid pixel values to the model.

2.4 SOLUTIONS

To avoid these common mistakes, we designed an open-source image classification cloud service, named DeepAPI (see Appendix A), to ensure adversarial perturbations are generated and applied before image encoding and pre-processing.

Additionally, we provide an open-source Black-box Adversarial Toolbox that demonstrates how to conduct online black-box attacks against cloud APIs (see Appendix B), with a focus on practical considerations in real-world scenarios.

3 EXPERIMENTAL RESULTS

3.1 BLACK-BOX ADVERSARIAL ATTACKS

Black-box attacks aim to deceive deep-learning models without having access to their internal structure or weights. We evaluated two common types of black-box attacks: Gradient Estimation and Local Search methods, which have been widely studied in the literature (Bhambri et al., 2019; Wang et al., 2022).

Local Search Methods: The task of generating adversarial inputs can be approached as a problem of selecting what pixels to attack. Thus, we can use existing local search methods to search for combinations of pixels to be perturbed. One simple, yet effective, baseline attack that use this idea is the Simple Black-box Attack (SimBA) (Guo et al., 2019). With SimBA, a vector is randomly sampled from a predefined orthonormal basis and then added or subtracted from the image.

To improve sample efficiency, Andriushchenko et al. proposed the Square Attack (Andriushchenko et al., 2020). This attack initializes the perturbation using vertical stripes because CNNs are sensitive to high-frequency perturbations (Yin et al., 2019), and then generates square-shaped perturbations at random locations to deviate model predictions.

Gradient Estimation Methods: Inspired by white-box attacks that use gradients to generate adversarial perturbations (Goodfellow et al., 2015) (Madry et al., 2017), gradient estimation methods estimate gradients through queries, and then use these estimated gradients to construct adversarial perturbations. To estimate gradients, Chen et al. used the finite-differences method to compute the directional derivative at a local point (Chen et al., 2017). To improve query efficiency, Ilyas et al. proposed a natural evolutionary strategy (NES) based method (Wierstra et al., 2014) to approximate gradients, and proved that the standard least-squares estimator is an optimal solution to the gradient-estimation problem (Ilyas et al., 2018a).

In our experiments, we evaluated the Bandits Attack, which further improved the classifier by using priors on the gradient distribution (Ilyas et al., 2018b), thereby exploiting the fact that the gradients at the current and previous steps are highly correlated.

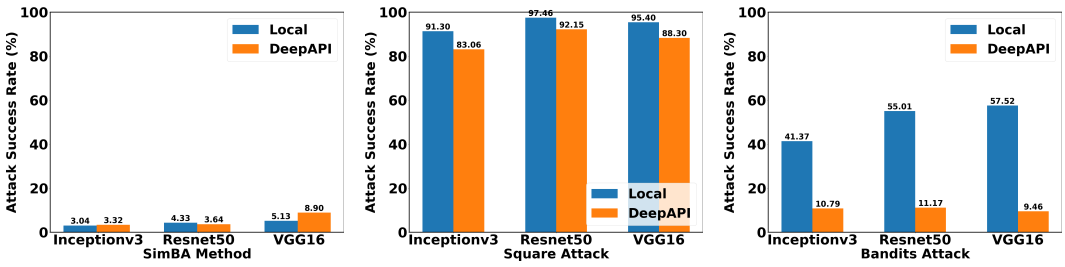
3.2 Attacking Local Model and Cloud APIs

We evaluated three black-box attacks, SimBA, Square Attack, and Bandits Attack, using 1,000 images, each belonging to a unique class in ImageNet. For all attacks, we applied perturbations with a consistent strength of $\epsilon = 0.05$ under the L_∞ norm[1]. Our experimental results reveal that these attacks achieve significantly lower success rates when attacking cloud APIs.

The **SimBA Attack**, a baseline method, achieves comparable low attack success rates and requires similar number of queries for both local models and cloud APIs (see Figs. 3a and 4a). However, it is important to note that the success rate of SimBA is relatively low (approximately 5%), and most attacks exhaust the full query budget (1,000 queries).

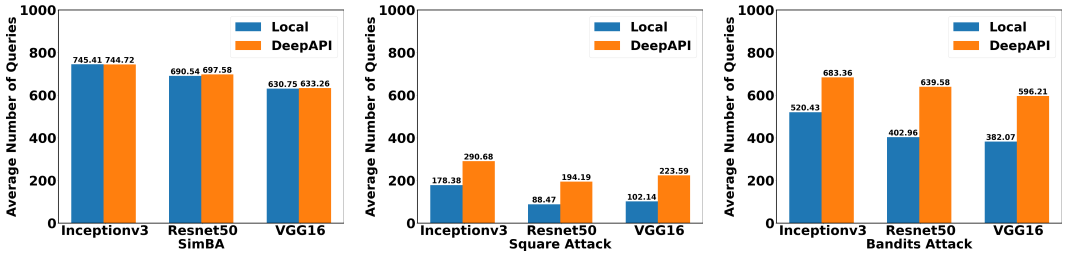
Square Attack, a local search method, applies perturbations to high-resolution images when attacking cloud APIs, resulting in a a lower success rate (Fig. 3b) and requires more queries (Fig. 4b). Due to the absence of image resizing, it is more challenging to find adversarial examples in a larger space, and thus the attack against cloud APIs is less effective Guo et al. (2017).

Bandits Attack, a gradient estimation method, struggles to estimate gradients accurately before image resizing. Bilinear interpolation creates low-resolution images by subsampling from high-resolution inputs, resulting in zero gradients at unsampled points, which makes it difficult to produce valid estimates. As a result, the attack success rate against cloud APIs is significantly lower compared to attacks on local models (Figs. 3c and 4c).



(a) SimBA (Baseline) (b) Square Attack (Local Search) (c) Bandits Attack (Gradients)

Figure 3: The attack success rate of attacking local models and cloud APIs.



(a) SimBA (Baseline) (b) Square Attack (Local Search) (c) Bandits Attack (Gradients)

Figure 4: The average number of queries of attacking local models and cloud APIs.

4 Conclusion

This paper aims to investigate if black-box adversarial attacks have become a practical threat against image classification cloud services. We identify some common mistakes in prior research that leads to an overestimation of the efficiency of black-box attacks.

Additionally, we contribute to the research community by open-sourcing our image classification cloud service, DeepAPI, and Black-box Adversarial Toolbox to facilitate future research on practical black-box attacks against cloud APIs.

[1]Our source code: `https://github.com/wuhanstudio/adversarial-classification/`.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision (ECCV)*, pp. 484–501, 2020.

Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667*, 2019.

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*, pp. 15–26, 2017.

Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning (ICML)*, pp. 2484–2493, 2019.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning (ICML)*, pp. 2137–2146, 2018a.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018b.

Jun Liu, Jiantao Zhou, Jiandian Zeng, and Jinyu Tian. Difattack: Query-efficient black-box adversarial attack via disentangled feature space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3666–3674, 2024.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Victor Millnert and Johan Eker. Holoscale: horizontal and vertical scaling of cloud resources. In *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, pp. 196–205, 2020.

Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pp. 4636–4645, 2019.

Jeonghwan Park, Paul Miller, and Niall McLaughlin. Hard-label based small query black-box adversarial attack. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3986–3995, 2024.

Yu Ran, Ao-Xiang Zhang, Mingjie Li, Weixuan Tang, and Yuan-Gen Wang. Black-box adversarial attacks against image quality assessment models. *Expert Systems with Applications*, 260:125415, 2025.

Chenxu Wang, Ming Zhang, Jinjing Zhao, and Xiaohui Kuang. Black-box adversarial attacks on deep neural networks: A survey. In *2022 4th International Conference on Data Intelligence and Security (ICDIS)*, pp. 88–93. IEEE, 2022.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15(1):949–980, 2014.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32, 2019.

A DEEPAPI

To facilitate future research on black-box attacks that attack cloud APIs rather than local models, we designed DeepAPI [2], an open-source image classification cloud service (see Fig. 5) that supports:

- The three most commonly evaluated classification models in existing research on black-box attacks: VGG16, ResNet50, and Inceptionv3 provided by Keras model zoo.
- Both soft labels (with probabilities) and hard labels (no probabilities) for label-only setting.
- Top k predictions ($k \in \{1, 3, 5, 10\}$) for partial-information setting.

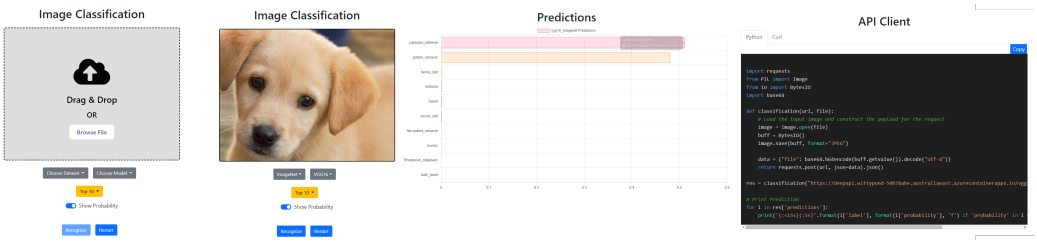


Figure 5: DeepAPI provides both web interface and APIs for research on black-box attacks.

B BLACK-BOX ADVERSARIAL TOOLBOX

To demonstrate how to implement online black-box attacks against cloud APIs, we open-source the Black-box Adversarial Toolbox[3]. To further enhance the practicality of existing black-box attacks, we propose horizontal and vertical distribution strategies (see Fig. 6), inspired by the horizontal and vertical scaling of cloud resources (Millnert & Eker, 2020).

Horizontal Distribution concurrently sends queries for different images within the same iteration, thereby allowing the generation of multiple adversarial examples concurrently. This can be achieved without altering existing black-box attack methods. Since horizontal distribution does not require significant modifications to the original attack method, we can apply horizontal distribution by implementing a distributed query function that sends concurrent requests to cloud APIs.

Vertical Distribution, on the other hand, sends multiple concurrent queries for the same image, thereby accelerating the attack for that particular image. Existing black-box attack methods need to be redesigned to decouple the queries across iterations.

In summary, horizontal distribution achieves concurrent attacks against multiple images, while vertical distribution speeds up attacks on a single image.

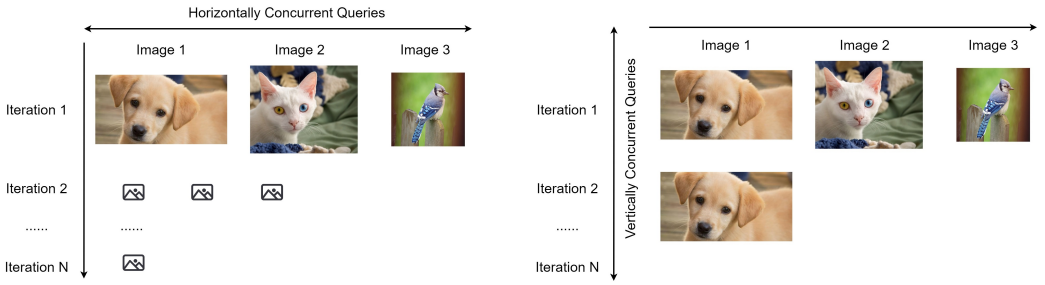


Figure 6: The difference between horizontal and vertical distribution.

[2]The source code of DeepAPI is available on Github: `https://github.com/wuhanstudio/DeepAPI/`.

[3]The source code of the Black-box Adversarial Toolbox is available on GitHub: `https://github.com/wuhanstudio/blackbox-adversarial-toolbox`.