One Pixel Attack for Fooling Deep Neural Networks

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai

***Abstract*—Recent research has revealed that the output of deep neural networks (DNNs) can be easily altered by adding relatively small perturbations to the input vector. In this paper, we analyze an attack in an extremely limited scenario where only one pixel can be modified. For that we propose a novel method for generating one-pixel adversarial perturbations based on differential evolution (DE). It requires less adversarial information (a black-box attack) and can fool more types of networks due to the inherent features of DE. The results show that 67.97% of the natural images in Kaggle CIFAR-10 test dataset and 16.04% of the ImageNet (ILSVRC 2012) test images can be perturbed to at least one target class by modifying just one pixel with 74.03% and 22.91% confidence on average. We also show the same vulnerability on the original CIFAR-10 dataset. Thus, the proposed attack explores a different take on adversarial machine learning in an extreme limited scenario, showing that current DNNs are also vulnerable to such low dimension attacks. Besides, we also illustrate an important application of DE (or broadly speaking, evolutionary computation) in the domain of adversarial machine learning: creating tools that can effectively generate low-cost adversarial attacks against neural networks for evaluating robustness.**

***Index Terms*—Convolutional neural network, differential evolution (DE), image recognition, information security.**

I. Introduction

In THE domain of image recognition, deep neural network (DNN)-based approach has outperform traditional image processing techniques, achieving even human-competitive results [25]. However, several studies have revealed that artificial perturbations on natural images can easily make DNN misclassify and accordingly proposed effective algorithms for generating such samples called “adversarial images” [7], [11], [18], [24]. A common idea for creating adversarial images is adding a tiny amount of well-tuned additive perturbation, which is expected to be imperceptible to human eyes, to a correctly classified natural image. Such modification can cause the classifier to label the modified image as a completely different class. Unfortunately, most of the previous attacks did not consider extremely limited scenarios for adversarial attacks, namely the modifications might be excessive (i.e., the amount of modified pixels is fairly large) such that it may be perceptible to human eyes (see Fig. 3 for an example). Additionally, investigating adversarial images created under extremely limited scenarios might give new insights about the geometrical characteristics and overall behavior of DNN’s model in high-dimensional space [9]. For example, the characteristics of adversarial images close to the decision boundaries can help describing the boundaries’ shape.

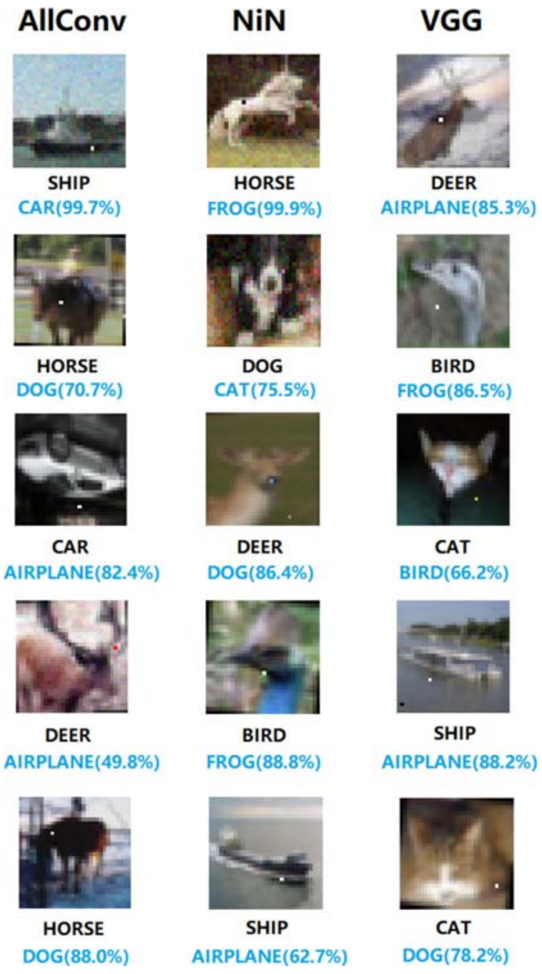


Fig. 1. One-pixel attacks created with the proposed algorithm that successfully fooled three types of DNNs trained on CIFAR-10 dataset: the AllConv, NiN, and VGG. The original class labels are in black color while the target class labels and the corresponding confidence are given below.

In this paper, by perturbing only one pixel with differential evolution (DE), we propose a black-box DNN attack in a scenario where the only information available is the probability

Manuscript received March 12, 2018; revised June 21, 2018 and October 29, 2018; accepted December 20, 2018. Date of publication January 4, 2019; date of current version October 1, 2019. This work was supported in part by the Collaboration Hubs for International Program of SICORP, in part by the Japan Science and Technology Agency, and in part by the Kyushu University Education and Research Center for Mathematical and Data Science Grant. *(Jiawei Su and Danilo Vasconcellos Vargas contributed equally to this work.) (Corresponding author: Jiawei Su.)*

J. Su and D. V. Vargas are with the Graduate School/Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan (e-mail: jiawei.su@inf.kyushu-u.ac.jp).

K. Sakurai is with the Graduate School/Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan, and also with Advanced Telecommunications Research Institute International, Kyoto, Japan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2019.2890858

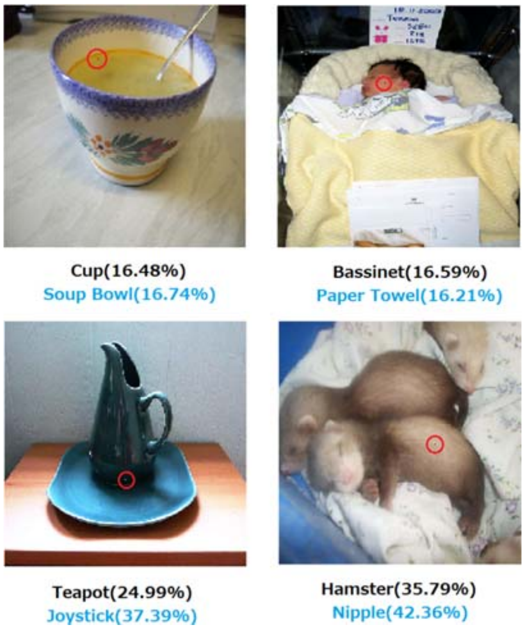


Fig. 2. One-pixel attacks on ImageNet dataset where the modified pixels are highlighted with red circles. The original class labels are in black color while the target class labels and their corresponding confidence are given below.

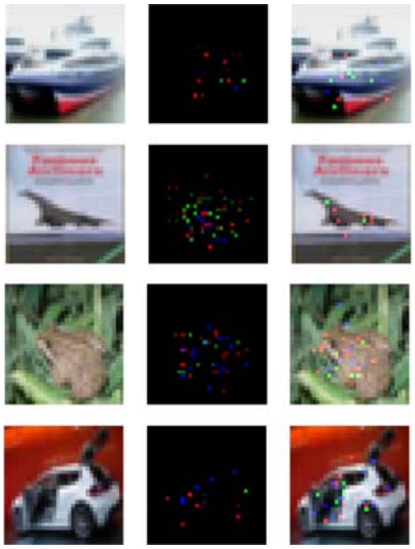


Fig. 3. Illustration of the adversarial images generated by using Jacobian saliency-map approach [18]. The perturbation is conducted on about 4% of the total pixels and can be obvious to human eyes. Since the adversarial pixel perturbation has become a common way of generating adversarial images, such abnormal "noise" might be recognized with expertise.

labels (Figs. 1 and 2) Our proposal has mainly the following advantages compared to previous works.

1) *Effectiveness:* On Kaggle CIFAR-10 dataset, being able to launch nontargeted attacks by only modifying one pixel on three common DNN structures with 68.71%, 71.66%, and 63.53% success rates. We additionally find that each natural image can be perturbed to 1.8, 2.1, and 1.5 other classes. On the original CIFAR-10 dataset with a more limited attack scenario, we show 22.60%, 35.20%, and 31.40% success rates. On ImageNet dataset, nontargeted attacking the BVLC AlexNet model also by changing one pixel shows that 16.04% of the test images can be attacked.
2) *Semiblack-Box Attack:* Requires only black-box feedback (probability labels) but no inner information of target DNNs such as gradients and network structures. Our method is also simpler than existing approaches since it does not abstract the problem of searching perturbation to any explicit target functions but directly focus on increasing the probability label values of the target classes.
3) *Flexibility:* Can attack more types of DNNs (e.g., networks that are not differentiable or when the gradient calculation is difficult).

Regarding the extremely limited one-pixel attack scenario, there are several main reasons why we consider it.

1) *Analyze the Vicinity of Natural Images:* Geometrically, several previous works have analyzed the vicinity of natural images by limiting the length of perturbation vector. For example, the universal perturbation adds small value to each pixel such that it searches the adversarial images in a sphere region around the natural image [14]. On the other side, the proposed few-pixel perturbations can be regarded as cutting the input space using very low-dimensional slices, which is a different way of exploring the features of high-dimensional DNN input space. Among them, one-pixel attack is an extreme case of several-pixel attack. Theoretically, it can give geometrical insight to the understanding of CNN input space, in contrast to another extreme case: universal adversarial perturbation [14] that modifies every pixel.
2) *A Measure of Perceptiveness:* The attack can be effective for hiding adversarial modification in practice. To the best of our knowledge, none of the previous works can guarantee that the perturbation made can be completely imperceptible. A direct way of mitigating this problem is to limit the amount of modifications to as few as possible. Specifically, instead of theoretically proposing additional constraints or considering more complex cost functions for conducting perturbation, we propose an empirical solution by limiting the number of pixels that can be modified. In other words, we use the number of pixels as units instead of length of perturbation vector to measure the perturbation strength and consider the worst case which is one-pixel modification, as well as two other scenarios (i.e., 3 and 5 pixels) for comparison.

II. Related Works

The security problem of DNN has become a critical topic [1], [2]. Szegedy *et al.* [24] first revealed the sensitivity to well-tuned artificial perturbation which can be crafted by several gradient-based algorithms using back-propagation for obtaining gradient information [11], [24]. Specifically, Goodfellow *et al.* [11] proposed "fast gradient sign" algorithm for calculating effective perturbation based on a hypothesis in which the linearity and high dimensions of inputs are the main reason that a broad class of networks are sensitive to small perturbation. Moosavi-Dezfooli *et al.* [7] proposed a greedy perturbation searching method by assuming the linearity of DNN decision boundaries. In addition, Papernot *et al.* [18] utilized Jacobian matrix to build "adversarial saliency map" which indicates the effectiveness of conducting a fixed length perturbation through the direction of each axis [18], [20]. Except adversarial perturbation, there are other ways of creating adversarial images to make the DNN misclassify, such as artificial image [16] and rotation [36]. Besides, adversarial perturbation can be also possible in other domains, such as speech recognition [33], natural language processing [34] and malware classification [35].

A number of detection and defense methods have been also proposed to mitigate the vulnerability induced by adversarial perturbation [38]. For instance, network distillation which was originally proposed for squeezing information of an network to a smaller one is found to be able to reduce the network sensitivity enhancing the robustness of the neural network [39]. Adversarial training [40] is proposed for adding adversarial images to the training data such that the robustness against known adversarial images can be improved. On the other side, some image processing methods are proved to be effective for detecting adversarial images. For example, Liang *et al.* [42] showed that noise reduction methods, such as scalar quantization and spatial smoothing filter can be selectively utilized for mitigating the influence of adversarial perturbation. By comparing the label of an image before and after the transformation the perturbation can be detected. The method works well on detecting adversarial images with both low and high entropy. Similarly, Xu *et al.* [41] showed that squeezing color bits and local/nonlocal spatial smoothing can have high success rate on distinguishing adversarial images. However, recent studies show that many of these defense and detection methods can be effectively evaded by conducting little modification on the original attacks [44], [45], [59].

Several black-box attacks that require no internal knowledge about the target systems, such as gradients, have also been proposed [5], [15], [17]. In particular, to the best of our knowledge, the only work before ours that ever mentioned using one-pixel modification to change class labels is carried out by Narodytska and Kasiviswanathan [15]. However, differently from this paper, they only utilized it as a starting point to derive a further semi black-box attack which needs to modify more pixels (e.g., about 30 pixels out of 1024) without considering the scenario of one-pixel attack. In addition, they have neither measured systematically the effectiveness of the attack nor obtained quantitative results for evaluation. An analysis of the one-pixel attack's geometrical features as well as further discussion about its implications are also lacking.

There have been many efforts to understand DNN by visualizing the activation of network nodes [28]–[30] while the geometrical characteristics of DNN boundary have gained less attraction due to the difficulty of understanding high-dimensional space. However, the robustness evaluation of DNN with respect to adversarial perturbation might shed light in this complex problem [9]. For example, both natural and random images are found to be vulnerable to adversarial perturbation. Assuming these images are evenly distributed, it suggests that most data points in the input space are gathered near to the boundaries [9]. In addition, Fawzi *et al.* [10] revealed more clues by conducting a curvature analysis. Their conclusion is that the region along most directions around natural images are flat with only few directions where the space is curved and the images are sensitive to perturbation [10]. Interestingly, universal perturbations (i.e., a perturbation that when added to any natural image can generate adversarial images with high effectiveness) were shown possible and to achieve a high effectiveness when compared to random perturbation. This indicates that the diversity of boundaries might be low while the boundaries' shapes near different data points are similar [14].

III. Methodology

A. Problem Description

Generating adversarial images can be formalized as an optimization problem with constraints. We assume an input image can be represented by a vector in which each scalar element represents one pixel. Let f be the target image classifier which receives n-dimensional inputs, $\mathbf{x} = (x_1, \ldots, x_n)$ be the original natural image correctly classified as class t. The probability of $\mathbf{x}$ belonging to the class t is therefore $f_t(\mathbf{x})$. The vector $e(\mathbf{x}) = (e_1, \ldots, e_n)$ is an additive adversarial perturbation according to $\mathbf{x}$, the target class adv and the limitation of maximum modification L. Note that L is always measured by the length of vector $e(\mathbf{x})$. The goal of adversaries in the case of targeted attacks is to find the optimized solution $e(\mathbf{x})^*$ for the following question:

$$\begin{aligned} &\underset{e(\mathbf{x})^*}{\text{maximize}} && f_{\text{adv}}(\mathbf{x} + e(\mathbf{x})) \\ &\text{subje ct to} && \|e(\mathbf{x})\| \leq L. \end{aligned}$$

The problem involves finding two values: 1) which dimensions that need to be perturbed and 2) the corresponding strength of the modification for each dimension. In our approach, the equation is slightly different

$$\begin{aligned} &\underset{e(\mathbf{x})^*}{\text{maximize}} && f_{\text{adv}}(\mathbf{x} + e(\mathbf{x})) \\ &\text{subject to} && \|e(\mathbf{x})\|_0 \leq d \end{aligned}$$

where d is a small number. In the case of one-pixel attack $d = 1$. Previous works commonly modify a part of all dimensions while in our approach only d dimensions are modified with the other dimensions of $e(\mathbf{x})$ left to zeros.

The one-pixel modification can be seen as perturbing the data point along a direction parallel to the axis of one of the n dimensions. Similarly, the 3 (5)-pixel modification moves the data points within 3 (5)-dimensional cubes. Overall, few-pixel attack conducts perturbations on the low-dimensional slices of input space. In fact, one-pixel perturbation allows the modification of an image towards a chosen direction out of n possible directions with arbitrary strength. This is illustrated in Fig. 4 for the case when $n = 3$.

Thus, usual adversarial images are constructed by perturbating all pixels with an overall constraint on the strength of accumulated modification [8], [14] while the few-pixel

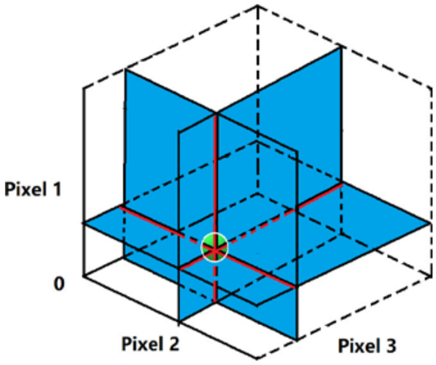


Fig. 4. Illustration of using one and two-pixel perturbation attack in a 3-D input space (i.e., the image has three pixels). The green point (sphere) denotes a natural image. In the case of one-pixel perturbation, the search space is the three perpendicular lines that intersect at point of natural image, which are denoted by red and black stripes. For two-pixel perturbation, the search space is the three blue (shaded) 2-D planes. In summary, one- and two-pixel attacks search the perturbation on, respectively, 1-D and 2-D slices of the original 3-D input space.

attack considered in this paper is the opposite which specifically focus on few pixels but does not limit the strength of modification.

B. Differential Evolution

DE is a population-based optimization algorithm for solving complex multi-modal optimization problems [6], [23]. DE belongs to the general class of evolutionary algorithms (EAs). Moreover, it has mechanisms in the population selection phase that keep the diversity such that in practice it is expected to efficiently find higher quality solutions than gradient-based solutions or even other kinds of EAs [4]. In specific, during each iteration another set of candidate solutions (children) is generated according to the current population (parents). Then the children are compared with their corresponding parents, surviving if they are more fitted (possess higher fitness value) than their parents. In such a way, only comparing the parent and his child, the goal of keeping diversity and improving fitness values can be simultaneously achieved.

DE does not use the gradient information for optimizing and therefore does not require the objective function to be differentiable or previously known. Thus, it can be utilized on a wider range of optimization problems compared to gradient-based methods (e.g., nondifferentiable, dynamic, noisy, among others). The use of DE for generating adversarial images have the following main advantages.

1) *Higher Probability of Finding Global Optima:* DE is a meta-heuristic which is relatively less subject to local minima than gradient descent or greedy search algorithms (this is in part due to diversity keeping mechanisms and the use of a set of candidate solutions). Moreover, the problem considered in this paper has a strict constraint (only one pixel can be modified) making it relatively harder.
2) *Require Less Information From Target System:* DE does not require the optimization problem to be differentiable as is required by classical optimization methods, such as gradient descent and quasi-Newton methods. This is critical in the case of generating adversarial images since: a) there are networks that are not differentiable, for instance [26] and b) calculating gradient requires much more information about the target system which can be hardly realistic in many cases.
3) *Simplicity:* The approach proposed here is independent of the classifier used. For the attack to take place it is sufficient to know the probability labels.

There are many DE variations/improvements, such as self-adaptive [3], multiobjective [27], among others. This paper can be further improved by taking these variations/improvements into account.

C. Method and Settings

We encode the perturbation into an array (candidate solution) which is optimized (evolved) by DE. One candidate solution contains a fixed number of perturbations and each perturbation is a tuple holding five elements: *x*-*y* coordinates and RGB value of the perturbation. One perturbation modifies one pixel. The initial number of candidate solutions (population) is 400 and at each iteration another 400 candidate solutions (children) will be produced by using the usual DE formula

$$\begin{aligned} x_i(g+1) &= x_{r1}(g) + F(x_{r2}(g) - x_{r3}(g)) \\ r1 &\neq r2 \neq r3 \end{aligned}$$

where x_i is an element of the candidate solution, $r1$–$r3$ are random numbers, F is the scale parameter set to be 0.5, and g is the current index of generation. Once generated, each candidate solution compete with their corresponding parents according to the index of the population and the winner survive for next iteration. The maximum number of iteration is set to 100 and early-stop criterion will be triggered when the probability label of target class exceeds 90% in the case of targeted attacks on Kaggle CIFAR-10, and when the label of true class is lower than 5% in the case of nontargeted attacks on ImageNet. Then the label of true class is compared with the highest nontrue class to evaluate if the attack succeeded. The initial population is initialized by using uniform distributions $U(1, 32)$ for CIFAR-10 images and $U(1, 227)$ for ImageNet images, for generating *x*-*y* coordinate (e.g., the image has a size of 32 × 32 in CIFAR-10 and for ImageNet we unify the original images with various resolutions to 227 × 227) and Gaussian distributions N ($\mu = 128$, $\sigma = 127$) for RGB values. The fitness function is simply the probabilistic label of the target class in the case of CIFAR-10 and the label of true class in the case of ImageNet. The crossover is not included in our scheme.

IV. Evaluation and Results

The evaluation of the proposed attack method is based on CIFAR-10 and ImageNet datasets. We introduce several metrics to measure the effectiveness of the attacks.

1) *Success Rate:* In the case of nontargeted attacks, it is defined as the percentage of adversarial images that were successfully classified by the target system as an arbitrary target class. In the case of targeted attack, it is

TABLE I
AllConv

conv2d layer(kernel=3, stride = 1, depth=96)
conv2d layer(kernel=3, stride = 1, depth=96)
conv2d layer(kernel=3, stride = 2, depth=96)
conv2d layer(kernel=3, stride = 1, depth=192)
conv2d layer(kernel=3, stride = 1, depth=192)
dropout(0.3)
conv2d layer(kernel=3, stride = 2, depth=192)
conv2d layer(kernel=3, stride = 2, depth=192)
conv2d layer(kernel=1, stride = 1, depth=192)
conv2d layer(kernel=1, stride = 1, depth=10)
average pooling layer(kernel=6, stride=1)
flatten layer
softmax classifier

defined as the probability of perturbing a natural image to a specific target class.

2) *Adversarial Probability Labels (Confidence):* Accumulates the values of probability label of the target class for each successful perturbation, then divided by the total number of successful perturbations. The measure indicates the average confidence given by the target system when misclassifying adversarial images.
3) *Number of Target Classes:* Counts the number of natural images that successfully perturb to a certain number (i.e., from 0 to 9) of target classes. In particular, by counting the number of images that can not be perturbed to any other classes, the effectiveness of nontargeted attack can be evaluated.
4) *Number of Original-Target Class Pairs:* Counts the number of times each original-destination class pair was attacked.

A. Kaggle CIFAR-10

We train three types of common networks: 1) all convolutional network (AllConv) [22]; 2) network in network (NiN) [13]; and 3) VGG16 network [21] as target image classifiers on CIFAR-10 dataset [12], [63]. The structures of the networks are described in Tables I–III. The network setting were kept as similar as possible to the original with a few modifications in order to get the highest classification accuracy. Both the scenarios of targeted and nontargeted attacks are considered. For each of the attacks on the three types of neural networks 500 natural images are randomly selected from the Kaggle CIFAR-10 test dataset to conduct the attack.

Note that we use the Kaggle CIFAR-10 test dataset [63] instead of the original one for this experiments. The dataset contains 300 000 CIFAR-10 images which can be visually inspected to have the following modifications: duplication, rotation, clipping, blurring, adding few random bad pixels, and so on. However, the exact employed modification algorithm is not released. This makes it a more practical dataset which simulates common scenarios that images can contain unknown random noise. We also show the results on the original CIFAR-10 test dataset in Section V for comparison.

In addition, an experiment is conducted on the AllConv [22] by generating 500 adversarial images with three and five pixel-modification. The objective is to compare one-pixel attack with three and five pixel attacks. For each natural image, nine target attacks are launched trying to perturb it to the other nine target classes. Note that we actually only launch targeted attacks and the effectiveness of nontargeted attack is evaluated based on targeted attack results. That is, if an image can be perturbed to at least one target class out of total nine classes, the nontargeted attack on this image succeeds. Overall, it leads to the total of 36 000 adversarial images created. To evaluate the effectiveness of the attacks, some established measures from the literature are used as well as some new kinds of measures are introduced.

TABLE II
NiN

conv2d layer(kernel=5, stride = 1, depth=192)
conv2d layer(kernel=1, stride = 1, depth=160)
conv2d layer(kernel=1, stride = 1, depth=96)
max pooling layer(kernel=3, stride=2)
dropout(0.5)
conv2d layer(kernel=5, stride = 1, depth=192)
conv2d layer(kernel=5, stride = 1, depth=192)
conv2d layer(kernel=5, stride = 1, depth=192)
average pooling layer(kernel=3, stride=2)
dropout(0.5)
conv2d layer(kernel=3, stride = 1, depth=192)
conv2d layer(kernel=1, stride = 1, depth=192)
conv2d layer(kernel=1, stride = 1, depth=10)
average pooling layer(kernel=8, stride=1)
flatten layer
softmax classifier

TABLE III
VGG16 Network

conv2d layer(kernel=3, stride = 1, depth=64)
conv2d layer(kernel=3, stride = 1, depth=64)
max pooling layer(kernel=2, stride=2)
conv2d layer(kernel=3, stride = 1, depth=128)
conv2d layer(kernel=3, stride = 1, depth=128)
max pooling layer(kernel=2, stride=2)
conv2d layer(kernel=3, stride = 1, depth=256)
conv2d layer(kernel=3, stride = 1, depth=256)
conv2d layer(kernel=3, stride = 1, depth=256)
max pooling layer(kernel=2, stride=2)
conv2d layer(kernel=3, stride = 1, depth=512)
conv2d layer(kernel=3, stride = 1, depth=512)
conv2d layer(kernel=3, stride = 1, depth=512)
max pooling layer(kernel=2, stride=2)
conv2d layer(kernel=3, stride = 1, depth=512)
conv2d layer(kernel=3, stride = 1, depth=512)
conv2d layer(kernel=3, stride = 1, depth=512)
max pooling layer(kernel=2, stride=2)
flatten layer
fully connected(size=2048)
fully connected(size=2048)
softmax classifier

B. ImageNet

For ImageNet, we applied a nontargeted attack with the same DE parameter settings used on the CIFAR-10 dataset, although ImageNet has a search space 50 times larger than CIFAR-10. Note that we actually launch the nontargeted attack for ImageNet by using a fitness function that aims to decrease the probability label of the true class. Different from CIFAR-10, whose effectiveness of nontargeted attack is calculated based on the targeted attack results carried out by using a fitness function for increasing the probability of target classes. Given the time constraints, we conduct the experiment without proportionally increasing the number of evaluations, i.e., we keep the same number of evaluations. Our tests are run over the BVLC AlexNet using 105 images from ILSVRC 2012 test set selected randomly for the attack. For ImageNet, we only conduct one pixel attack because we want to verify if such a tiny modification can fool images with larger size and if it is computationally tractable to conduct such attacks. The ILSVRC 2012 images are in lossy jpeg format with nonunified sizes. In order to reduce the practical interference to the evaluation as much as possible, we first convert all target images from jpeg to png therefore during later processing it will be lossless. The images are further resized to 227 × 227 resolution for inputting to AlexNet (using nearest filter). Then we follow the same procedure to attacking CIFAR-10. Note that the discrepancy on preprocessing raw images (e.g., using center cropping instead of simple resizing) can influence the classification performance of AlexNet and attack rate. Here we only show the result on one setting and leave the comprehensive evaluation of attacking AlexNet using difference preprocessing methods for future work.

C. Results

The success rates and adversarial probability labels for one-pixel perturbations on three CIFAR-10 networks and BVLC network are shown in Table IV and the three- and five-pixel perturbations on Kaggle CIFAR-10 is shown in Table V. The number of target classes is shown by Fig. 5. The number of original-target class pairs is shown by the heat-maps of Figs. 6 and 7. In addition to the number of original-target class pairs, the total number of times each class had an attack which either originated or targeted it is shown in Fig. 8. Since only nontargeted attacks are launched on ImageNet, the "number of target classes" and "number of original-target class pairs" metrics are not included in the ImageNet results.

1) Success Rate and Adversarial Probability Labels (Targeted Attack Results): On Kaggle CIFAR-10, the success rates of one-pixel attacks on three types of networks show the generalized effectiveness of the proposed attack through different network structures. On average, each image can be perturbed to about two target classes for each network. In addition, by increasing the number of pixels that can be modified to three and five, the number of target classes that can be reached increases significantly. By dividing the adversarial probability labels by the success rates, the confidence values (i.e., probability labels of target classes) are obtained which are 79.39%, 79.17%, and 77.09%, respectively, to one-, three-, and five-pixel attacks.

On ImageNet, the results show that the one pixel attack generalizes well to large size images and fool the corresponding neural networks. In particular, there is 16.04% chance that an arbitrary ImageNet test image can be perturbed to a target class with 22.91% confidence. Note that the ImageNet results are done with the same settings as CIFAR-10 while the resolution of images we use for the ImageNet test is 227 × 227, which is 50 times larger than CIFAR-10 (32 × 32). Notice that in each successful attack the probability label of the target class is the highest. Therefore, the confidence of 22.91% is relatively low but tell us that the other remaining 999 classes are even lower to an almost uniform soft label distribution. Thus, the one-pixel attack can break the confidence of BVLC AlexNet to a nearly uniform soft label distribution. The low confidence is caused by the fact that we utilized a nontargeted evaluation that only focuses on decreasing the probability of the true class. Other fitness functions should give different results.

TABLE IV
RESULTS OF CONDUCTING ONE-PIXEL ATTACK ON FOUR DIFFERENT TYPES OF NETWORKS: ALLCONV, NIN, VGG16, AND BVLC ALEXNET. THE ORIGINALACC IS THE ACCURACY ON THE NATURAL TEST DATASETS. TARGETED/NONTARGETED INDICATE THE ACCURACY OF CONDUCTING TARGETED/NONTARGETED ATTACKS. CONFIDENCE IS THE AVERAGE PROBABILITY OF TARGET CLASSES

	AllConv	NiN	VGG16	BVLC
OriginAcc	85.6%	87.2%	83.3%	57.3%
Targeted	19.82%	23.15%	16.48%	–
Non-targeted	68.71%	71.66%	63.53%	16.04%
Confidence	79.40%	75.02%	67.67%	22.91%

TABLE V
RESULTS OF CONDUCTING THREE-PIXEL ATTACK ON ALLCONV NETWORKS AND FIVE-PIXEL ATTACK ON NIN

	3 pixels	5 pixels
Success rate(tar)	40.57%	44.00%
Success rate(non-tar)	86.53%	86.34%
Rate/Labels	79.17%	77.09%

2) Number of Target Classes (Nontargeted Attack Results): Regarding the results shown in Fig. 5, we find that with only one-pixel modification a fair amount of natural images can be perturbed to two, three, and four target classes. By increasing the number of pixels modified, perturbation to more target classes becomes highly probable. In the case of nontargeted one-pixel attack, the VGG16 network got a slightly higher robustness against the proposed attack. This suggests that all three types of networks (AllConv network, NiN, and VGG16) are vulnerable to this type of attack.

The results of attacks are competitive with previous nontargeted attack methods which need much more distortions (Table VI). It shows that using 1-D perturbation vectors is enough to find the corresponding adversarial images for most of the natural images. In fact, by increasing the number of pixels up to five, a considerable number of images can be simultaneously perturbed to eight target classes. In some rare cases, an image can go to all other target classes with one-pixel modification, which is illustrated in Fig. 9.

3) Original-Target Class Pairs: Some specific original-target class pairs are much more vulnerable than others (Figs. 6 and 7). For example, images of cat (class 3) can be much more easily perturbed to dog (class 5) but can hardly reach the automobile (class 1). This indicates that the vulnerable target classes (directions) are shared by different data points that belong to the same class. Moreover, in the case of one-pixel attack, some classes are more robust than others since their

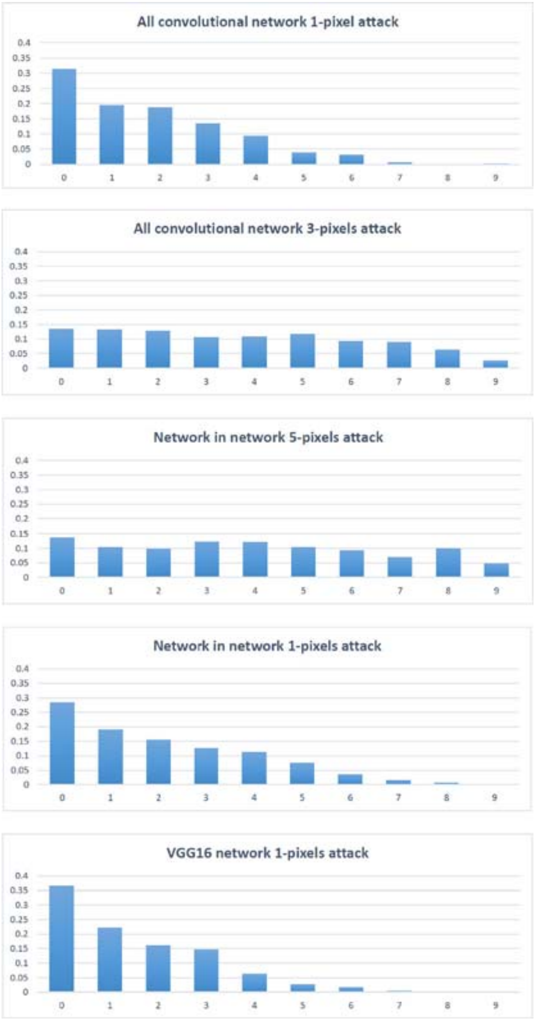
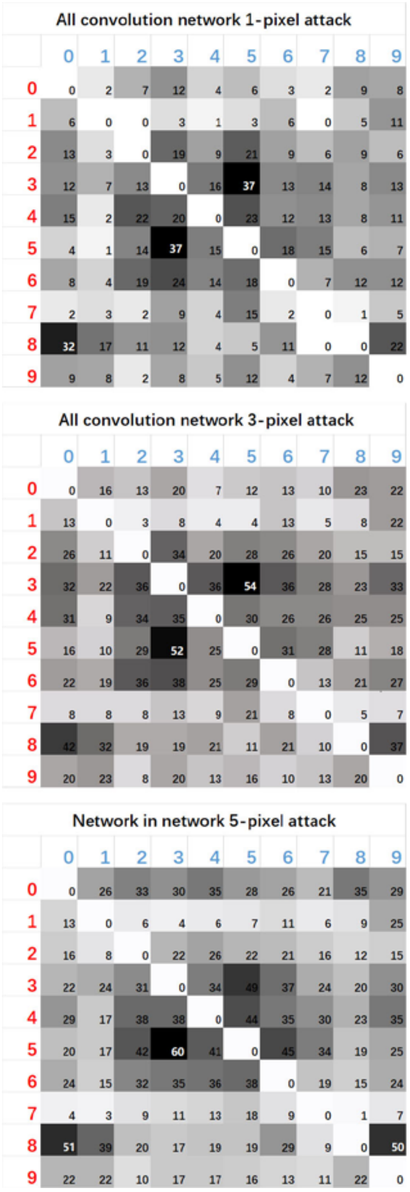


Fig. 5. Graphs shows the percentage of natural images that were successfully perturbed to a certain number (from 0 to 9) of target classes by using one-, three-, or five-pixel perturbation. The vertical axis shows the percentage of images that can be perturbed while the horizontal axis indicates the number of target classes.



All convolution network 1-pixel attack

	0	1	2	3	4	5	6	7	8	9
0	0	2	7	12	4	6	3	2	9	8
1	6	0	0	3	1	3	6	0	5	11
2	13	3	0	19	9	21	9	6	9	6
3	12	7	13	0	16	37	13	14	8	13
4	15	2	22	20	0	23	12	13	8	11
5	4	1	14	37	15	0	18	15	6	7
6	8	4	19	24	14	18	0	7	12	12
7	2	3	2	9	4	15	2	0	1	5
8	32	17	11	12	4	5	11	0	0	22
9	9	8	2	8	5	12	4	7	12	0

All convolution network 3-pixel attack

	0	1	2	3	4	5	6	7	8	9
0	0	16	13	20	7	12	13	10	23	22
1	13	0	3	8	4	4	13	5	8	22
2	26	11	0	34	20	28	26	20	15	15
3	32	22	36	0	36	54	36	28	23	33
4	31	9	34	35	0	30	26	26	25	25
5	16	10	29	52	25	0	31	28	11	18
6	22	19	36	38	25	29	0	13	21	27
7	8	8	8	13	9	21	8	0	5	7
8	42	32	19	19	21	11	21	10	0	37
9	20	23	8	20	13	16	10	13	20	0

Network in network 5-pixel attack

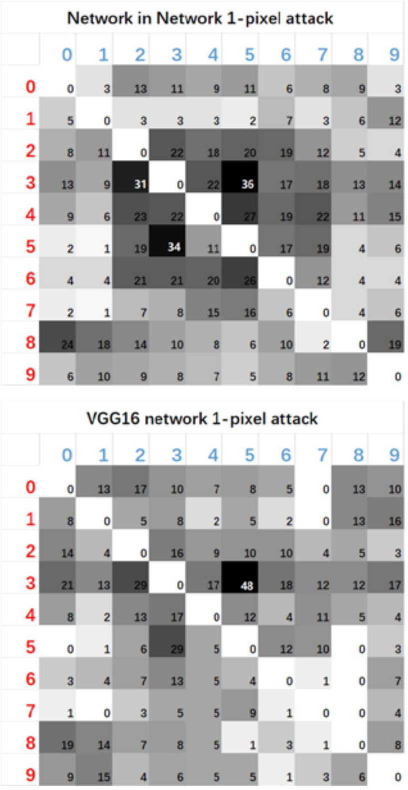
	0	1	2	3	4	5	6	7	8	9
0	0	26	33	30	35	28	26	21	35	29
1	13	0	6	4	6	7	11	6	9	25
2	16	8	0	22	26	22	21	16	12	15
3	22	24	31	0	34	49	37	24	20	30
4	29	17	38	38	0	44	35	30	23	35
5	20	17	42	60	41	0	45	34	19	25
6	24	15	32	35	36	38	0	19	15	24
7	4	3	9	11	13	18	9	0	1	7
8	51	39	20	17	19	19	29	9	0	50
9	22	22	10	17	17	16	13	11	22	0

Fig. 6. Heat-maps of the number of times a successful attack is present with the corresponding original-target class pair in one-, three-, and five-pixel attack cases. Red (vertical) and (horizontal) blue indices indicate, respectively, the original and target classes. The number from 0 to 9 indicates, respectively, the following classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

data points can be relatively hard to perturb to other classes. Among these data points, there are points that can not be perturbed to any other classes. This indicates that the labels of these points rarely change when going across the input space through n directions perpendicular to the axes. Therefore, the corresponding original classes are kept robust along these directions. However, it can be seen that such robustness can rather easily be broken by merely increasing the dimensions of perturbation from one to three and five because both success rates and number of target classes that can be reached increase when conducting higher-dimensional perturbations.

Additionally, it can also be seen that each heat-map matrix is approximately symmetric, indicating that each class has similar number of adversarial images which were crafted from these classes as well as to these classes (Fig. 8). Having said that, there are some exceptions for example the class 8 (ship) when attacking NiN, the class 4 (deer) when attacking AllConv networks with one pixel, among others. In the ship class when attacking NiN networks, for example, it is relatively easy to craft adversarial images from them while it is relatively hard to craft adversarial images to them. Such unbalance is intriguing since it indicates the ship class is similar to most of the other classes like truck and airplane but not vice-versa. This might be due to: 1) boundary shape and 2) how close are natural images to the boundary. In other words, if the boundary shape is wide enough it is possible to have natural images far away from the boundary such that it is hard to craft adversarial images from it. On the contrary, if the boundary shape is mostly long and thin with natural images close to the border, it is easy to craft adversarial images from them but hard to craft adversarial images to them.

In practice, such classes which are easy to craft adversarial images from may be exploited by malicious users which may make the whole system vulnerable. In the case



Network in Network 1-pixel attack

	0	1	2	3	4	5	6	7	8	9
0	0	3	13	11	9	11	6	8	9	3
1	5	0	3	3	3	2	7	3	6	12
2	8	11	0	22	18	20	19	12	5	4
3	13	9	31	0	22	36	17	18	13	14
4	9	6	23	22	0	27	19	22	11	15
5	2	1	19	34	11	0	17	19	4	6
6	4	4	21	21	20	26	0	12	4	4
7	2	1	7	8	15	16	6	0	4	6
8	24	18	14	10	8	6	10	2	0	19
9	6	10	9	8	7	5	8	11	12	0

VGG16 network 1-pixel attack

	0	1	2	3	4	5	6	7	8	9
0	0	13	17	10	7	8	5	0	13	10
1	8	0	5	8	2	5	2	0	13	16
2	14	4	0	16	9	10	10	4	5	3
3	21	13	29	0	17	48	18	12	12	17
4	8	2	13	17	0	12	4	11	5	4
5	0	1	6	29	5	0	12	10	0	3
6	3	4	7	13	5	4	0	1	0	7
7	1	0	3	5	5	9	1	0	0	4
8	19	14	7	8	5	1	3	1	0	8
9	9	15	4	6	5	5	1	3	6	0

Fig. 7. Heat-maps for one-pixel attack on NiN and VGG.

TABLE VI

COST OF CONDUCTING ONE-PIXEL ATTACK ON FOUR DIFFERENT TYPES OF NETWORKS. AVGEVALUATION IS THE AVERAGE NUMBER OF EVALUATIONS TO PRODUCE ADVERSARIAL IMAGES. AVGDISTORTION IS THE REQUIRED AVERAGE DISTORTION IN ONE-CHANNEL OF A SINGLE PIXEL TO PRODUCE ADVERSARIAL IMAGES

	AllConv	NiN	VGG16	BVLC
AvgEvaluation	16000	12400	20000	25600
AvgDistortion	123	133	145	158

here, however, the exceptions are not shared between the networks, revealing that whatever is causing the phenomenon is not shared. Therefore, for the current systems under the given attacks, such a vulnerability seems hard to be exploited.

4) Time Complexity and Average Distortion: To evaluate the time complexity we use the number of evaluations which is a common metric in optimization. In the DE case, the number of evaluations is equal to the population size multiplied by the number of generations. We also calculate the average distortion on the single pixel attacked by taking the average modification on the three color channels, which is a more straight forward and explicit measure of modification strength. We did not use the L_p norm due to its limited effectiveness of measuring perceptiveness [37]. The results of two metrics are shown in Table VII.

5) Comparing With Random One-Pixel Attack: We compare the proposed method with the random attack to evaluate if DE is truly helpful for conducting one-pixel nontargeted attack on Kaggle CIFAR-10 dataset, which is shown in Table VIII.

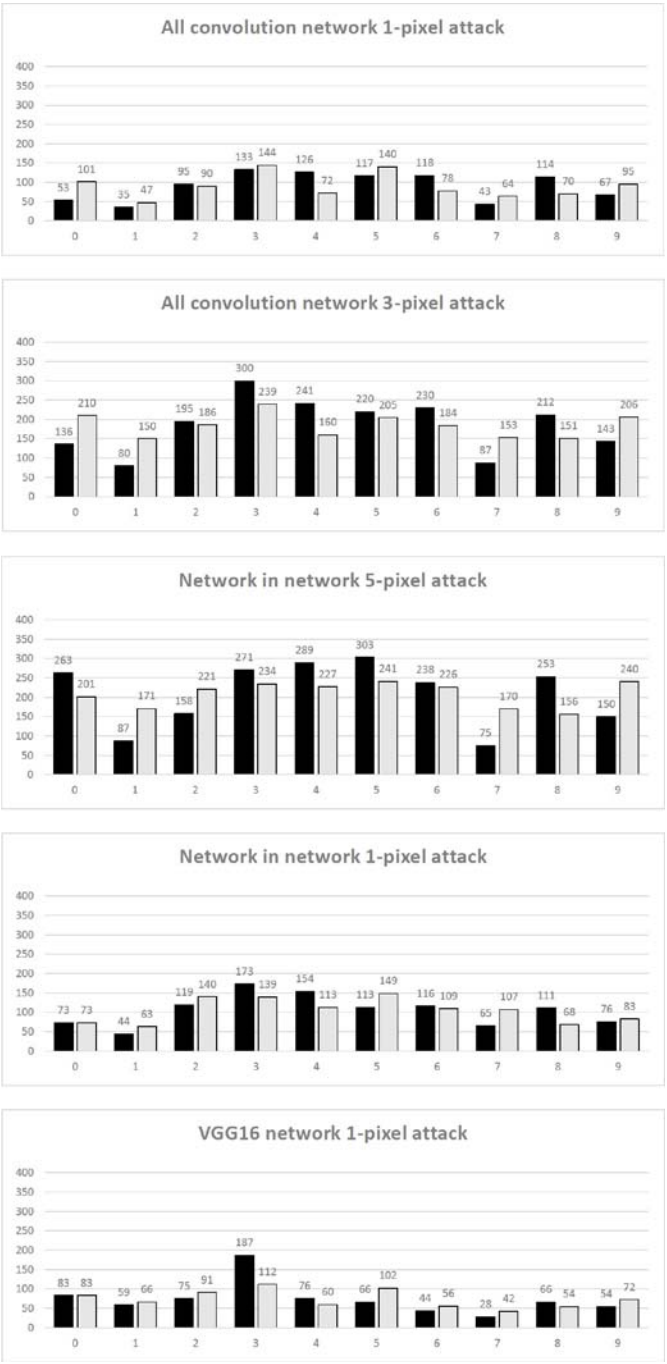


Fig. 8. Number of successful attacks (vertical axis) for a specific class acting as the original (black) and target (gray) class. The horizontal axis indicates the index of each class which is the same as Fig. 7.

Specifically, for each natural image, the random search repeats 100 times, each time randomly modifies one random pixel of the image with random RGB value to attempt to change its label. The confidence of the attack with respect to one image is set to be the highest probability target class label of 100 attacks.

In this experiment, we use the same number of evaluations (80 000) for both DE and random search. According to the comparison, the DE is superior to the random attack regarding attack accuracy, especially in the case of VGG16 network.

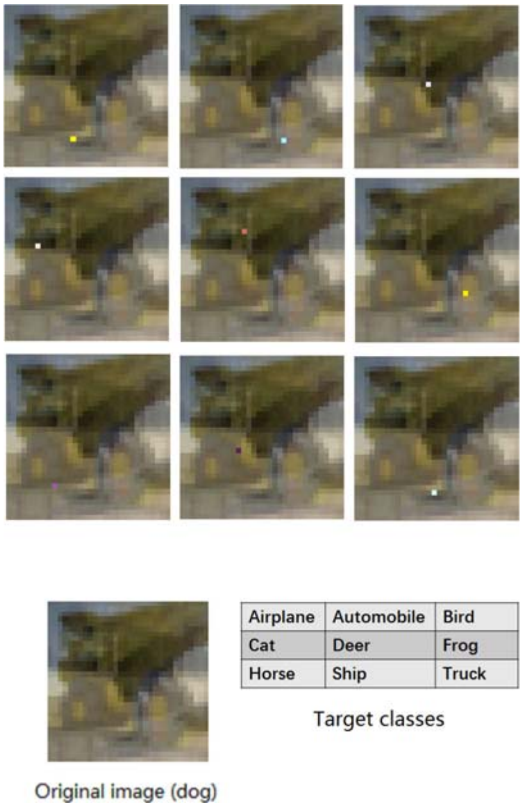


Fig. 9. Natural image of the dog class that can be perturbed to all other nine classes. The attack is conducted over the AllConv network using the proposed one pixel attack. The table in the bottom shows the class labels output by the target DNN, all with approximately 100% confidence. This curious result further emphasize the difference and limitations of current methods when compared to human recognition.

TABLE VII
COMPARISON OF ATTACK RATE AND CONFIDENCE BETWEEN DE ONE-PIXEL ATTACK AND RANDOM ONE-PIXEL ATTACK (NONTARGETED) ON KAGGLE CIFAR-10 DATASET

	AllConv	NiN	VGG16
DE success rate	68.71%	71.66%	63.53%
Confidence	79.40%	75.02%	67.67%
Random Search success rate	49.70%	41.72%	15.57%
Confidence	87.73%	75.83%	59.90%

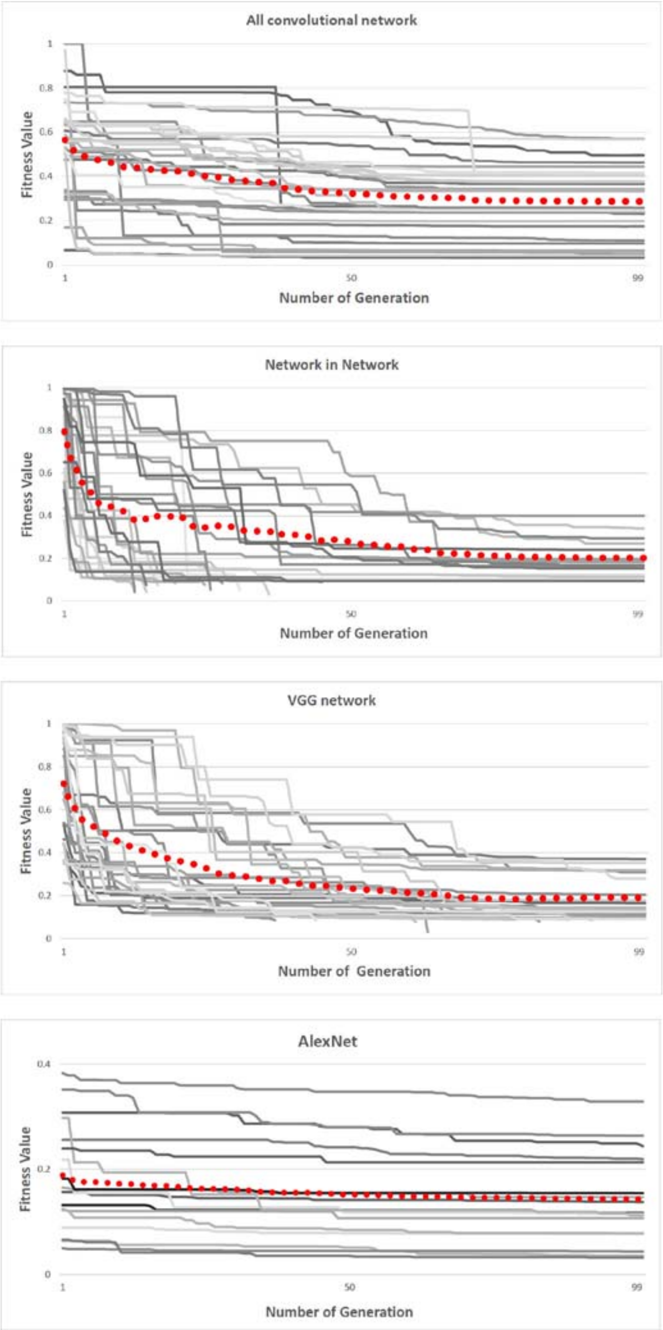


Fig. 10. Change of fitness values during 100 generations of evolution of images (nontargeted) attacked by the proposed method among different network structures. The average values are highlighted by red dotted lines.

Specifically, DE is 19.01%, 29.94%, and 47.96% more efficient than random search, respectively, for AllConv, NiN, and VGG16. Even with a less efficient result, random search is shown to find 49.70% and 41.72% of the time for, respectively, AllConv and NiN, therefore the vulnerable pixels that can change the image label significantly are quite common. That seems not to be the case for VGG though in which random search achieves only 15.57%. DE has a similar accuracy in all of them showing also a better robustness.

6) Change in Fitness Values: We run an experiment over different networks to examine how the fitness changes during evolution. The 30 (15) curves come from 30 (15) random Kaggle CIFAR-10 (ImageNet) images successfully attacked by the proposed one-pixel attack (Fig. 10). The fitness values are, as previously described, set to be the probability label of the true class for each image. The goal of the attack is to minimize this fitness value. According to the results, it can be seen that the fitness values can occasionally drop abruptly between two generations while in other cases they decrease smoothly. Moreover, the average fitness value decreases monotonically with the number of generations, showing that the evolution works as expected. We also find that BVLC network is harder to fool due to the smaller decrease in fitness values.

V. RESULTS ON ORIGINAL CIFAR-10 TEST DATA

We present another evaluation of one pixel attack which is on original CIFAR-10 test dataset [12]. Comparing to the results on Kaggle CIFAR-10 aforementioned, the scenario is more limited since the images contain much less practical

TABLE VIII
RESULTS OF CONDUCTING ONE-PIXEL ATTACK ON ORIGINAL CIFAR-10 TEST SET. NONTARGETED1 INDICATES THE NONTARGETED ATTACK ACCURACY CALCULATED FROM TARGETED ATTACK RESULTS AND NONTARGETED2 INDICATES THE TRUE NONTARGETED ATTACK ACCURACY. OTHER METRICS ARE THE SAME TO TABLE IV

	AllConv	NiN	VGG16
Targeted	3.41%	4.78%	5.63%
Non-targeted1	22.67%	32.00%	30.33%
Confidence	54.58%	55.18%	51.19%
Non-targeted2	22.60%	35.20%	31.40%
Confidence	56.57%	60.08%	53.58%

noise. Therefore, the target CNNs can have higher classification accuracy and confidence which definitely makes the attack harder. Additionally, we only use images correctly classified by the target CNNs while in the experiment on Kaggle CIFAR-10 set we use all images (i.e., which contain wrongly classified images) with their true labels predicted by the target CNNs.

We use 500 random images for nontargeted attack and 300 for targeted attack. We also make small modification on network structure for better implementation. Specifically, for the NiN, we remove the second average pooling layers. For AllConv, we remove the batch normalization on the first layer. Three CIFAR-10 networks are retrained to have similar natural accuracy to Table IV. An early-stop criterion will be triggered when the probability label of the target class exceeds the original class. All other settings are kept the same. The attack results are shown by Table VIII. The number of target classes is shown by Fig. 11. The number of original-target class pairs is shown by the heat-maps of Figs. 12 and 13. In addition to the number of original-target class pairs, the total number of times each class had an attack which either originated or targeted it is shown in Figs. 14 and 15.

According to the attack results shown, we find the following features of one-pixel attack on original CIFAR-10.

A. Attack Rate

The three networks have higher robustness to one-pixel attack according to the lower attack rate and confidence (Table VIII). This might due to the higher classification accuracy and confidence of three networks on original CIFAR-10 test-set. Similar to the results on Kaggle set, the NiN still gets the lowest overall robustness considering both attack rate and confidence. This might be related to the proximity to the decision boundary. However, VGG network becomes much more vulnerable in this case. The discrepancy indicates that the robustness among different networks can be varied when handling images with low (e.g., Kaggle CIFAR-10) and high (e.g., original CIFAR-10) confidence.

B. Number of Targeted Classes

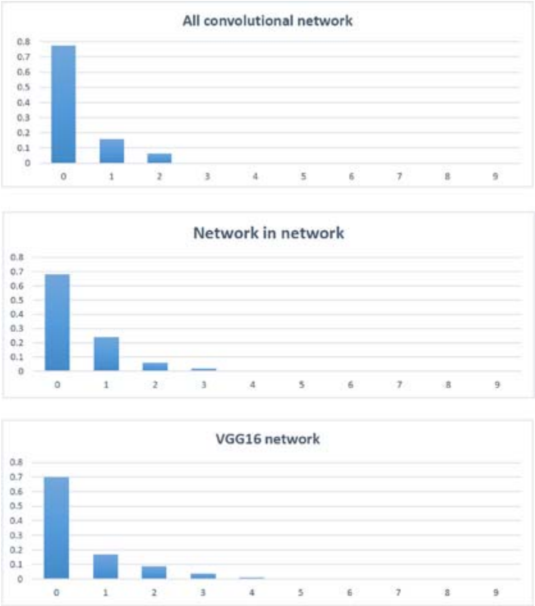


Fig. 11. Percentage of natural images that were successfully perturbed to a certain number (from 0 to 9) of target classes by one pixel targeted attack.

According to Fig. 11, it can be seen that in the case of targeted attack, it is still quite common that a vulnerable image can be perturbed to more than one class. In other words, the image might be locate near to the boundaries to multiple classes, especially in the case of VGG. This is similar to the Kaggle CIFAR-10 results shown by Fig. 5.

Note that one image can be perturbed to a final target class A through the original target class B (i.e., semisuccessful targeted attack). For some images, the number of B can be more than one. We do not count it as a successful targeted attack unless $A = B$.

C. Original-Target Class Pairs

In both cases of targeted and nontargeted attack, we again found the existence of vulnerable original-target classes pairs such as dog (fifth)-cat (third) (Figs. 12 and 13). In most cases, for a class pair between classes A and B, the number of successful perturbation from A to B is similar to the number of B to A, which makes the heat-maps almost symmetric. However, there are exceptions, such as ship (eighth)-airplane (zeroth) pair, which the perturbation from ship to airplane class is very frequent but not vice versa.

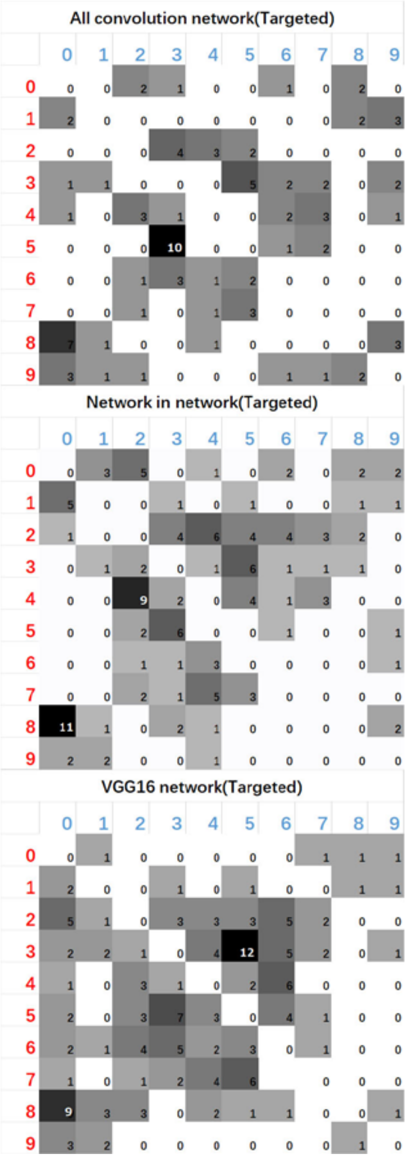
Additionally, it also can be seen from Figs. 14 and 15, some vulnerable classes exist which have higher number of times being both original and target class of the attack. A vulnerable original class is probably also vulnerable being a target class to a similar extend.

Most of these features, together with the specific vulnerable class-pairs shown by Figs. 12 and 13 and vulnerable classes shown by Figs. 14 and 15, are similar or even exactly the same to the finding on attacking Kaggle CIFAR-10 dataset.

VI. DISCUSSION

A. Adversarial Perturbation

Previous results have shown that many data points might be located near to the decision boundaries [9]. For the analysis, the data points were moved small steps in the input space while quantitatively analyzing the frequency of change in the



All convolution network(Targeted)

	0	1	2	3	4	5	6	7	8	9
0	0	0	2	1	0	0	1	0	2	0
1	2	0	0	0	0	0	0	0	2	3
2	0	0	0	4	3	2	0	0	0	0
3	1	1	0	0	0	5	2	2	0	2
4	1	0	3	1	0	0	2	3	0	1
5	0	0	0	10	0	0	1	2	0	0
6	0	0	1	3	1	2	0	0	0	0
7	0	0	1	0	1	3	0	0	0	0
8	7	1	0	0	1	0	0	0	0	3
9	3	1	1	0	0	0	1	1	2	0

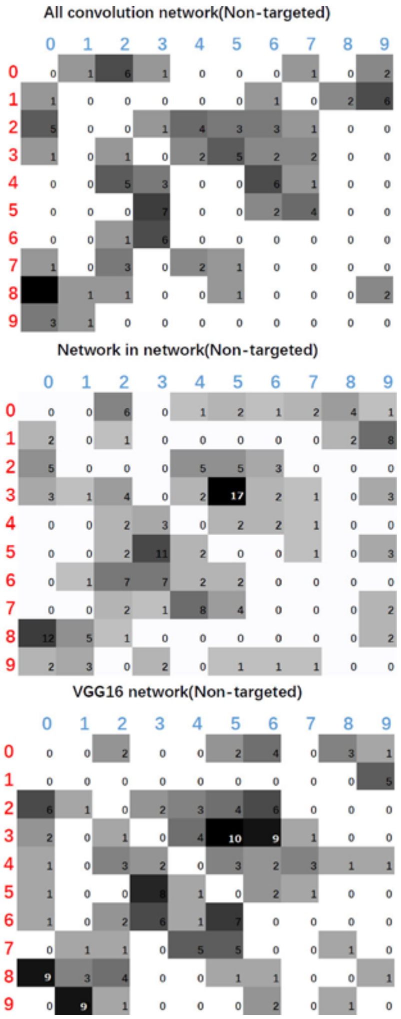
Network in network(Targeted)

	0	1	2	3	4	5	6	7	8	9
0	0	3	5	0	1	0	2	0	2	2
1	5	0	0	1	0	1	0	0	1	1
2	1	0	0	4	6	4	4	3	2	0
3	0	1	2	0	1	6	1	1	1	0
4	0	0	9	2	0	4	1	3	0	0
5	0	0	2	6	0	0	1	0	0	1
6	0	0	1	1	3	0	0	0	0	1
7	0	0	2	1	5	3	0	0	0	0
8	11	1	0	2	1	0	0	0	0	2
9	2	2	0	0	1	0	0	0	0	0

VGG16 network(Targeted)

	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	0	0	0	1	1	1
1	2	0	0	1	0	1	0	0	1	1
2	5	1	0	3	3	3	5	2	0	0
3	2	2	1	0	4	12	5	2	0	1
4	1	0	3	1	0	2	6	0	0	0
5	2	0	3	7	3	0	4	1	0	0
6	2	1	4	5	2	3	0	1	0	0
7	1	0	1	2	4	6		0	0	0
8	9	3	3	0	2	1	1	0	0	1
9	3	2	0	0	0	0	0	0	1	0

Fig. 12. Heat-maps of the number of times a successful attack is present with the corresponding original-target class pair, for targeted attacks.



All convolution network(Non-targeted)

	0	1	2	3	4	5	6	7	8	9
0	0	1	6	1	0	0	0	1	0	2
1	1	0	0	0	0	0	1	0	2	6
2	5	0	0	1	4	3	3	1	0	0
3	1	0	1	0	2	5	2	2	0	0
4	0	0	5	3	0	0	6	1	0	0
5	0	0	0	7	0	0	2	4	0	0
6	0	0	1	6	0	0	0	0	0	0
7	1	0	3	0	2	1	0	0	0	0
8		1	1	0	0	1	0	0	0	2
9	3	1	0	0	0	0	0	0	0	0

Network in network(Non-targeted)

	0	1	2	3	4	5	6	7	8	9
0	0	0	6	0	1	2	1	2	4	1
1	2	0	1	0	0	0	0	0	2	8
2	5	0	0	0	5	5	3	0	0	0
3	3	1	4	0	2	17	2	1	0	3
4	0	0	2	3	0	2	2	1	0	0
5	0	0	2	11	2	0	0	1	0	3
6	0	1	7	7	2	2	0	0	0	0
7	0	0	2	1	8	4	0	0	0	2
8	12	5	1	0	0	0	0	0	0	2
9	2	3	0	2	0	1	1	1	0	0

VGG16 network(Non-targeted)

	0	1	2	3	4	5	6	7	8	9
0	0	0	2	0	0	2	4	0	3	1
1	0	0	0	0	0	0	0	0	0	5
2	6	1	0	2	3	4	6	0	0	0
3	2	0	1	0	4	10	9	1	0	0
4	1	0	3	2	0	3	2	3	1	1
5	1	0	0	8	1	0	2	1	0	0
6	1	0	2	6	1	7	0	0	0	0
7	0	1	1	0	5	5	0	0	1	0
8	9	3	4	0	0	1	1	0	0	1
9	0	9	1	0	0	0	2	0	1	0

Fig. 13. Heat-maps of the number of times a successful attack is present with the corresponding original-target class pair, for nontargeted attacks.

class labels. In this paper, we showed that it is also possible to move the data points along few dimension to find points where the class labels change. Our results also suggest that the assumption made by Goodfellow *et al*. that small addictive perturbation on the values of many dimensions will accumulate and cause huge change to the output [11], might not be necessary for explaining why natural images are sensitive to small perturbation. Since we only changed one pixel to successfully perturb a considerable number of images.

According to the experimental results, the vulnerability of CNN exploited by the proposed one pixel attack is generalized through different network structures as well as different image sizes. In addition, the results shown here mimics an attacker and therefore, uses a low number of DE iterations with a relatively small set of initial candidate solutions. Therefore, the perturbation success rates should improve further by having either more iterations or a bigger set of initial candidate solutions. Implementing more advanced algorithms such as covariance matrix adaptation evolution strategy [32] instead of DE might also achieve the same improvement. Additionally, the proposed algorithm and the widely vulnerable images (i.e., natural images that can be used to craft adversarial images to most of the other classes) collected might be useful for generating better artificial adversarial images in order to augment the training dataset. This aids the development of more robust models [19] which is left as future work.

B. Robustness of One-Pixel Attack

Some recently proposed detection methods have shown high accuracy of detecting adversarial perturbation. For example, Liang *et al*. [42] utilized noise reduction to effectively detect both high and low-entropy images (e.g., bigger images give high entropy values). In addition, Xu *et al*. [41] showed that squeezing color bits and local/nonlocal spatial smoothing can simultaneously detect L_0, L_2, and L_∞ attacks. As the trade-off of being a low-cost, easy-implemented L_0 attack, we do not expect one pixel attack can achieve significantly better robustness against such detection methods compared to other L_0 attacks such as [32].

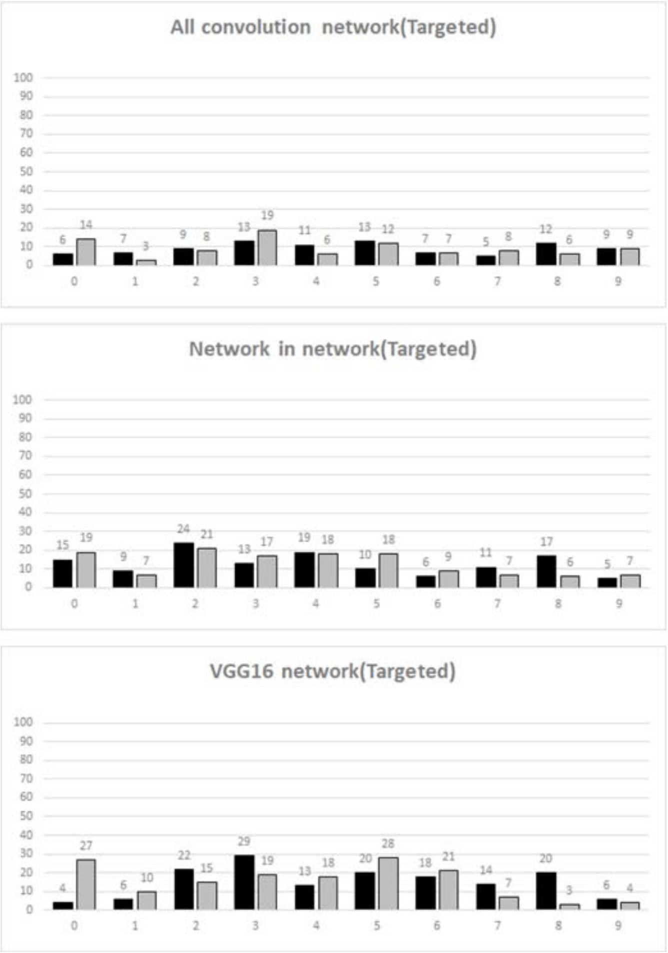


Fig. 14. Number of successful attacks (vertical axis) for a specific class acting as the original (black) and target (gray) class, for targeted attacks.

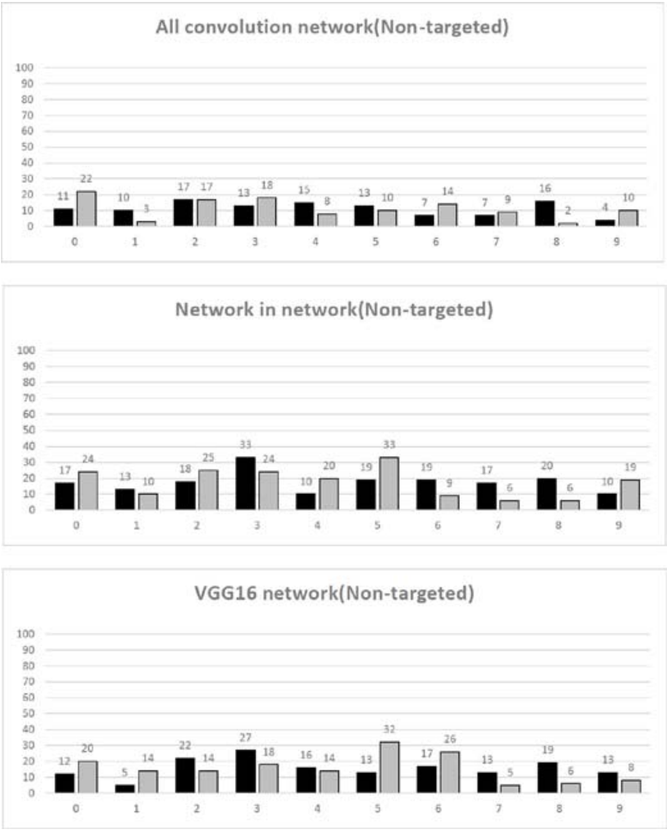


Fig. 15. Number of successful attacks (vertical axis) for a specific class acting as the original (black) and target (gray) class, for nontargeted attacks.

TABLE IX

COMPASSION OF NONTARGETED ATTACK EFFECTIVENESS BETWEEN THE PROPOSED METHOD AND TWO PREVIOUS WORKS. THIS SUGGESTS THAT ONE PIXEL IS ENOUGH TO CREATE ADVERSARIAL IMAGES FROM MOST OF THE NATURAL IMAGES

Method	Success rate	Confidence	Number of pixels	Network
Our method	35.20%	60.08%	1 (0.098%)	NiN
Our method	31.40%	53.58%	1 (0.098%)	VGG
LSA[15]	97.89%	72%	33 (3.24%)	NiN
LSA[15]	97.98%	77%	30 (2.99%)	VGG
FGSM[11]	93.67%	93%	1024 (100%)	NiN
FGSM[11]	90.93%	90%	1024 (100%)	VGG

However, such detection schemes add another layer of preprocessing which increases the response time of the system. For example, both Xu *et al.* [41] and Liang *et al.* [42] required image processing and reclassification of the resulting images. Therefore, they can be inefficient when dealing with adversarial scenarios such as novelty detection on security camera and image recognition systems on autonomous driving applications which run in real time with high frame rate. Besides, the impact of preprocessing on the classification accuracy is still not fully understood.

Detecting adversarial perturbation indeed can be helpful in practice. However, the fundamental problem is still left unsolved: the neural networks are still not able to recognize similar images as such, ignoring small adversarial perturbation. By proposing novel attack methods, we aim to emphasize the existence of different types of vulnerabilities and the corresponding understanding.

VII. FUTURE WORK

The DE utilized in this paper belongs to a big class of algorithms called evolutionary strategies [47] which includes other variants, such as adaptive DE [48] and covariance matrix adaptation evolution strategy (CMA-ES) [49]–[51]. In fact, there are a couple of recent developments [60]–[62] in evolutionary strategies and related areas that could further improve the current method, allowing for more efficient and accurate attacks.

Furthermore, evolutionary computation also provides some promising approaches to solve adversarial machine learning related vulnerabilities. In fact, evolutionary-based machine learning allows for a great flexibility of models and may be an answer to the same problems it is revealing. First, in an area of evolutionary machine learning called neuroevolution, it was shown to be possible to learn not just the weights but also the topology of the network with evolutionary computation [26], [43], [52]. In fact, SUNA [26] goes beyond current neural models to propose a unified neuron model (e.g., time-scales, neuromodulation, feedback, and long-term memory) that can adapt its structure and models to learn completely different problems (including non-Markov problems) without changing any of its hyper-parameters. This generality is currently surpassing most if not all deep learning algorithms. Last but not least, self-organizing and novelty-organizing classifiers can adapt to changes in the environment by using flexible representations [53]–[55]. For example, they can adapt to mazes that change in shape and to problems where the scope of variables change throughout the experiment [56]: a very challenging scenario in which most if not all deep learning algorithms fail. These among other achievements [57], [58] show a promising path that may solve current problems in DNNs in the years to come.

Besides, it can be seen that the one-pixel attack can be potentially extended to other domains, such as natural language processing, speech recognition, which will be also left for future work.

References

[1] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010.

[2] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf. Comput. Commun. Security*, 2006, pp. 16–25.

[3] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.

[4] P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 315–346, 2013.

[5] H. Dang, Y. Huang, and E.-C. Chang, "Evading classifiers by morphing in the dark," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 119–133.

[6] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[7] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.

[8] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto, "Analysis of universal adversarial perturbations," *arXiv preprint arXiv:1705.09554*, 2017.

[9] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "A geometric perspective on the robustness of deep networks," Inst. Elect. Electron. Eng., Piscataway, NJ, USA, Rep., 2017.

[10] A. Fawzi, S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto, "Classification regions of deep neural networks," *arXiv preprint arXiv:1705.09552*, 2017.

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

[12] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.

[13] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[14] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 86–94.

[15] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2017, pp. 1310–1318.

[16] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Con. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Security*, 2017, pp. 506–519.

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

[19] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 25–32.

[20] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. ICLR*, 2015, pp. 1–14.

[23] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[24] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. ICLR*, 2014.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[26] D. V. Vargas and J. Murata, "Spectrum-diverse neuroevolution with unified neural models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1759–1773, Aug. 2017.

[27] D. V. Vargas, J. Murata, H. Takano, and A. C. B. Delbem, "General subpopulation framework and taming the conflict inside populations," *Evol. Comput.*, vol. 23, no. 1, pp. 1–36, 2015.

[28] D. Wei, B. Zhou, A. Torrabla, and W. Freeman, "Understanding intra-class knowledge inside CNN," *arXiv preprint arXiv:1507.02379*, 2015.

[29] J. Yosinski, J. Clune, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Proc. ICML Workshop Deep Learn.*, 2015.

[30] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[31] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *arXiv preprint arXiv:1710.08864*, 2017.

[32] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a New Evolutionary Computation*. Heidelberg, Germany: Springer, 2006, pp. 75–102.

[33] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? Adversarial examples against automatic speech recognition," *arXiv preprint arXiv:1801.00554*, 2018.

[34] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," *arXiv preprint arXiv:1604.08275*, 2016.

[35] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

[36] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "A rotation and a translation suffice: Fooling CNNs with simple transformations," *arXiv preprint arXiv:1712.02779*, 2017.

[37] M. Sharif, L. Bauer, and M. K. Reiter, "On the suitability of L_p-norms for creating and preventing adversarial examples," *arXiv preprint arXiv:1802.09653*, 2018.

[38] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *arXiv preprint arXiv:1712.07107*, 2017.

[39] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Security Privacy (SP)*, 2016, pp. 582–597.

[40] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvari, "Learning with a strong adversary," *arXiv preprint arXiv:1511.03034*, 2015.

[41] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks" *arXiv preprint arXiv:1704.01155*, 2017.

[42] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep networks with adaptive noise reduction," *arXiv preprint arXiv:1705.08378*, 2017.

[43] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.

[44] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Security*, 2017, pp. 3–14.

[45] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," *arXiv preprint arXiv:1607.04311*, 2016.

[46] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 39–57.

[47] H. G. Beyer and H. P. Schwefel, "Evolution strategies—A comprehensive introduction," *Nat. Comput.*, vol. 1, no. 1, pp. 3–52, Mar. 2002.

[48] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 1785–1791.

[49] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput*, vol. 11, no. 1, pp. 1–18, Mar. 2003.

[50] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proc. IEEE Conf. Evol. Comput.*, 1996, pp. 312–317.

[51] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput*, vol. 9, no. 2, pp. 159–195, Jun. 2001.

[52] D. Whitley, S. Dominic, R. Das, and C. W. Anderson, "Genetic reinforcement learning for neurocontrol problems," *Mach. Learn.*, vol. 13, nos. 2–3, pp. 259–284, 1993.

[53] D. V. Vargas, H. Takano, and J. Murata, "Self organizing classifiers: First steps in structured evolutionary machine learning," *Evol. Intell.*, vol. 6, no. 2, pp. 57–72, 2013.

[54] D. V. Vargas, H. Takano, and J. Murata, "Self organizing classifiers and niched fitness," in *Proc. 15th Annu. Conf. Genet. Evol. Comput.*, 2013, pp. 1109–1116.

[55] D. V. Vargas, H. Takano, and J. Murata, "Novelty-organizing team of classifiers—A team-individual multi-objective approach to reinforcement learning," in *Proc. SICE Annu. Conf. (SICE)*, 2014, pp. 1785–1792.

[56] D. V. Vargas, H. Takano, and J. Murata, "Novelty-organizing team of classifiers in noisy and dynamic environments," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2015, pp. 2937–2944.

[57] R. J. Urbanowicz and J. H. Moore, "ExSTraCS 2.0: Description and evaluation of a scalable learning classifier system," *Evol. Intell.*, vol. 8, no. 2, pp. 89–116, 2015.

[58] I. M. Alvarez, W. N. Browne, and M. Zhang, "Compaction for code fragment based learning classifier systems," in *Proc. Aust. Conf. Artif. Life Comput. Intell.*, 2016, pp. 41–53.

[59] N. Carlini and D. Wagner, "Magnet and 'efficient defenses against adversarial attacks are not robust to adversarial examples,"' *arXiv preprint arXiv:1711.08478*, 2017.

[60] A. Abdolmaleki, B. Price, N. Lau, L. P. Reis, and G. Neumann, "Deriving and improving CMA-ES with information geometric trust regions," in *Proc. Genet. Evol. Comput. Conf.*, 2017, pp. 657–664.

[61] K. Nishida and Y. Akimoto, "PSA-CMA-ES: CMA-ES with population size adaptation," in *Proc. Genet. Evol. Comput. Conf.*, 2018, pp. 865–872.

[62] M. Groves and J. Branke, "Sequential sampling for noisy optimisation with CMA-ES," in *Proc. Genet. Evol. Comput. Conf.*, 2018, pp. 1023–1030.

[63] *CIFAR-10—Object Recognition in Images@Kaggle*. Accessed: Feb. 1, 2018. [Online]. Available: https://www.kaggle.com/c/cifar-10/data