Improving The Decision-Based Adversarial Boundary Attack by Square Masked Movement

Tran Van Sang[1] Tran Phuong Thao[1] Rie Shigetomi Yamaguchi[1] Toshiyuki Nakata[1]

Abstract: Adversarial image attack is a well-known attack methodology in the image recognition field where the input images are purposely modified to make no difference to the human perception but can fool the image recognition models to classify them incorrectly. Recently, the adversarial attack has drawn much attention from researchers due to its ability to fool even state-of-the-art and commercial image recognition models. Researching the adversarial attack is crucial to know the potential risk, thus preparing needed earlier prevention.

In this paper, we investigated an improvement on the Boundary Attack algorithm because of its effectiveness, flexibility and the absent of a direct protection mechanism. Previously, in the randomization step, the Boundary Attack algorithm randomizes the movement vector from the whole image space. In this research, we have improved the algorithm by applying a square mask to the space in this step. We have applied on the CIFAR10 dataset and successfully improved the distance between the adversarial and the original images without increasing the number of queries. Our work suggests a new possibility of an attack vector that can exploit the prior knowledge of the model to improve the distance without affecting the query count.

1. Introduction

Recently, with the rise of computing power, the deep neural network has achieved impressive performance on various tasks, surpassing human ability in many aspects ([13]). Among the deep neural network application domain, image recognition is truly an attractive field thanks to its ample range of utilization. Besides the high accuracy of recent image recognition models on natural input samples, the application of those models poses a security concern when applied in security-or-safety-crucial applications. Researching the attack method is crucial in understanding the potential risk to prepare for necessary prevention ahead of time.

One of the most vulnerable attack vectors is the adversarial image attack([34]), where the input samples are purposely or incidentally slightly modified in a way that makes no change to the human perception but fools the image recognition models to classify them incorrectly. Many adversarial image attack methods ([7], [6], [24], etc.) were investigated and shown that they are efficient against state-of-the-art models or even many commercial models. This implies the urge to give serious attention to this type of attack.

The adversarial attack was initially recognized in ([14]) with the creation of the Fast Gradient Sign Method (FGSM). FGSM is a method in which the target image is iterated by a slight move along the sign of the gradient at the current position. FGSM and the likes are known as white-box attack methods because they take the model's internal information to calculate the gradient precisely. As a result, these methods are not feasible if the model's detail is not available at least in most commercial models. To compensate for that demerit, researchers investigated black-box attack methods in which only the output of the model is required to generate an adversarial image.

1.1 Motivation

We want to investigate the black-box adversarial attacks in image recognition to reveal its capability before continuing with protection mechanism analysis. Out of various black-box techniques, we were motivated by the Boundary Attack method ([5]) because of its outstanding gradient-free concept, along with its accuracy performance, flexibility and the absence of a direct protection mechanism. Boundary Attack is a decision-based method that only requires the label without the associated probability result from the model. Furthermore, it does not rely on the gradient calculation, hence being immune to gradient masking protections such as defensive distillation ([31]). Subsequently, we observed that the original algorithm in the Boundary Attack method, in each iteration, randomizes the next movement vector from the entire space. By scanning the entire space, the algorithm skips any aggregated information of the target model. On the other hand, witnessing that most of the image recognition models detecting the image feature in a square area, in this paper, we proposed an improvement to the Boundary Attack by restricting the movement vector randomization space in a

[1] The University of Tokyo

© 2021 Information Processing Society of Japan

square shape. We have applied our improvement and taken a benchmark on the CIFAR10 dataset([21]). In 80% of 20 uniformly randomized input samples, we have successfully reduced the final l_2 distance between the original images and the adversarially generated images.

1.2 Our contributions

In summary, our main contributions are:

- We successfully reduced the l_2 distance between the original image and the adversarial image, thus improving the original Boundary Attack method.
- Compared to other methods, our proposed method does not introduce additional queries.
- Our method exploits the statistic information over the model, suggesting a new direction for future improvement on the Boundary Attack method.

The rest of this paper is organized as follows. In Section 2, we list several papers that try to improve the Boundary Attack and papers related to space dimension reduction. In Section 3, we explain the detail of our approach. The succeeding Section, Section 4 depicts our experiment environment before discussing the result in 5 section. Finally, Section 6 gives a summarized conclusion and suggests several ideas for the future development.

2. Related Works

The first well-known paper in adversarial image attack research was proposed in [14] in 2014. Goodfellow et al. offered a straightforward yet efficient method called Fast Gradient Sign Method (FGSM) that generates adversarial images by adding a slight vector along the sign of the gradient of the target model. After that, we have witnessed a mass number of papers on adversarial image generation algorithms in diversity approaches. FGSM is one of the approaches requiring knowledge of the internal information of the target model (to obtain the exact gradient). These approaches are classified as white-box attack models.

The white-box attack method is an interesting subject and attracted the attention of many researchers in many works, namely: L-BFGS ([34]), FGSM, Momentum Iterative FGSM (MI-FGSM) ([14]), Basic Iterative Method (BIM) ([23]), Projected Gradient Descent (PGD) ([26]), Carlini & Wagner ([7]) Adversarial Transformation Networks (ATN) ([3]) DeepFool ([27]), Jacobian-based Saliency Map Attack (JSMA) ([30]), Weighted JSMA (WJSMA) ([11]), Taylor JSMA (TJSMA) ([11]), etc. Some of these white-box attack methods achieved surprisingly impressive performances against even state-of-the-art models. However, they all require knowing the target model's internal structure and configuration, causing them to be pointless in practice because, in most available attack scenes, the model's internal details are not provided or can be hidden effortlessly.

Opposite to the white-box attack model, there is an attack method called black-box attack where the attackers do not know the parameters or the structure of the target models but can query the model and obtain the output classes with or without the classification probability. A large number of works extended the concept founded in many white-box attack approaches of moving the input samples along the gradient. Various black-box attack techniques of speculating the model's gradient were developed, for instance: Zeroth Order Optimization (ZOO) ([9]), Autoencoder-Based Zeroth Order Optimization Method (AutoZOOM) ([35]), I. Andrew et al. ([17]), Query-Efficient Hard-Label ([10]), Bhagoji et al. ([4]), input-free attack ([12]), Bandits and Priors A. Ilyas et al. ([18]), etc. Nonetheless, there exist various gradient-masking techniques (e.g., defensive distillation ([31]), making the model flat ([29])) that can be used to protect the model from these gradient-based attack methods by adding non-differentiable terms. Furthermore, these attack methods are ineffective against non-differentiable classifiers, for instance, k-means ([20]).

In 2018, W. Brendel et al. took a creative movement with a different approach called Boundary Attack in [5]. Boundary Attack is a black-box gradient-free attack method. It is classified as a decision-based method and only needs the final label decision from the model, opposed to the score-based method (e.g., N. Narodytska et al. [28], J. Hayes et al. [15]), which relies on the probability associated with the returned label. Besides, Boundary Attack is highly flexible and can be applied with any distance function. This implies a broader application domain of the Boundary Attack method. The idea of Boundary Attack is to start from a uniformly random misclassified image (in targeted attack mode, start from a sample from the target class), iterate the sample by walking around the boundary between the target class and the misclassified class while trying to reduce the distance in each step. The original proposal is very simple by only applying geometrical projection in multidimensional space.

Consequently, Boundary Attack drew researchers' attention via [6], [8], [25], etc. However, *qFool* in [25] and the biased sampling improvement in [6] depend on the gradient calculation. They can be prevented with the gradient masking technique. The other method, called *HopSkipJumpAttack* in [8], uses Binary Search to reach the boundary directly. Nevertheless, *HopSkipJumpAttack* requires additional queries for the binary search procedure. On the other hand, M. Andriushchenko et al. ([2]) recently proposed another gradient-free method called Square Attack. Square Attack is based on the random search with restricting the movement space in a square shape. It is proved to be efficient in some untargeted attack scenes, but the approach is not applicable in a targeted attack.

Through this paper, we propose a method whose idea is similar to Square Attack but applied on the Boundary Attack. Our method does not depend on gradient calculation or negatively affect the query count on the original algorithm.

© 2021 Information Processing Society of Japan

3. Methodology

In this section, we go through several subsections to cover the fundamental understanding which supports our study. Whilst details of the original Boundary Attack algorithm is fully explained in [5], we first briefly summarize it before introducing our proposed improvement.

In this paper, we only discuss the untargeted attack. However, the method can be straightforwardly generalized for the targeted attacks. Similarly, the original algorithm and our modification do not affect the flexibility of distance function selection. In this paper, we only evaluate the l_2 distance function.

Algorithm 1 illustrates the Boundary Attack algorithm. The algorithm starts from an initial random adversarial sample via the function InitialAttack, it then moves the sample toward the target input image by the returned vector from OrthogonalPerturbation if the movement improves the distance and keeps the sample being adversarial.

There are multiple choices of the algorithms for InitialAttack. The selection of the initial attack does not affect our comparison result, thus, any initial attack algorithm can be used. In the algorithm in this paper, an input is uniformly randomized until its label classified by the target model M is not equal to the target label $M(X)$. Subsequently, the Binary Search algorithm is used to find the nearest point toward the target image X in the line connecting X and the randomized sample.

Algorithm 1 Boundary Attack

```
    Input $X$: target input image
    Output $A$: an adversarial image whose distance $\mathrm{d}(A, X)$ is minimal
function BoundaryAttack($X$)
    $A \leftarrow$ InitialAttack($X$)
    repeat
        $\eta \leftarrow$ OrthogonalPerturbation($X, A$)
        if $A + \eta$ is adversarial, and, $\mathrm{d}(A + \eta, X) < \mathrm{d}(A, X)$ then
            $A \leftarrow A + \eta$
        end if
    until $\mathrm{d}(A, X) < \epsilon$, or, number of loops reaches the limit
    return $A$
end function
```

Algorithm 2 Original Orthogonal Perturbation

```
    Input $X$: target input image
    Input $A$: current adversarial image
    Output $\eta$: the next movement
1: function OriginalOrthogonalPerturbation($X$)
2:     $\eta \leftarrow$ sample from Gaussian distribution $\mathcal{N}(0, AdaptiveSize)$
3:     Update $\eta$ by projecting $A + \eta$ onto a sphere around $X$
4:     Update $\eta$ by moving $A + \eta$ nearer toward $X$
5:     Update $\eta$ by clipping $A + \eta$ within the input range $[0, 255]$
6:     return $\eta$
7: end function
```

After the initial attack is initialized with InitialAttack, a perturbation is generated via OrthogonalPerturbation. This function is where we made our optimization for the algorithm. Algorithm 3 describes the original algorithm for the perturbation generation. In the original algorithm, the perturbation is generated from an independent and identically distributed Gaussian distribution $\mathcal{N}(0, AdaptiveSize)$ before being scaled and clipped to make it nearer to the target image and to satisfy the requirement of the image domain. Although this simple heuristic implementation works *surprisingly well*([5]), we noticed that the Gaussian distribution over the whole image space is too large and could drop important statistic information of the model. Therefore, we propose our improvement of the OrthogonalPerturbation which is discussed in the next Subsection.

3.1 Our Proposed Orthogonal Perturbation Algorithm

Observing that in most image recognition models, the image features are usually detected by a square shape filter, we hypothesized that if the perturbation returned from OrthogonalPerturbation is generated within a square shape, we might have a higher chance to seize the image features and the movement might be more robust. Eventually, we proposed an alternative OrthogonalPerturbation algorithm. Our proposed algorithm is outlined in Algorithm 3. Instead of randomizing the whole space (in Algorithm 2, line 2), we restrict the perturbation within a square shape whose corner is uniformly randomized and the square size is determined by the GetSquareSize function (in Algorithm 3, line 2 to line 5). The rest of the algorithm is kept the same.

Algorithm 3 Proposed Orthogonal Perturbation

```
    Input $X$: target input image
    Input $A$: current adversarial image
    Output $\eta$: the next movement
1: function ProposedOrthogonalPerturbation($X$)
2:     $SquareSize \leftarrow$ GetSquareSize
3:     $SquareCornerX \leftarrow$ uniformly random sample from
       $[0, ImageSize - SquareSize]$
4:     $SquareCornerY \leftarrow$ uniformly random sample from
       $[0, ImageSize - SquareSize]$
5:     $\eta \leftarrow$ sample from Gaussian distribution
       $\mathcal{N}(0, AdaptiveSize)$ within the rectangular whose corners are
       $(SquareCornerX, SquareCornerY)$ and $(SquareCornerX +$
       $SquareSize, SquareCornerY + SquareSize)$
6:     Update $\eta$ by projecting $A + \eta$ onto a sphere around $X$
7:     Update $\eta$ by moving $A + \eta$ nearer toward $X$
8:     Update $\eta$ by clipping $A + \eta$ within the input range $[0, 255]$
9:     return $\eta$
10: end function
```

4. Experiment

In the experiment, we built and trained a ResNet ([16])

© 2021 Information Processing Society of Japan

Table 1: Distance between the target image and the adversarial image by the original and proposed algorithms. Difference: positive means better

Sample Number	Original algorithm	Proposed algorithm	Difference	Percentage (%)
1	0.329	0.299	0.030	9.108
2	0.164	0.140	0.024	14.879
3	0.632	0.567	0.065	10.220
4	0.582	0.486	0.096	16.565
5	0.518	0.458	0.060	11.637
6	0.503	0.422	0.081	16.158
7	0.007	0.006	0.001	13.825
8	1.312	1.612	−0.301	−22.947
9	0.397	0.828	−0.431	−108.619
10	0.751	0.681	0.070	9.330
11	0.891	0.924	−0.034	−3.765
12	0.549	0.353	0.196	35.655
13	1.123	1.033	0.090	8.039
14	0.470	0.335	0.135	28.750
15	0.943	0.890	0.053	5.634
16	0.329	0.280	0.049	14.985
17	0.454	0.364	0.090	19.768
18	0.526	0.505	0.021	4.032
19	0.063	0.061	0.002	2.416
20	0.973	0.986	−0.013	−1.335

model on the CIFAR10 ([22]) dataset using the TensorFlow framework ([1]). We followed the implementation of ResNet from [19]. The trained model is then used as the target model. 20 images were uniformly selected from the CIFAR10 dataset to be employed for the untargeted attack measurement.

We followed the implementation provided in [32] for the main algorithm (Algorithm 1), in order to compare the result. We set the maximal number of iterations of the loop at line 8 of Algorithm 1 be 5000.

The implementation of the InitialAttack function, is picked from [19], an implementation for the HopSkipJumpAttack algorithm([8]). Besides, to achieve a fair comparison, we store the initial adversarial point for each target image, and restore them in each run.

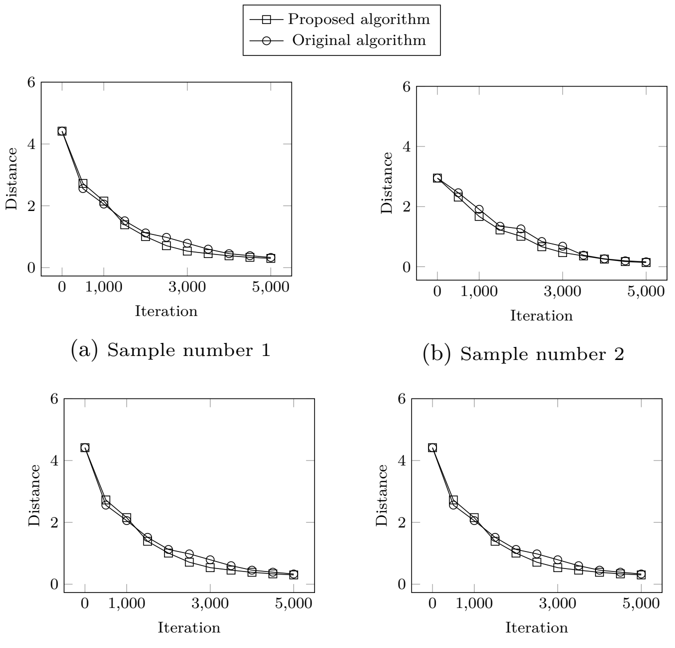
Accordingly, the square sizes returned by the GetSquareSize are 32, 28, 24, 20, 16, 4, 2 for the iteration number from range [0, 50], [51, 200], [201, 500], [501, 2000], [2001, 3000], [3001, 4000], [4001, 5000], respectively.

5. Results and Discussion

In this section, we show and discuss the result of our experiment.

Table 1 displays the numeric result of our experiment. The second and third columns depicts the distance between the final adversarial image and the target image in the original algorithm and our proposed algorithm, respectively. The fourth column (*Difference*) is the difference, it indicates the improvement made by our proposed algorithm, followed by the last column (*Percentage*) representing its ratio to the distance produced by the original algorithm. From the table, we figure out that our proposal improves the Boundary Attack algorithm in 16 cases over total 20 cases.

We recorded the distance changes in each iteration and plotted them in Figure 1 and Figure 2. Figure 1 draws the changes of distances between the target image and the adversarial image in each iteration, of the first 4 samples. Figure 2 draws the same measurement for the 4 negative samples (samples numbered 8, 9, 11, and 20). In the majority of the cases, our algorithm improves the distances. Especially, the distances was improved more in the middle of the loop, around steps between 2000 and 4000. This indicates that there is possibility to tune our algorithm's parameters such as the square size strategy in GetSquareSize.



(a) Sample number 1

(b) Sample number 2

(c) Sample number 3

(d) Sample number 4

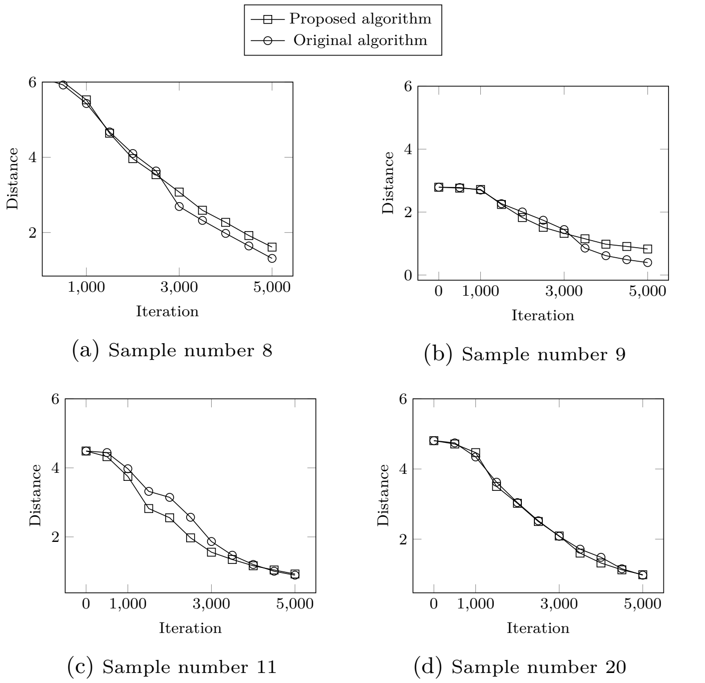
Fig. 1: Distances between the target image and the adversarial image by iteration (lower is better)

On the other hand, Figure 3 combines and visualizes the results of all 20 samples in our experiment. In Figure 3, there are 20 groups of 3 images, associated with 20 samples in our experiment, arranged from left to right, top to bottom. In each group of 3 images, the left image is the initial image (i. e., the result from InitialAttack) the middle image is the final image returned by our algorithm, the right image is the target image. The demonstration video of the algorithm's progress is also available in [33]. The similarity of the middle and the right images indicates that our algorithm produces an acceptable result, even in the negative cases (cases numbered 8, 9, 11, and 20).

6. Conclusion and Future Works

We have improved the Boundary Attack algorithm by introducing a new strategy of randomizing the movement vectors. In which, we restricted the randomization space to a square shape helping the algorithm to capture more features from the image. Via this approach, we made use of the useful statistic information over the image recognition models. Hence, we improved the algorithm without carrying out any additional queries. This opens a very

© 2021 Information Processing Society of Japan



(a) Sample number 8

(b) Sample number 9

(c) Sample number 11

(d) Sample number 20

Fig. 2: Distances between the target image and the adversarial image by iteration (negative cases) (lower is better)

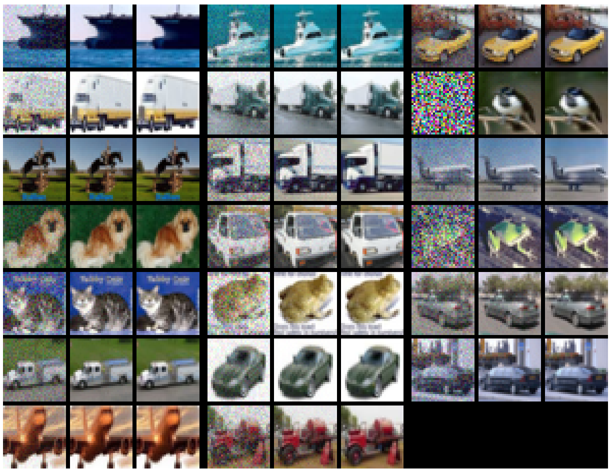


Fig. 3: Result images of the attack algorithm. For each group 3 horizontal images, left: initial image, center: final adversarial image, right: original image

promising opportunity of future improvement such as using other patterns (like oval shapes, triangular, etc.), or investigating more efficient square sizing strategies, etc.

Besides, we conducted a comparison experiment on 20 samples from the CIFAR10 dataset and observed that our modification improved the algorithm in 16 over 20 cases. The similarity of the generated image and the target image in the visualization result indicated that our algorithm produced acceptable result even in negative cases. In the future, we will investigate the experiment on more samples, also considering other datasets. Especially, we are going to evaluate other Boundary Attack-based methods and make the comparisons. Moreover, in the experiment we observed that the time for the adversarial image generation was quite long. In the next study, we want to improve the speed performance of the algorithm by various methods such as migrating the calculation from CPU to GPGPU, parallelizing the calculation whenever possible, etc.

Via this work, we can understand the possibility of the adversarial attack via boundary movement techniques. This is an important step for us to continue with the investigation on protection, preventing image recognition models from the similar risks.

7. Acknowledgement

The authors would like to thank Doctor Irvan Ipan and other the members at Yamaguchi Laboratory, Social ICT Center, the University of Tokyo, for many constructive discussions during our research.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search, 2020.

[3] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples, 2017.

[4] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–169, 2018.

[5] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, 2018.

[6] Thomas Brunner, Frederik Diehl, Michael Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. pages 4957–4965, 10 2019.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[8] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack.

© 2021 Information Processing Society of Japan

In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294, 2020.

[9] 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*, AISec '17, page 15–26, New York, NY, USA, 2017. Association for Computing Machinery.

[10] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack:an optimization-based approach, 07 2018.

[11] Théo Combey, António Loison, Maxime Faucher, and Hatem Hajri. Probabilistic jacobian-based saliency maps attacks, 2020.

[12] Yali Du, Meng Fang, Jinfeng Yi, Jun Cheng, and Dacheng Tao. Towards query efficient black-box attacks: An input-free perspective. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, pages 13–24, 2018.

[13] Robert Geirhos, David HJ Janssen, Heiko H Schütt, Jonas Rauber, Matthias Bethge, and Felix A Wichmann. Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*, 2017.

[14] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[15] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[17] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2137–2146. PMLR, 10–15 Jul 2018.

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

[19] Jianbo-Lab. Hopskipjumpattack. `https://github.com/Jianbo-Lab/HSJA/tree/daecd5c7055d5214b39c34a7a28a98acd3557fbc`, 2019.

[20] K Krishna and M Narasimha Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.

[21] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[22] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. 07 2016.

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

[25] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4890–4898, 2019.

[26] 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.

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016.

[28] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks, 2016.

[29] Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks, 2017.

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

[31] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks, 2016.

[32] Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *Journal of Open Source Software*, 5(53):2607, 2020.

[33] Tran Van Sang, Tran Phuong Thao, Rie Shigetomi Yamaguchi, and Nakata Toshiyuki. Boundary attack step by step movement https://youtu.be/u20wtatp6wk.

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

[35] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

© 2021 Information Processing Society of Japan