One pixel attack for fooling deep neural networks

Jiawei Su
Kyushu University
Japan
jiawei.su@inf.kyushu-u.ac.jp

Danilo Vasconcellos Vargas
Kyushu University
Japan
vargas@inf.kyushu-u.ac.jp

Sakurai Kouichi
Kyushu University
Japan
sakurai@csce.kyushu-u.ac.jp

Abstract

Recent research has revealed that the output of Deep Neural Networks (DNN) 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. It requires less adversarial information and can fool more types of networks. The results show that 70.97% of the natural images can be perturbed to at least one target class by modifying just one pixel with 97.47% confidence on average. 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.

1. Introduction

In the domain of image recognition, DNN-based approach has overcome 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" [18, 11, 24, 7]. 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 the amount of modified pixels is fairly large) such that it may be perceptible to human eyes (see Figure 2 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 samples close to the decision boundaries can help describing the boundaries' shape.

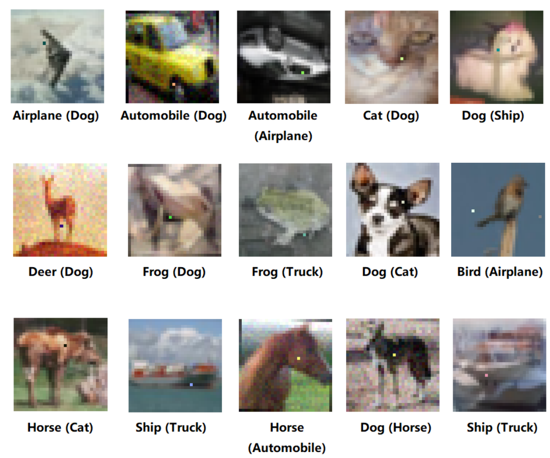

Figure 1. One-pixel attacks created with the proposed algorithm that successfully fooled a target DNN. The original class labels are written below each image with the target class label written inside brackets.

In this paper, by perturbing only one pixel with differential evolution, we propose a black-box DNN attack in a scenario where the only information available is the probability labels (Figure 1) Our proposal has mainly the following advantages compared to previous works:

- **Effectiveness** - Being able to launch non-targeted attacks by only modifying one pixel on three common deep neural network structures with 73.80%, 73.04% and 66.08% success rates. We additionally find that each natural image can be perturbed to 2.3, 2.5 and 1.9 other classes. In particular, 19.67% , 15.20% and 12.91% of the images can be perturbed to respectively one, two or three target classes on average.

- **Semi-Black-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 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.

- **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 two main reasons why we consider it:

- **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.

- **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.

2. Related works

The security problem of DNN has become a critical topic [2] [1]. C. Szegedy et al. first revealed the sensitivity to well-tuned artificial perturbation [24] which can be crafted by several gradient-based algorithms using back-propagation for obtaining gradient information [11, 24]. Specifically, I.J.Goodfellow et al. 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 [11]. S.M. Moosavi-Dezfooli et al. proposed a greedy perturbation searching method by assuming the linearity of DNN decision boundaries [7]. In addition, N. Papernot et al. utilize 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]. Another kind of adversarial image is also proposed by A. Nguyen et al. [16]. The images can hardly be recognized by human eyes but nevertheless classified by the network with high confidence.

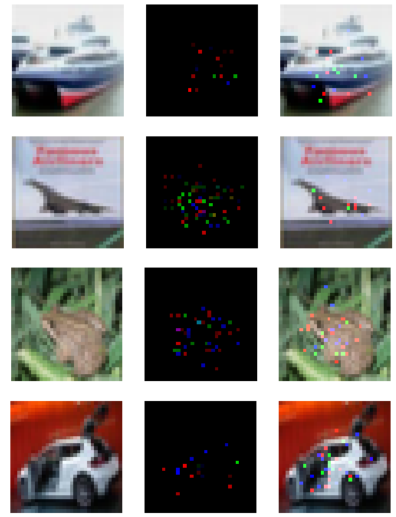

Figure 2. An illustration of the adversarial images generated by using Jacobian saliency-map approach [18]. The perturbation is conducted on about 4% of the total pixels and can be obvious to human eyes. Since the adversarial pixel perturbation has become a common way of generating adversarial images, such abnormal "noise" might be recognized with expertise.

Several black-box attacks that require no internal knowledge about the target systems such as gradients, have also been proposed [15, 17, 5]. 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 N. Narodytska et al[15]. However, differently from our work, 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 [30, 29, 28] 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, A. Fawzi et al. 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 samples with high effectiveness) were shown possible and to achieve a high effectiveness when compared to random pertubation. This indicates that the diversity of boundaries might be low while the boundaries' shapes near different data points are similar [14].

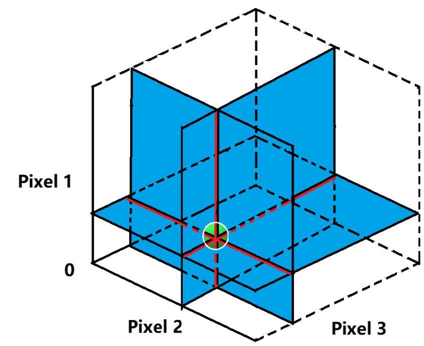

Figure 3. An illustration of using one and two-pixel perturbation attack in a 3-dimensional input space (i.e. the image has three pixels). The green point denotes a natural image. In the case of one-pixel perturbation, the search space is the three perpendicular lines denoted by red and black stripes. For two-pixel pertubations, the search space is the three blue two-dimensional planes. In summary, one and two-pixel attacks search the perturbation on respectively one and two dimensional slices of the original three dimensional input space.

3. Methodology

3.1. 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, $\mathbf{x} = (x_1, .., x_n)$ 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 $e(\mathbf{x}) = (e_1, .., e_n)$ 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:

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

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

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

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 Figure 3 for the case when $n = 3$.

Thus, usual adversarial samples are constructed by perturbating all pixels with an overall constraint on the strength of accumulated modification[14, 8] while the few-pixel attack considered in this paper is the opposite which specifically focus on few pixels but does not limit the strength of modification.

3.2. Differential Evolution

Differential evolution (DE) is a population based optimization algorithm for solving complex multi-modal optimization problems [23], [6]. DE belongs to the general class of evolutionary algorithms (EA). 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 (fathers). Then the children are compared with their corresponding fathers, surviving if they are more fitted (possess higher fitness value) than their fathers. In such a way, only comparing the father 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, non-differentiable, dynamic, noisy, among others). The use of DE for generating adversarial images have the following main advantages:

- **Higher probability of Finding Global Optima** - DE is a metaheuristic 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 article has a strict constraint (only one pixel can be modified) making it relatively harder.

- **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 1) There are networks that are not differentiable, for instance [26]. 2) Calculating gradient requires much more information about the target system which can be hardly realistic in many cases.

- **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], multi-objective [27], among others. The current work can be further improved by taking these variations/improvements into account.

3.3. Method and Settings

We encode the perturbation into an array (candidate solution) which is optimized (evolved) by differential evolution. 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:

$$x_i(g+1) = x_{r1}(g) + F(x_{r2}(g) + x_{r3}(g)),$$
$$r1 \neq r2 \neq r3,$$

where x_i is an element of the candidate solution, $r1, r2, r3$ are random numbers, F is the scale parameter set to be 0.5, g is the current index of generation. Once generated, each candidate solution compete with their corresponding father according to the index of the population and the winner survive for next iteration. The maximum number of iteration is set to 75 and an early-stop criteria will be triggered when the probability label of target class exceeds 99%. The initial population is initialized by using uniform distributions $U(1, 32)$ for generating x-y coordinate (e.g. the image has a size of 32X32) and Gaussian distributions $N(128, 127)$ for RGB values. The fitness function is simply the probabilistic label of the target class.

4. Evaluation and Results

We train 3 types of common networks: All convolution network [22], Network in Network[13] and VGG16 network[21] as target image classifiers on cifar-10 data set [12]. The structures of the networks are described by Table 1, 2 and 3. 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 non-targeted attacks are considered. For each of the attacks on the three types of neural networks 500 natural image samples are randomly selected from the cifar-10 test dataset to conduct the attack. In addition, an experiment is conducted on the all convolution network [22] by generating 500 adversarial samples 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 9 target classes. Overall, it leads to the total of 36000 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:

- **Success Rate** - In the case of non-targeted attacks, it is defined as the percentage of adversarial images that were successfully classified by the target system as an arbitrary target class. And in the case of targeted attack, it is defined as the probability of perturbing a natural image to a specific target class.

- **Adversarial Probability Labels** - 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 mis-classifying adversarial samples.

- **Number of Target Classes** - Counts the number of natural images that successfully perturb to a certain

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

Table 1. All convolution network

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)
flatten layer
softmax classifier

Table 2. Network in 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

Table 3. VGG16 network

	AllConv	NiN	VGG16
Success rate(tar)	23.46%	26.32%	19.78%
Success rate(non-tar)	73.80%	73.04%	66.08%
Accumulated labels	23.08%	25.29%	19.37%
Rate/Labels	98.37%	96.11%	97.93%

Table 4. Results of conducting one-pixel attack on three networks: All Convolution network (AllConv), Network in Network (NiN), and VGG16.

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 non-targeted attack can be evaluated.

- **Number of Original-Target Class Pairs** - Counts the number of times each original-destination class pair was attacked.

4.1. Results

The success rates and adversarial probability labels for one-pixel perturbations on three networks are shown in Table 4 and the comparison of one, three and five-pixel perturbations is shown in Table 5. The number of target classes is shown by Figure 4. The number of original-target class pairs is shown by the heat-maps of Figure 5 and 6. 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 Figure 7.

4.1.1 Success Rate and Adversarial Probability Labels (Targeted Attack Results)

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 the case of the All Convolution network, 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 98.7%, 96.11% and 97.93% respectively to one, three and five-pixel attacks. Thus, increasing the number of modified pixels does not significantly improve the confidence values.

4.1.2 Number of Target Classes (Non-targeted Attack Results)

Regarding the results shown in Figure 4, 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

	1 pixel	3 pixels	5 pixels
Success rate(tar)	23.46%	35.52%	42.78%
Success rate(non-tar)	73.8%	82.0%	87.3%
Accumulated labels	23.08%	35.08%	42.36%
Rate/Labels	98.37%	98.76%	99.02%

Table 5. Results of conducting one, three and five-pixel attack on AllConv networks.

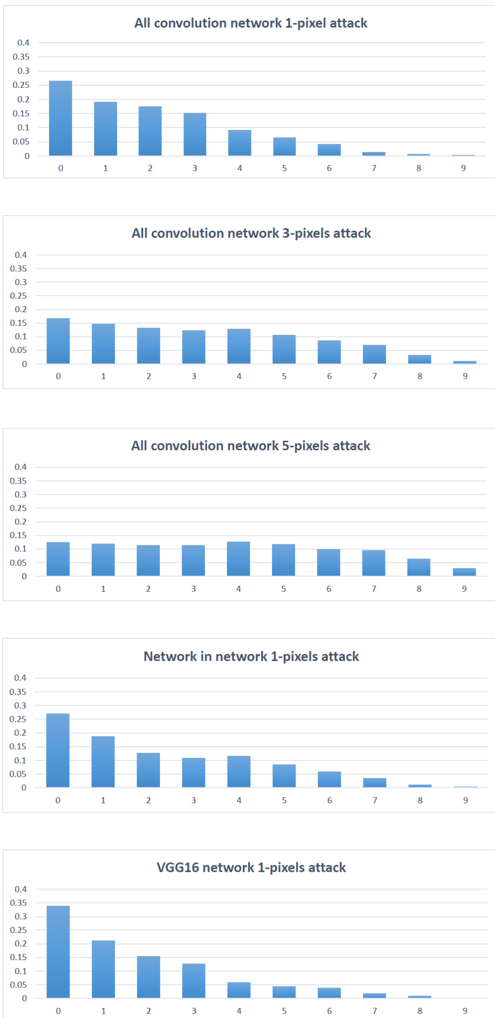

Figure 4. The graphs shows the percentage of natural images that were successfully perturbed to a certain number (from 0 to 9) of target classes by using one, three or five-pixel perturbation. The vertical axis shows the percentage of images that can be pertubated while the horizontal axis indicates the number of target classes.

increasing the number of pixels modified, perturbation to more target classes becomes highly probable. In the case of non-targeted 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,

Method	Success rate	Confidence	Number of pixels	Network
Our method	73.04%	96.11%	1(0.098%)	NiN
Our method	66.08%	97.93%	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

Table 6. Compassion of non-targeted attack effectiveness between the proposed method and two previous works. This suggests that one pixel is enough to create adversarial samples from most of the natural images.

NiN and VGG16) are vulnerable to this type of attack.

The results of attacks are competitive with previous non-targeted attack methods which need much more distortions (Table 6). It shows that using one dimensional pertubation 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 Figure 8.

4.1.3 Original-Target Class Pairs

Some specific original-target class pairs are much more vulnerable than others (Figure 5 and 6). 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 that others since their 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

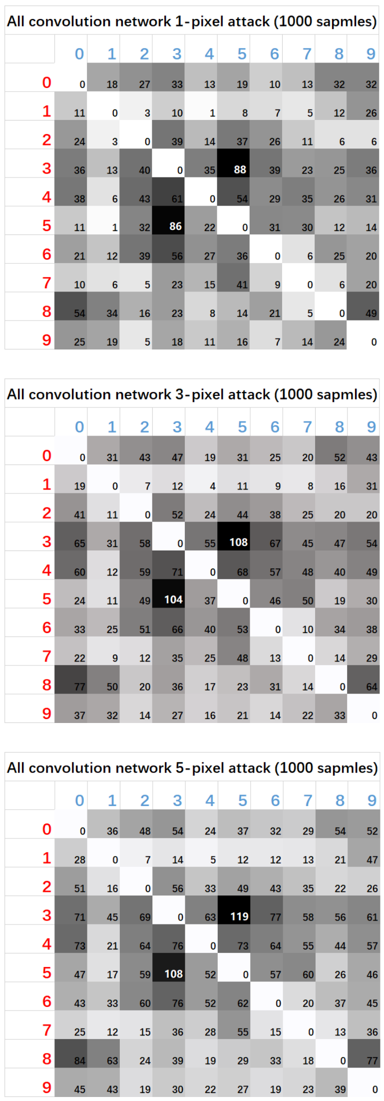

Figure 5. Heat-maps of the number of times a successful attack is present with the corresponding original-target class pair in one, three and five-pixel attack cases on AllConv network. Red and blue indices indicate respectively the original and target classes. The number from 0 to 9 indicates respectively the following classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

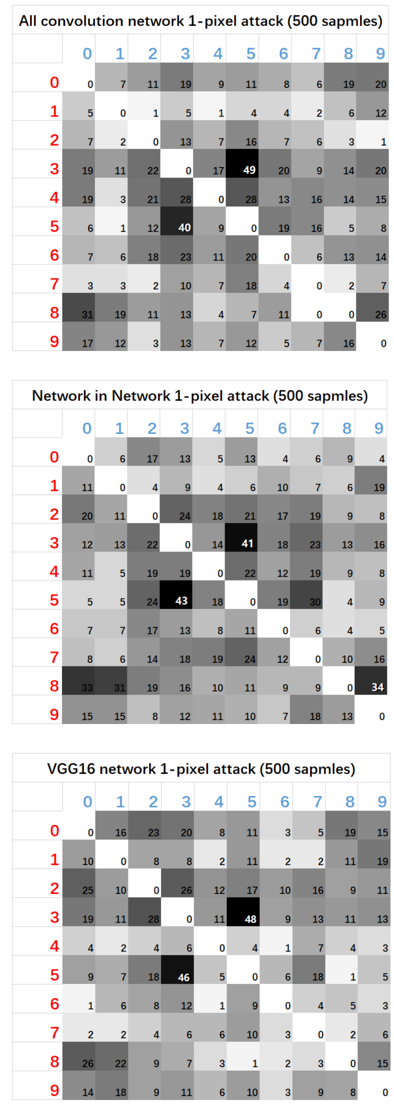

Figure 6. Heat-maps for one-pixel attack on three types of networks.

class has similar number of adversarial samples which were crafted from these classes as well as to these classes (Figure 7). 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 samples from them while it is relatively hard to craft adversarial samples 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 (a) boundary shape and (b) how close are natural images to the boundary. In other words, if the boundary shape is wide enought 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 here, however, the exceptions are not shared between the networks, revealing that whatever is causing the phenomenon

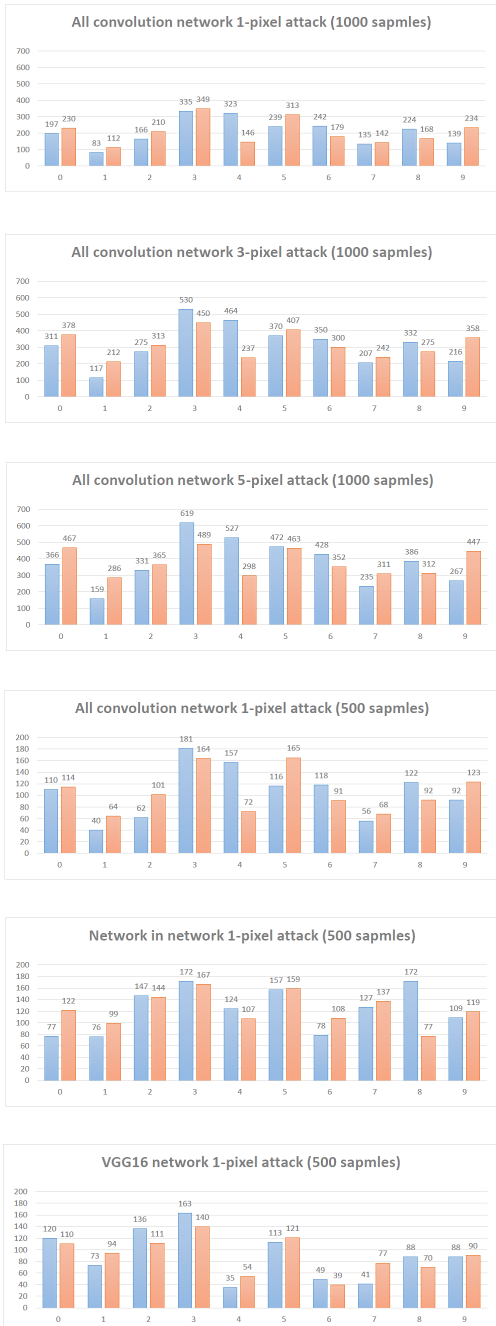

Figure 7. Number of successful attacks (vertical axis) for a specific class acting as the original (blue) and target (red) class. The horizontal axis indicates the index of each class which is the same as Figure 6.

is not shared. Therefore, for the current systems under the given attacks, such a vulnerability seems hard to be exploited.

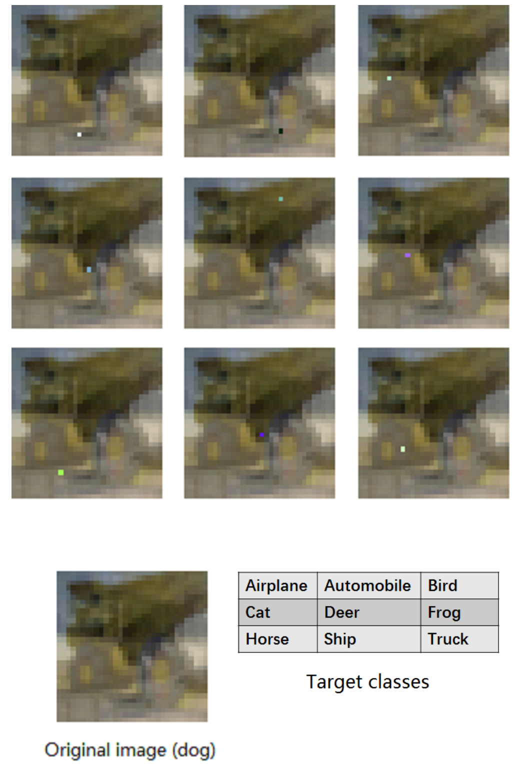

Figure 8. A natural image of the dog class that can be perturbed to all other nine classes. The attack is conducted over the All-Conv network using the proposed one pixel attack. The table in the bottom shows the class labels output by the target DNN, all with approximately 100% confidence. This curious result further emphasize the difference and limitations of current methods when compared to human recognition.

5. Discussion and Future Work

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 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 I. J. 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.

Actually, the results showed 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. Additionally, the proposed algorithm and the widely vulnerable samples (i.e. natural images that can be used to craft adversarial samples to most of the other classes) collected might be useful for generating better artificial adversarial samples in order to augment the training data set. This aids the development of more robust models[19] which is left for future work.

References

[1] M. Barreno, B. Nelson, A. D. Joseph, and J. Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

[2] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25. ACM, 2006.

[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 transactions on evolutionary computation*, 10(6):646–657, 2006.

[4] P. Civicioglu and E. Besdok. A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial intelligence review*, pages 1–32, 2013.

[5] H. Dang, Y. Huang, and E.-C. Chang. Evading classifiers by morphing in the dark. 2017.

[6] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31, 2011.

[7] M.-D. et al. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[8] M.-D. et al. 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. Technical report, Institute of Electrical and Electronics Engineers, 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. 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 *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-226156, 2017.

[15] N. Narodytska and S. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1318. IEEE, 2017.

[16] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

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

[19] A. Rozsa, E. M. Rudd, and T. E. Boult. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32, 2016.

[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 *ICLR (workshop track)*.

[23] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[24] C. e. a. Szegedy. Intriguing properties of neural networks. In *In ICLR*. Citeseer, 2014.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[26] D. V. Vargas and J. Murata. Spectrum-diverse neuroevolution with unified neural models. *IEEE transactions on neural networks and learning systems*, 28(8):1759–1773, 2017.

[27] D. V. Vargas, J. Murata, H. Takano, and A. C. B. Delbem. General subpopulation framework and taming the conflict inside populations. *Evolutionary computation*, 23(1):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 *In ICML Workshop on Deep Learning*. Citeseer.

[30] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.