Saliency Map Guided Scoring-Based Adversarial Black Box Attack

Gülsüm Tuba Çibuk Girgin
Department of Computer Engineering
Bogazici University
Istanbul, Turkey, 34342
gulsum.cibuk@boun.edu.tr

Emre Girgin
Department of Computer Engineering
Bogazici University
Istanbul, Turkey, 34342
emre.girgin@boun.edu.tr

Abstract

Adversarial samples can easily fool the neural networks. The white box adversarial attacks where model parameters are known by the attacker are unrealistic.In contrast, black-box attacks are more suitable for real-world applications since model parameters are unknown in most cases. We propose a scoring-based black-box attack that forms adversarial images by optimizing the continuous confidence level of the network output with the guidance of the saliency map. The adjusted features of the input are selected not randomly but based on a saliency map that indicates each feature's foreground level and provides extra query efficiency. The saliency map is constructed in an unsupervised manner to achieve a class agnostic guidance. We benchmarked our approach on CIFAR-10 and MNIST datasets to be able to compare it with previous works. Our work is publicly shared[1].

1 Introduction

Deep neural network models have achieved marvelous accomplishments for an enormous number of tasks throughout the decade. Their non-linear and highly complex nature revealed the ability to represent the sophisticated structure of huge data piles. However, deep neural networks are brittle. Recent studies indicate that their data representation is over-sensitive to the carefully selected small perturbations on the input.[20] These fabricated adversary input images can fool deep convolutional neural networks [5, 14] by making them predict a particular target class, whereas the groundtruth is different[20]. Since neural networks are deployed to many safety prioritized real-world applications in domains such as aviation and autonomous vehicles, their robustness to such phenomena becomes more and more critical every day.

The objective of the deep neural networks is to minimize their loss functions to predict the label of test images with high accuracy and confidence. On the other hand, **adversarial attacks** try to fabricate new images to maximize the loss of the deep neural networks for the true labels. These changes in the images should not be noticeable by the human eye. Suppose the deformation on an image is too large, and the image itself can not be perceived as the original label anymore. In that case, it is not considered an adversarial sample. Therefore a limit is set for the change in the pixels of an image.

$$\max_{\delta} loss_y(x+\delta) \text{ s.t. } ||\delta|| < \epsilon \tag{1}$$

$$LP_{ai}(t) = \frac{\Sigma_{j=1}^{s} loss_{ai}(t-j)}{s} - \frac{\Sigma_{j=1}^{s} loss_{ai}(t-j-time)}{s} \tag{2}$$

[1] Our work can be found in saliencySimBA

Eqn. 1 is from [7]. δ is the small perturbation made to an image which has an upper limit ϵ. The distance metric can be l_0, l_1, or l_2 ball. $loss_y$ is the exact optimization function of the model to be attacked or a substitute one.

In white-box attacks, it is assumed that the model's current weights and optimization function are reachable. Therefore the exact optimization function can be written into the maximization problem of the adversarial attack model. On the contrary, **in black-box setting**, the parameters of the model are not known. Such cases can be encountered more in the real world. With an image query, the predicted label and the confidence level of the prediction can be acquired depending on whether partial or complete information is available. With the information gathered, an attack algorithm is used to create an adversary image and lower the confidence level of the model's prediction. Another critical constraint of the black-box attacks other than insufficient information is the limited number of queries. The limitation of the queries is caused by the time and monetary constraints. Therefore it is aimed to find an efficient adversary image with the least number of queries.

Since each feature's contribution is not the same to the output of the neural network, the prediction is more sensitive to some of the pixels than others. Therefore, the success of the adversarial attack depends on not only the perturbation amount but also its position. Su et al.[19] demonstrated that attacks can be performed by just changing one pixel of the input image. Therefore, the success of the attack depends not only on the amount of perturbation but also on the pixel of interests of the target network strictly. This phenomena can be visualized by existing methods such as class activation map[24] or **saliency map** [23].

Black-box adversarial attacks aim to model the effect of features on the output by changing the value of a pixel and observing the confidence score of the prediction. Most of the works select those pixels randomly since there is no prior information about the model and the dataset it is trained with. However, a saliency map can be exploited as guidance for feature selection to increase the success probability of the attack and decrease the number of queries required to form an adversarial sample.

To our best knowledge, this study is the first one that utilizes a saliency map as the feature selection guide for a black-box adversarial attack. The Related Work section presents a comprehensive overview of the black box attacks and saliency detection methods. In the Methodology section, we explain the details of the proposed method. In the Experiments section, we show some of our experiments' qualitative and quantitative results whose findings support our proposition. The last section wraps up our proposed method and concludes our study.

2 Related Work

2.1 Black Box Adversarial Attacks

Black box adversarial attacks are agnostic to model's parameters they attack. Black box adversarial attacks can be grouped into three categories:[11, 1] Decision-based attacks[1, 10] (1) conducts a local search in the input space by just the guidance of the final predicted class of the input sample. Although these types of attacks are most suitable for real-world scenarios, they require a way more number of queries (calls) to produce consistent results. [11] Besides, their performance is restricted to the dimension and directions of the search. [21] Scoring-based attacks [7, 10] (2) have access to the probabilities of each class. This property provides more detailed feedback during the local search. Transfer-based attacks [18] (3) require knowing the dataset that the model is trained on, so a substitute model can be trained and attacked. These types of attacks are heavily sensitive to the transferability properties of both target and substitute models, which may not be guaranteed all the time. Also, there are other kinds of attacks [17, 10] relying on the gradient estimation of the target network.

Since black-box attacks can only reach the output of the network, adversarial samples are formed in a trial-and-error manner. At each run, the attack gathers a piece of information about the way model behaves and takes appropriate action. However, this mechanism requires lots of feed-forward queries, and each query has a cost in the real world applications. Therefore, query efficiency is a key factor when assessing the performance of the black-box adversarial attacks. Each query may subject to time and monetary constraints since the attacked network is accessible through a paid API[2], in general.

[2]Popular ones are Google Cloud Vision API, Amazon Machine Learning and Clarifai.

Guo et al.[7] proposed a simple black-box adversarial attack called *SimBA* that searches the orthogonal directions of the input space. The algorithm randomly selects orthogonal basis of the input and takes steps towards these vectors with an ϵ amount. If the confidence level of the output does not decrease, the algorithm takes this step back and moves in the opposite direction.

The attack is conducted on various basis. The randomly selected pixels are utilized as the orthogonal basis directions in the cartesian basis. They also investigated the frequency space by applying discrete cosine transform and selected a subset of the directions with the lowest frequencies. However, they reported that the performance of these two basis is quite similar.

Wang et al. [21] designed a multi-objective evolutionary algorithm to produce adversarial samples with the guidance of the class activation map (CAM)[24]. They trained a proxy model that estimates the activation map and produced a noise to be added to the original input. However, this proxy model is trained on the same dataset, and the problem turns into a gradient estimation attack.

2.2 Saliency Detection

Saliency map creation is a well-studied area of research. Historically, it is done via hand-crafted features [16, 9, 2, 4], and these methods are still the most well-known unsupervised methods.[22] However, their outputs are extremely noisy and unsuitable for consideration as groundtruth solely. On the other hand, supervised approaches [8, 13] are promising and exhibit plausible results. However, they require human supervised datasets and are not useful when a data distribution shift is present. Therefore, supervised methods are not sufficient to be used as guidance for black-box attacks if we consider the unknown properties of the model to be attacked, such as the dataset it is trained on.

Hou et al. [9] proposed a simple model to calculate the saliency of a given image by analyzing the log-spectrum properties without any prior knowledge. They first proposed that many natural images contain the same redundant information. Then, the attention-seeking part, called spectral residual, can be popped out and be interpreted as a saliency map by removing the redundant part. However, the output of this technique is extremely noisy and is not reliable for absolute guidance.

Gu et al. [6] proposed a guided backpropagation algorithm to partially recover the image as saliency maps to explore the classification of adversary decisions of the neural network. Etmann et al. [3] aligned the saliency maps of the input images while feeding them to the neural network to train more robust classifiers. Besides, Mangla et al. [15] showed the correlation between detailed saliency maps and the robustness of the neural network against adversarial attacks.

3 Methodology

SimBA [7] does a local search on the input space by randomly selecting pixels. We enhance this algorithm by guiding the feature selection phase with the supervision of a saliency map. The saliency map will be obtained by a hand-crafted manner. We also investigate the common properties of the successful attacks and both the probabilistic and the guided feature selection methods.

3.1 SimBA Attack

After training the image classifiers, we integrated the vanilla SimBA [7] into the network. The attack picks a pixel and adds perturbation to it. The softmax output of the last layer is fed back to the attack algorithm so that it can investigate the directions leading to an adversarial sample.

3.2 Probabilistic Pixel Selection

The vanilla SimBA selects pixel coordinates to be attacked from a uniform distribution. (Eqn. 3) However, the center of the image has more distinctive features than the sides since most images are dominated by the background. Hence, we slightly adjusted the SimBA algorithm to sample the pixel coordinates from Standard-Gaussian distribution. (Eqn. 4) Then the sampled values are scaled to be consistent with the image size.

$$u,\ v \sim \mathcal{U}(0,1) \tag{3}$$

$$u,\ v \sim \mathcal{N}(0,1) \tag{4}$$

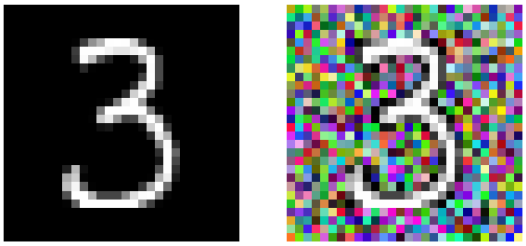

Figure 1: From original MNIST to modified MNIST

3.3 Hand-Crafted Saliency Maps

One of the prevalent and classic approaches to create saliency maps is the *spectral residual* method[9]. We exploit this method as the baseline saliency integration. The resulting saliency map is normalized. The normalized pixel values are considered as a probability distribution, and pixels to be attacked are sampled based upon that.

$$(u,\ v) \sim \mathcal{S}\text{aliency}(X) \tag{5}$$

3.4 MNIST Saliency Maps

In order to measure the performance of the guided pixel selection, the probability distribution that is sampled must be approximated perfectly. In other words, the success of the guided pixel selection strictly depends on the ability to represent the distribution of the pixels sampled, which is the saliency map in our case. Otherwise, our assumption that the foreground is more important than the background can not be proven because of the poor distinction of pixels. MNIST dataset consists of grayscale images with white-colored digits. This property makes the sample itself a saliency map.

4 Experiments

In this section, we explain the details and the results of the experiments. The whole setup is composed of several parts. In the first part, we explain the classifiers we trained. In the remaining parts, we evaluate the results of the attacks in different aspects.

All the pipeline is implemented in Python, and as the deep learning framework, PyTorch[3] is utilized. The training is done on a single RTX3060 GPU. Each experiment took approximately half a day and an hour on the CIFAR-10 and MNIST testsets, respectively. The details of the experiments are explained in the following subsections.

4.1 MNIST Dataset Preparation

MNIST handwritten digits is a lightweight dataset with only one channel, and the background pixel values are all zero. These properties make each sample itself a perfect saliency map. However, we changed the background of the image with random noise to make the dataset more challenging, whereas the natural saliency map is kept. Figure 1 shows the transformation from an original MNIST sample to a modified one.

4.2 Training

We trained different classifiers for different datasets, as shown in Table 2. The training hyperparameters of the processes can be found in Table 3. As an optimizer, Adam[12] is utilized, and the learning rate is scheduled. The accuracies in the test sets are also shown in Table 2.

[3] https://pytorch.org/

Max Queries	Perturbation Size
500	$\pm 16/255$

Table 1: Attack hyperparameters.

Classifier Name	Dataset	Test Accuracy
Resnet50	CIFAR-10 Train Dataset	86%
4 Layered Perceptron	MNIST Train Dataset	98%
4 Layered Perceptron	Modified MNIST Train Dataset	92%

Table 2: Classifiers trained on different datasets.

4.3 Attack

For each classifier, we adopted SimBA attack with different sampling strategies to decrease the accuracy of the classifier and the average number of queries. We set the hyper-parameters of each attack as shown in Table 1. We constrained the maximum number of queries to decrease the time needed to run the algorithm. The maximum number of queries denotes that the algorithm skips that sample after how many queries. The maximum number of queries is **500** and perturbation size is $\pm 16/255$.

4.3.1 Attack on CIFAR-10

We choose the attack images from the CIFAR-10 test dataset for the first classifier. In the first trial, we used the vanilla SimBA algorithm as a baseline, where features are sampled from uniform distribution (Eqn. 3). In second trial, we changed the pixel sampling part of the SimBA with standard Gaussian (Eqn. 4). In the last trial, pixels are sampled using spectral residual[9] hand-crafted saliency map. (Eqn. 5)

4.3.2 Attack on MNIST

We choose the attack images from the MNIST test dataset and its modified counterpart for the last two classifiers.(See Figure 1) We used the vanilla SimBA algorithm as a baseline in the first trial, where features were sampled from the uniform distribution (Eqn. 3). In the second trial, we only sampled pixels from the foreground. In the last trial, pixels are sampled from the background.

4.4 Attack Success Rate

For each sample in the test set, the SimBA [7] attack is conducted on the sample, and the prediction of the network is noted. However, the way the target pixel is sampled differs. Note that we skip a sample if it is misclassified initially.

4.4.1 Attack Success on CIFAR10

The results of the attack on CIFAR10 testset are shown in Table 4. We can see that the Gaussian-Sampling method is the most successful one among other methods.

In Figure 2, some of the successful attacks of the Gaussian-Sampling method are presented. The noise is upscaled 150 times to make them more observable. The perturbated pixels are distributed around the center of the image. We can explain it as the centers of the images are more important than the edges (background) of the images for the classifier. Therefore, attacking the center pixels helped us to fool the classifier more.

The saliency sampling did not perform well, contrary to our expectations. If we inspect the average noise added to the images per method, as shown in Figure 3, we can see that some of the pixels of the

Epochs	Batch Size	Learning Rate	LR. Decay Rate
25	128	0.001	0.9

Table 3: Training hyperparameters for all experiments.

Effect of methods on accuracy		
Method	Accuracy (lower is better)	Attack Success Rate (higher is better)
Original Training	86%	*Not Defined*
Vanilla SimBA	35%	59%
Gaussian-Sampling	**34%**	**60%**
Saliency-Sampling	35%	58%

Table 4: Success rate of each method on CIFAR10 Dataset.

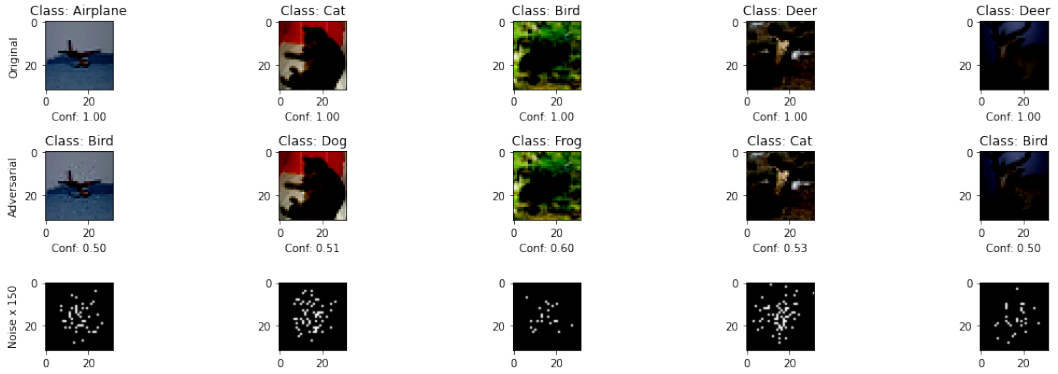

Figure 2: Example successful standard Gaussian attacks on CIFAR10 Dataset.

saliency sampled noise are accumulated on edges. While we are evaluating the results of Gaussian sampling, we claim that the centers of the images are more important than the edges. Therefore, looking at the average noise of the saliency-sampled pixels, we can understand why it performed worse. As a result, we could not detect the objects in the images with the hand-crafted saliency map, showing that the hand-crafted saliency detection method[9] is the bottleneck. We fixed this issue thanks to the properties of the MNIST dataset in the following experiment. (See Section 4.4.2)

Attack Success Based on Confidence		
Method	Initial Confidence	Last Confidence
Vanilla SimBA	98%	52%
Gaussian-Sampling	98%	52%
Saliency-Sampling	98%	52%

Table 5: Average confidence change of the network on the fooled samples on CIFAR10 Dataset.

We also investigated the change in the confidence level of the network in the samples where the attack was successful. However, we observed no differences among the methods, as shown in Table 5. However, the statistics of the unsuccessful samples are also interesting. For all of the samples that

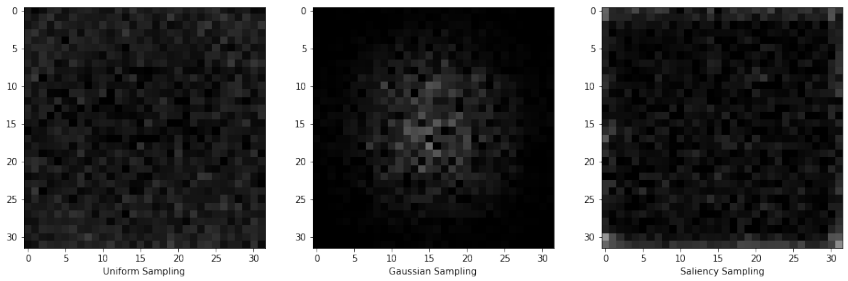

Figure 3: Average noise of the successful attacks per method on CIFAR10 Dataset.

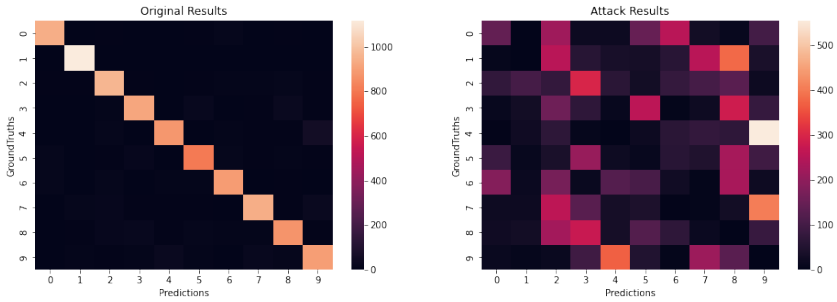

Figure 4: Numbers of predicted labels and their groundtruths of the Foreground-Sampling method are shown as a heatmap on modified MNIST.

the attack was unsuccessful, the network had 100% confidence in the true label, and the attack could not decrease the confidence for any of them. We interpret that the attack algorithm exploits the small confidence gaps of the network and applies local search in that direction. However, if the network is 100% sure, the algorithm generally fails.

4.4.2 Attack Success on MNIST

In this section, the results on both original and the modified MNIST datasets are demonstrated. Very different behaviors are observed between datasets. In Table 6, the huge performance gap between saliency-guided attacks and others is shown. The foreground sampling was able to fool nearly all of the testset. However, another interesting point is the performance difference between vanilla SimBA and background sampling. When the pixels to be attacked are sampled from the background on purpose, the success rate was worse than uniform sampling. The uniform sampling is slightly better because it can sample pixels from the foreground.(See Figure5) As a result, it can be clearly said that the classifier nearly does not give any attention to the background of the modified MNIST dataset since it is composed of pure noise. Also, the gaussian sampling done in the previous experiment displays a similar behavior.

Effect of methods on accuracy		
Method	Accuracy (lower is better)	Attack Success Rate (higher is better)
Original Training	92%	*Not Defined*
Vanilla SimBA	27%	71%
Foreground-Sampling	**3%**	**96%**
Background-Sampling	40%	57%

Table 6: Success rate of each method on modified MNIST Dataset.

The last experiments are conducted on the original MNIST dataset. As shown in Table 7, any sampling method could not fool the network. This is caused by the simple structure of the MNIST samples. The dataset distribution is easier to learn for a non-convex optimizer, making it more robust to any perturbation or noise. However, there is still a gap where the attack algorithm was able to fool towards that direction, but it is limited. Moreover, since all of the sampling methods had the same success rate, it can be clearly said that the original MNIST classifier is not more sensitive to the digit itself but the whole image instead. This may be considered as a problem because we, as humans, perceive the digit by focusing on the foreground object. This shows that the classifier learned to distinguish digits in a way that is different from what humans do.

4.5 Query Efficiency

4.5.1 Query Efficiency on CIFAR10

The total number of queries is limited to 500, as shown in Table 1. We also investigated whether any of these methods provide query efficiency. It is observed that the Gaussian-Sampling provides

Effect of methods on accuracy		
Method	Accuracy (lower is better)	Attack Success Rate (higher is better)
Original Training	99%	*Not Defined*
Vanilla SimBA	90%	8%
Foreground-Sampling	**90%**	**8%**
Background-Sampling	90%	8%

Table 7: Success rate of each method on original MNIST Dataset.

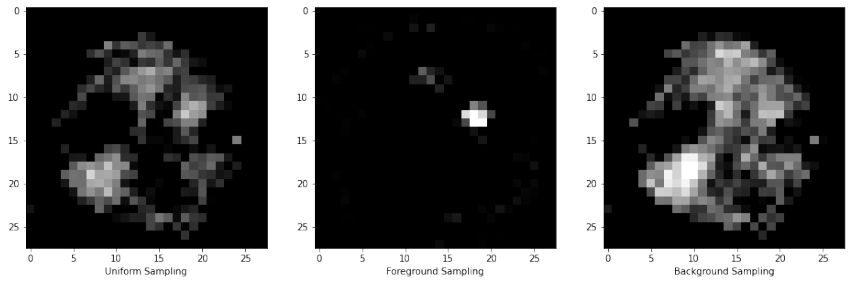

Figure 5: Average noise of the successful attacks per method on modified MNIST Dataset. The uniform and background sampling methods are only successful when the selected pixel is around the foreground.

a higher attack success rate (See Table 4) and query efficiency compared to the Vanilla SimBA, as shown in Table 8. We can say that the centers of the images carry more information than the edges (background) of the images for the classifier. Therefore, a change in a center pixel affects the confidence more. Therefore we can fool the classifier with a less number of queries.

Query Statistics for each method		
Method	Avg. Num. Queries (lower is better)	Median of Queries (lower is better)
Vanilla SimBA	135.9	114
Gaussian-Sampling	**111.3**	**86**
Saliency-Sampling	134.3	112

Table 8: Query statistics for each method on CIFAR10 Dataset.

4.5.2 Query Efficiency on MNIST

For the MNIST dataset, we also limit the maximum number of queries to 500. It is observed that the saliency-guided attacks have fooled the network with less number of queries on average with a wide margin. (See Table 9, Figure 6). By adding noise to the background of the MNIST samples, we forced the classifier to focus more on the digit itself, which is the main part carrying the discriminative features between classes. Therefore it becomes more and more meaningful to search in the adversarial directions.

Query Statistics for each method		
Method	Avg. Num. Queries (lower is better)	Median of Queries (lower is better)
Vanilla SimBA	279.7	291
Foreground-Sampling	**173.3**	**156**
Background-Sampling	285.9	302

Table 9: Query statistics for each method on modified MNIST Dataset.

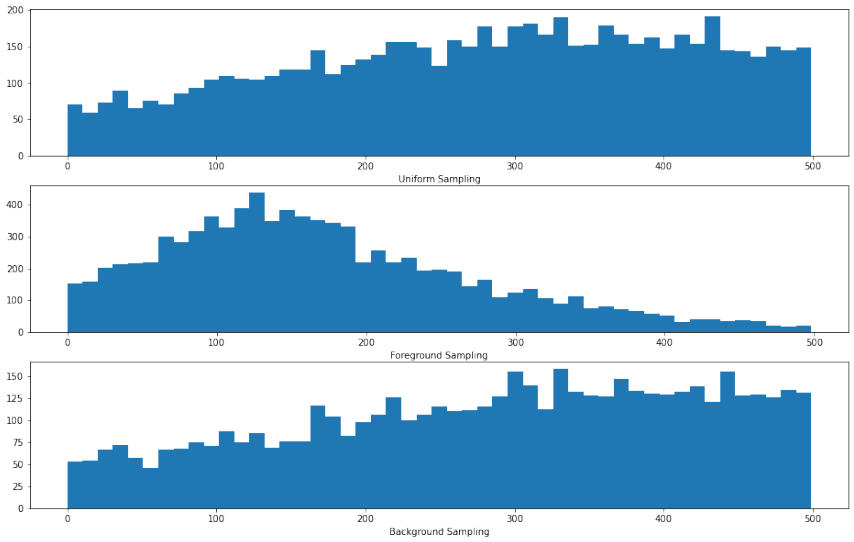

Figure 6: Histogram of queries per method on modified MNIST dataset. Both of the uniform sampling and background sampling methods lack off providing a query consistency whereas the peak around 110 is observable for foreground sampling (Table 9).

5 Conclusion

We proposed to improve the query efficiency of scoring-based black-box adversarial attacks by guiding with the saliency map as the prior distribution. The proposed approach is tested on both CIFAR10 and MNIST datasets. The base attack method, vanilla SimBA, does a local search on the input space, sampling the pixels uniformly in the image domain. The findings show that the deep classifier becomes more and more sensitive to the foreground objects as the background of the samples is extremely noisy. On the other hand, since the algorithm's performance strictly depends on the performance of the saliency detector, the original MNIST dataset is exploited as the perfect saliency map. This experiment strongly supports the proposed approach while providing query efficiency and increasing the attack success rate. Also, sampling from the background did not provide query efficiency compared to the vanilla setting, proving that the classifier focuses more on the object's foreground.

While this study proves that a prior input feature selection mechanism may decrease the number of queries needed to fool, it may be improved with future work. Not every classifier is more sensitive to the foreground object. Therefore, a joint optimization mechanism that highlights the parts where an unknown classifier is more focused on, expands the method's application domains and makes it possible to adapt each data distribution. Also, the vice versa is valid for a defense mechanism. This joint optimization architecture may reveal parts where the classifier is more focused on and can be used to increase the robustness. Moreover, it also enhances the interpretability of neural networks.

References

[1] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[2] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014.

[3] C. Etmann, S. Lunz, P. Maass, and C.-B. Schönlieb. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*, 2019.

[4] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(10):1915–1926, 2011.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] J. Gu and V. Tresp. Saliency methods for explaining adversarial attacks. *arXiv preprint arXiv:1908.08413*, 2019.

[7] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493. PMLR, 2019.

[8] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3203–3212, 2017.

[9] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2007.

[10] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018.

[11] H. Jing, C. Meng, X. He, and W. Wei. Black box explanation guided decision-based adversarial attacks. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, pages 1592–1596. IEEE, 2019.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 478–487, 2016.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[15] P. Mangla, V. Singh, and V. N. Balasubramanian. On saliency maps and adversarial robustness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 272–288. Springer, 2020.

[16] S. Montabone and A. Soto. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image and Vision Computing*, 28(3):391–402, 2010.

[17] N. Narodytska and S. P. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *CVPR Workshops*, volume 2, 2017.

[18] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[19] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

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

[21] J. Wang, Z. Yin, J. Jiang, and Y. Du. Attention-guided black-box adversarial attacks with large-scale multiobjective evolutionary optimization. *arXiv preprint arXiv:2101.07512*, 2021.

[22] J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9029–9038, 2018.

[23] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

[24] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.