Computers, Materials & Continua CMC, vol.64, no.1, pp.623-635, 2020

A Fast Two-Stage Black-Box Deep Learning Network Attacking Method Based on Cross-Correlation

Deyin Li[1, 2], Mingzhi Cheng[3], Yu Yang[1, 2, *], Min Lei[1, 2] and Linfeng Shen[4]

Abstract: Deep learning networks are widely used in various systems that require classification. However, deep learning networks are vulnerable to adversarial attacks. The study on adversarial attacks plays an important role in defense. Black-box attacks require less knowledge about target models than white-box attacks do, which means black-box attacks are easier to launch and more valuable. However, the state-of-arts black-box attacks still suffer in low success rates and large visual distances between generative adversarial images and original images. This paper proposes a kind of fast black-box attack based on the cross-correlation (FBACC) method. The attack is carried out in two stages. In the first stage, an adversarial image, which would be missclassified as the target label, is generated by using gradient descending learning. By far the image may look a lot different than the original one. Then, in the second stage, visual quality keeps getting improved on the condition that the label keeps being missclassified. By using the cross-correlation method, the error of the smooth region is ignored, and the number of iterations is reduced. Compared with the proposed black-box adversarial attack methods, FBACC achieves a better fooling rate and fewer iterations. When attacking LeNet5 and AlexNet respectively, the fooling rates are 100% and 89.56%. When attacking them at the same time, the fooling rate is 69.78%. FBACC method also provides a new adversarial attack method for the study of defense against adversarial attacks.

Keywords: Black-box adversarial attack, cross-correlation, two-module.

1 Introduction

In the past few years, machine learning has made great progress. Deep learning networks have become more and more effective, and more and more machine learning models are being used to help humans make crucial decisions: automatic driving, unmanned aerial vehicle, anomaly detection, malware, speech recognition, natural language processing, medical image analysis [Qasem, Nazar, Qamar et al. (2019); Zhao, Zhang, Shi et al.

[1] State Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, China.

[2] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China.

[3] College of New Media, Beijing Institute of Graphic Communication, Beijing, 102600, China.

[4] School of Computing Science, Simon Fraser University, Burnaby, Canada.

[*] Corresponding Author: Yu Yang. Email: yangyu@bupt.edu.cn.

Received: 19 January 2020; Accepted: 05 April 2020.

(2019)]. However, deep learning network has security problems. Szegedy et al. [Szegedy, Zaremba, Sutskever et al. (2013)] first noticed the existence of adversarial examples in image classification. Adding specially designed perturbations to the original samples caused the target model to make wrong predictions. The existence of adversarial samples can bring security risks to the real world. For example, adding perturbations to signs [Eykholt, Evtimov, Fernandes et al. (2017)] can make the deep learning network in automatic driving give wrong instructions, which leads to car operation errors and threatens human life. Adversarial attack will pose a serious threat to the application of deep learning model in practice.

In response to these threats, it is important to study adversarial attacks. This research began with a paper published by Szegedy et al. [Szegedy, Zaremba, Sutskever et al. (2013)]. This is the first time that a small disturbance to an image may prove to be fooling the classification model. Since then, many attack methods have been proposed, whether it is a white box attack or a black box attack.

White-box attacks launch attacks under the requirement of all or lots of target system information. In 2014, Szegedy et al. [Szegedy, Zaremba, Sutskever et al. (2013)] used the L-BFGS to solve the problem of perturbation generation for the first time. The L-BFGS method can find the difference between the original label of the image and the specified label. In 2015, Goodfellow et al. [Goodfellow, Shlens and Szegedy (2015)] proposed a more efficient perturbation generation method FGSM than the L-BFGS method, but both methods generate perturbation in one-step. In 2016, Carlini et al. [Carlini and Wagner (2016)] proposed an iterative attack method based on L1-norm, L0-norm and L2-norm, which can make the target model judge adversarial picture as the classification specified by the attacker. In 2019, Finlay et al. [Finlay, Pooladian and Oberman (2019)] proposed a gradient-based and generated LogBarrier attack method with the less visual difference between the adversarial image and the original image. But this method cannot make the target model judge the adversarial image as the classification specified by the attacker. In the white-box attack methods, only the Carlini and Wagner (C & W) method can make the target model judge the adversarial picture as the designated label, with a high fooling rate and small visual difference [Akhtar and Mian (2018)].

Black-box attacks launch attacks under the requirement of none or little of target system information. In 2017, Su et al. [Su, Vargas and Kouichi (2019)] proposed One Pixel attack, which can find a modified value of a pixel point and its RGB value enough to attack the whole image, so that it can fool the classification model to judge the image as a specified label. In 2017, Sarkar et al. [Sarkar, Bansal, Mahbub et al. (2017)] proposed Universal Perturbations for Steering to Exact Targets (UPSET) method that can fool multiple classification models at the same time to judge the adversarial images as the specified classification. In 2018, Wei et al. [Wei, Liang, Cao et al. (2018)] proposed a method for quickly generating adversarial images and videos based on Generative Adversarial Networks (GAN), but it could not convert images to specified labels. In this attack method, only the UPSET method can attack multiple target models and make the classification of adversarial image be judged as the specified classification.

Black-box attacks are more widely used than white-box attacks because they require less knowledge. It can be inferred that the UPSET method will have a high value in practical

applications. However, the UPSET method has two shortcomings.

1. The fooling rate is low. When using UPSET method to attack the five-layer convolutional deep learning network M1, M2 and the three-layer fully connected network M3 given by Sarkar, the fooling rate is 70.53%, 73.03%, and 56.29% respectively. When using the UPSET method to attack the LeNet5 and AlexNet networks, the fooling rate was only 40.33% and 11.11%.

2. The visual difference between the adversarial image and the original image is large. When using the UPSET method to attack the M1, M2 and M3 given by Sarkar, the average residual norms of the generated adversarial images are 1.29, 1.28, 1.09.

In order to overcome the shortcomings of the UPSET, this paper proposes a new fast black-box attack based on cross-correlation, and it is called as FBACC (Fast Black-box Attack based on Cross-Correlation) for abbreviation. The contributions of the methods are as follows.

1. This paper analyzes the shortcomings of the current attack methods and designs a fast, efficient black-box attack method, FBACC, that can attack multiple target models at the same time.

2. We find that the attack accuracy is lower when the attack network layer is deeper and the classification accuracy is higher. It provides an idea for the defense of an adversarial attack.

3. We provide a new attack method for defending adversarial attack methods, which can be used to test the effect of the defense method.

In order to evaluate the effect of the FBACC method, we construct two typical convolutional deep learning networks LeNet5 and AlexNet on the MNIST dataset, and achieve 98.5% and 99.5% classification accuracy respectively. Our FBACC method achieves 100.00% and 89.56% fooling rate when attacking the LeNet5 and AlexNet models respectively. When attacking both AlexNet and LeNet5 at the same time, it achieves a fooling rate of 69.78%.

The rest of this paper consists of 4 sections. The Background section introduces the threat and explains some nouns and symbols. The FBACC Method section introduces our method. The Result Evaluation section compares our method with the rand UPSET. The Conclusion and Outlook section summarizes the contributions of this paper and proposes the next possible research directions.

2 Background

2.1 Threat model

More and more machine learning models are used in decision-making scenarios, such as automatic driving, face recognition [Ghazi and Ekenel (2016)], image analysis [Litjens, Kooi, Bejnordi et al. (2017)], intrusion detection [Davis and Clark (2011)], etc. In these scenarios, the deep learning network model needs to classify the samples correctly. Therefore, the correct classification of the deep learning network model is the basis for these tasks to be applied. However, the existing deep learning network system is very vulnerable to attack. Szegedy et al. [Szegedy, Zaremba, Sutskever et al. (2013)] first discover that the adversarial perturbation in the image will make the deep learning network unable to correctly classify the image. This affects the use of deep learning

networks in various scenarios. Therefore, the deep learning network should not only pay attention to the correct rate of its classification, but also pay attention to its ability to resist attacks. In order to help deep learning network users check whether their deep learning networks can resist adversarial attacks, we design a fast black-box attack method: FBACC which can attack multiple deep learning networks at the same time.

2.2 Deep learning networks and notation

We regard a deep learning network as a function F(x)=y, where x is the input of the network and y is the output of the neural network. Since our method is a black-box attack method, only the x and y information is known. For an m-class classifier, the output y is a vector of m length. Each position in y indicates the relative probability or relative value of x belonging to the corresponding label. When using softmax function, the output is a relative probability. When not using softmax function, the output is a relative value. Our black-box attack method does not distinguish whether it uses softmax or not. The classifier performs an Argmax operation on the network output y, C(x)=argmax(y), to find the maximum value in y and return its index, that is, the network judges the category x belongs to. We use grey-scale images, this kind of simple 2D image, as input x. The simpler the image, the easier it is to be classed by the network, correspondingly, the more difficult the attack is. Attacking the target that is difficult to attack can better reflect the effects of our method.

2.3 Adversarial image

Szegedy et al. [Szegedy, Zaremba, Sutskever et al. (2013)] first proposed the concept of adversarial images/examples in the paper published in ICLR 2014: Applying an imperceptible non-random perturbation to a test image, it is possible to arbitrarily change the network's prediction. This kind of test image is called adversarial example by Szegedy et al. Since the samples we use are pictures, we call them adversarial pictures and express them with x'. "The imperceptible non-random perturbation" we call it perturbation, expressed in δ. "The test image" we call it the original picture, denoted by x. "The network" we call it target model. Target model judges the classification of x we use t to indicate. The classification that we want the target model class x' to belong to we use l to indicate. In our evaluation, only when target model judges x' as an l classification can we think that the target model was fooled.

2.4 Distance metrics

In our definition of adversarial images, we require the use of a distance metric to quantify the similarity. The current common distance metric measures give the same weight to all pixels in the image, which is not in line with the method that human beings focus on observing the main lines of the images when judging the images classification. So we investigated several new methods and evaluated them: Cross-Correlation, Minkowsky Measures, Angles Correlation Measures and HVS Based Measure. And choose the best way from them.

3 FBACC method

Equations and mathematical expressions must be inserted into the main text. Two

different types of styles can be used for equations and mathematical expressions. They are in-line style and display style.

3.1 Target models setup

Before developing the black-box attack algorithms, we describe how we train the models on which we will evaluate our attacks.

LeNet5 [Lecun, Bottou, Bengio et al. (1998)] and AlexNet [Krizhevsky, Sutskever and Hinton (2012)] are trained for the MNIST classification tasks. MNIST is a database of handwritten digits that was created by Yann LeCun, Corinna Cortes and Christopher J. C. Burges. It has a training set of 60,000 examples and a test set of 10,000 examples. The label of each picture is represented by one-hot encoding. In the same way as the processing of MNIST in UPSET method, the pixel values in the pictures of MNIST are clipped to [-1,1].

LeNet5 is a convolutional deep learning network applied in MNIST, which was proposed by Lecun et al. in 1998 and has been widely concerned. AlexNet was designed by Hinton and his student Alex Krizhevsky who won the Imagenet competition in 2012. AlexNet successfully applied ReLu, dropout and LRN in CNN for the first time. We modify its input to make it applicable to MNIST.

LeNet5 is trained in the same way as Lecun et al. And the Adam optimizer is used when LeNet5 is trained. The trained LeNet5 network gets a 98.5% correct rate in MNIST. AlexNet was trained in the same way and gets 99.5% correct rate in MNIST test set.

3.2 FBACC method

Now let's turn to the construction of our method. Our basic problems are similar to those of Szegedy et al. [Szegedy, Zaremba, Sutskever et al. (2013)]:

$$minimize\ D(x, x')\ such\ that\ C(x') = l\ and\ C(x) = t\ x, x' \in [-1,1]^n \quad (1)$$

x is the original picture, x' is the picture after adding perturbation i.e., adversarial image. $\mathrm{C}(x)$ denotes the target model, which is the network to be attacked. t is the classification of target model to x. l is the classification of target model to x', that is, target classification.

Our target is to generate x' that satisfies $\mathrm{C}(x') = l$ and $\mathrm{C}(x) = t$ and minimizes $\mathrm{D}(x, x')$. D (.) is a function that calculates the distance metric between two pictures.

3.2.1 The method to add perturbation

In order to generate a suitable x' to minimize $\mathrm{D}(x, x')$, the relationship between x' and perturbation δ, as well as the original picture x, must be determined first. In the UPSET method,

$$x' = \mathrm{U}(x, l) = \max(\min(s \times \mathrm{R}(l) + x, 1), -1) \quad (2)$$

where R (.) is the method of generating the perturbation δ with the same shape as the original picture, and s is a constant. Then clips the x' to [-1,1]. After that, R(.) is trained iteratively to get the most suitable x'. In the optimization literature, this is known as a "box

constraint". UPSET method uses the projected gradient descent method, which performs one step of standard gradient descent, and then clips all the coordinates to be within the box.

There are two disadvantages in this method of generating adversarial image x':

1. It cannot reduce the proportion of original picture x in x'. When calculating the minimum $\mathrm{D}(x, x')$ and making the targeted model judge the probability that x' is l is large enough, that is, when $\mathrm{C}(x') = l$ and $\mathrm{C}(x) = t$, the coefficient of x may be less than 1.

2. The UPSET method uses R (.) function, to generate perturbation and optimize the output of R (.) to change the perturbation. It is not direct enough to optimize the perturbation δ quickly and accurately.

For the first disadvantage, we set the corresponding coefficient *k* before x, that is,

$$x' = \mathrm{U}(\mathrm{x}, \mathrm{l}) = \max\left(\min(s \times \delta + k \times x, 1), -1\right) \quad (3)$$

δ is a perturbation with the same shape as the original picture. *s* and *k* are perturbation factor and maintenance factor. The larger the value of the perturbation factor is, the larger the proportion of perturbation in the adversarial picture is, and the greater the distance metric between the adversarial picture and the original picture is. Maintenance factor is used to maintain the proportion of the original image in the adversarial image. When we train δ iteratively, we also train perturbation factor and maintenance factor iteratively. In order to minimize $\mathrm{D}(x, x')$, according to experience, the most appropriate maintenance factor should be about 1.0, so as to ensure that x and x' are sufficiently similar. The larger the perturbation factor is, the larger the proportion of δ in x', the easier it is to change the classification that x' is classified by target model.

For the second disadvantage, we do not generate the perturbation δ through any function, and directly train the perturbation δ iteratively, which is more direct.

We fooled the LeNet5 450 times. The relationship between the initial value of perturbation factor and maintenance factor, whether they are optimized and the result is shown in Tabs. 1 and 2. "Train" indicates whether they are trained iteratively. "Rate" indicates the fooling rate. "Iterations" indicates the average number of iterations. "SD" indicates the standard deviation of iterations. It is a measure of the amount of variation or dispersion of iterations. "PSNR" indicates the average peak signal to noise ratio.

Table 1: The influence of s and k on the results-1

(s, k)	(3.0, 1.0)	(2.5, 1.0)	(2.0, 1.0)	(1.5, 1.0)	(1.0, 1.0)
Train	YES	YES	YES	YES	YES
Rate	100.00%	100.00%	99.78%	96.67%	76.22%
Iterations	209.06	229.20	288.20	370.10	531.39
SD	32.19	88.42	113.09	148.45	193.60
PSNR	10.51	11.21	11.80	12.65	10.71

From Tabs. 1 and 2, we can know. When the initial value of perturbation factor increase, the fooling rate will increase. When the initial value is the same, iterative training for

perturbation factor and maintenance factor will get better attack effect. Therefore, in order to achieve the best attack effect, we should select a larger perturbation factor and train perturbation factor and maintenance factor iteratively.

Table 2: The influence of s and k on the results-2

(s, k)	(3.0, 1.0)	(2.5, 1.0)	(2.0, 1.0)	(1.5, 1.0)	(1.0, 1.0)
Train	NO	NO	NO	NO	NO
Rate	100.00%	100.00%	99.56%	96.67%	76.22%
Iterations	78.72	124.98	192.99	385.73	561.08
SD	40.56	56.09	91.94	174.02	190.87
PSNR	2.44	6.38	10.89	12.36	10.54

3.2.1 Training module and loss function

We find that whether the UPSET method can change the classification of x' to l depends on the first several iterations of its iteration. When the label cannot be changed in the current rounds, the subsequent iterations will only reduce the value of $\mathrm{D}(x, x')$, and will not meet the condition of $\mathrm{C}(x') = l$. Therefore, for this problem, we design an iterative training method which is divided into two modules: classification change module and image distance reduction module. These two modules solve the following problems respectively:

1. When iterative training, the adversarial picture cannot be judged as l classification by target model, that is, x' cannot meet the condition of $\mathrm{C}(x') = l$.

2. Under the condition of keeping $\mathrm{C}(x') = l$, the problem of finding the value of minimizing $\mathrm{D}(x, x')$.

Classification change module

The main goal of the Classification change module is to solve the problem that when iterative training, the adversarial picture cannot be judged as l classification by target model, that is, x' cannot satisfy the condition of $\mathrm{C}(x') = l$. In order to make x' be judged as l by target model, we use the same loss function as in UPSET method:

$$L1 = -log\left(F(x')\right) \tag{4}$$

$F(x')$ represents the relative probability that the target model judge x' to be l, $F(x') \in [0,1]$. The larger the $F(x')$ value is, the smaller the L1 value is, and the easier the $\mathrm{C}(x') = l$ condition is satisfied. By reducing the loss function L1, $F(x')$ can be increased, and the probability of $\mathrm{C}(x') = l$ condition being satisfied can be increased, thereby increasing the fooling rate of the FBACC method. We used the complete FBACC method and the FBACC method without the Classification change module to perform 450 attacks on the LeNet5, and then compare the attack results in Tab. 3. ("Include" indicates that the Classification change module is included, and "Exclude" indicates that the Classification change module is not included.)

Table 3: The influence of Classification change module

	Include	Exclude
Rate	100.00%	48.22%
Iterations	229.20	308.25
SD	88.42	94.68
PSNR	11.21	4.35

From the experimental results, we can see that the success rate of our method has increased from 48.22% to 100.00%, and the average number of iterations has dropped from 308.25 to 229.20. At the same time, the PSNR has also increased from 4.35 to 11.21. The performance of FBACC method has been greatly improved after adding Classification change module, which shows that adding Classification change module to FBACC method can improve the effect of FBACC method.

Image distance reduction module

The problem to be solved by the image distance reduction module is to find the value of minimize $\mathrm{D}(x, x')$ while keeping $\mathrm{C}(x') = l$. To solve this problem, we designed the following loss function for the Image distance reduction module:

$$\mathrm{L2} = \mathrm{L1} + \mathrm{D}(x, x') \tag{5}$$

where L1 is the loss function L1 of the Classification change module. For the representation of $\mathrm{D}(x, x')$, we investigated some methods to represent $\mathrm{D}(x, x')$: Cross-Correlation, Minkowsky Measures, Angles Correlation Measures [Trahanias, Karakos and Venetsanopoulos (1996)] and HVS Based Measure [Watson (1993); Nill (1985); Avcibas, Memon and Bülent (2003)]. They are shown in Tab. 4. (In Angles Correlation Measures, x and x' are converted into two matrices of $N^2/2$ rows and 2 columns. x_i represents the ith row of converted x.)

Table 4: The method of $\mathrm{D}(x, x')$

Name	Formulas
Cross-Correlation	$D(x,x') = -\left(\sum_{i,j}^{N}\left(x_{i,j} \times x'_{i,j}\right)\right) \Big/ \left(\sqrt{\sum_{i,j}^{N} {x_{i,j}}^2}\sqrt{\sum_{i,j}^{N} {x'_{i,j}}^2}\right)$
Minkowsky Measures (r=1, 2)	$D_r(x,x') = \left\{\frac{1}{N^2}\sum_{i,j=1}^{N}\left\|x_{i,j} - x'_{i,j}\right\|^r\right\}^{1/r}$
Angles Correlation Measures	$D(x,x') = cos^{-1}\left\{\frac{2}{N^2}\sum_{i=1}^{N^2/2}\frac{\langle x_i, x'_i\rangle}{\lvert x_i\rvert \left\lvert x'_i\right\rvert}\right\}$

HVS Based Measure	$H(p) = \begin{cases} 0.05e^{p^{0.554}} & p < 12.25 \\ e^{-9(log_{10}p - log_{10}9)^{2.3}} & p \geq 12.25 \end{cases} \quad p = \sqrt{i^2 + j^2}$ $U(x) = IDCT(DCT(x) * H)$ $D(x, x') = \frac{\sum_{i,j=1}^{N} \left[U\{x_{i,j}\} - U\left\{x'_{i,j}\right\}\right]^2}{\sum_{i,j=1}^{N} \left[U\{x_{i,j}\}\right]^2}$

For the above five methods (in the Minkowski Measures, r=1 and 2 are considered as two methods), we use them to conduct 450 attacks on LeNet5 and AlexNet. And then compare the fooling rate, the average number of iterations, the standard deviation of iterations, adversarial images and the peak signal to noise ratio between x and x'.

As far as the fooling rate and the average number of iterations are concerned, the method using the Cross-Correlation method is more effective than the other methods both for LeNet5 and AlexNet. Although in terms of PSNR, the effect of using Cross-Correlation to represent $D(x, x')$ is not as good as that of Minkowsky Measures r=2, the Minkowsky Measures r=2 method gives the same weight to the pixel values of all positions in the image, which is not in line with the study of HVS. When humans judge the classification of an image, the noise in the smooth area will be ignored in human cognition. The Cross-Correlation function ignores the noise in the smooth region. Moreover, in terms of fooling rate and average number of iterations, the Cross-Correlation method is much better than Minkowsky Measures r=2. In summary, using the Cross-Correlation method to represent $D(x, x')$ will make the best effect of FBACC method. So we use a cross-correlation method to represent $D(x, x')$.

Table 5: Using different loss functions to attack LeNet5

	Cross	Minkowsky	Minkowsky	Angles	HVS
Rate	100.00%	99.56%	99.56%	99.78%	99.78%
Iterations	229.20	587.19	324.66	529.83	287.67
SD	88.42	82.52	102.39	352.12	80.14
PSNR	11.21	10.87	12.68	10.86	9.78

Table 6: Using different loss functions to attack AlexNet

	Cross	Minkowsky	Minkowsky	Angles	HVS
Rate	89.56%	75%.11	65.00%	57.78%	53.78%
Iterations	652.52	832.16	656.06	502.74	484.72
SD	112.99	89.10	108.39	350.48	85.45
PSNR	7.40	5.68	8.03	5.54	4.88

4 Result evaluation

4.1 Compared with C & W and UPSET

We reproduce the UPSET method and C&W method. We use them to perform 450 attacks on the trained AlexNet and LeNet5, respectively. The effect of the attack method is evaluated from four aspects: the fooling rate, the average number of iterations, the average peak signal to noise ratio, and the black/white-box attributes of the attack method. For the iterations, in the UPSET method, the authors perform 25 iterations. In the C & W method, the authors perform 20 iterations of binary search over "c". For each selected value of "c", they run 10,000 iterations of gradient descent with the Adam optimizer [Akhtar and Mian (2018)]. So the C & W method needs 200000 iterations in total.

Fooling rate: Fooling rate indicates the percentage of images on which a trained model changes its prediction classification after the images are perturbed. The fooling rate is equal to the success rate of an adversarial attack method, which is the most important evaluation index of an adversarial attack method.

Average number of iterations: FBACC method, C & W method, UPSET method and many other methods generate adversarial images by iteration. The average number of iterations can evaluate the performance of an adversarial attack method to produce adversarial images: First, to understand if the performance would be prohibitive for an adversarial to actually mount the attacks, and second, to be used as an inner loop in adversarial retraining [Goodfellow, Shlens and Szegedy (2015)].

Table 7: Using different adversarial attacks to attack LeNet5

	FBACC		C & W		UPSET	
Target model	LeNet5	AlexNet	LeNet5	AlexNet	LeNet5	AlexNet
Rate	100.00%	89.56%	100.00%	92.89%	40.33%	11.11%
Iterations	229.20	652.52	200000	200000	25	25
PSNR	11.21	7.40	23.92	16.33	24.92	29.81
black/white-box attack	black		white		black	

PSNR: It is used to quantify the difference between two images. The larger the peak signal to noise ratio, the smaller the distance metric between the two pictures, that is, the smaller $D(x, x')$ is.

FBACC and C & W can both achieve a 100.00% fooling rate when attacking LeNet5 and a 90% fooling rate when attacking AlexNet.

1. In terms of the number of iterations, the number of iterations of the FBACC is much smaller than that of the C & W, which shows that the attack efficiency of the FBACC method is higher than that of the C&W method.

2. In terms of black/white-box attack attributes, the C & W method is a white-box attack method [Akhtar and Mian (2018)], and the attacker needs to know some information about the target model. But our FBACC method is a black-box attack method, only need to know the output of the target model. This means that the FBACC method has broader application prospects than the C & W method.

Compared with the UPSET method: They are all black-box attack methods, but whether it is attacking the letet5 model or attacking the AlexNet model, the FBACC's fooling rate exceeds the UPSET's fooling rate by more than 60%. Because the UPSAT method will reduce the difference between the original image and the adversarial image when it fails, the PSNR is relatively large.

When the fooling rate is very different, it is meaningless to compare the PSNR and the number of iterations.

From the results in Tab. 7, it can be seen that the attack effect of several attack methods when attacking AlexNet model is worse than that of attacking LeNet5. Compared with LeNet5, AlexNet has more network layers and higher classification accuracy. It shows that when attacking networks with deep layers and high classification accuracy, the fooling rate of attack methods will be relatively poor, which provides a new idea for deep learning networks to resist adversarial attacks.

4.2 Attack multiple target models at the same time

Similar to the UPSET method, our method can also attack multiple target models at the same time. Only L1 needs to be modified to include multiple target models at the same time.

$$\mathrm{L1} = -\sum_{i=1}^{M} \log\left(F_i(x')\right) \tag{6}$$

M indicates the number of target models to be attacked, and $F_i(x')$ indicates the relative probability or value of the ith target model to determine that x' is l. L1 in the loss function L2 also changes accordingly, and the remaining modules do not need to be changed. The results of FBACC attacking AlexNet and LeNet5 at the same time are compared with those of UPSET attacking AlexNet and LeNet5 at the same time.

Table 8: The result of attacking both AlexNet and LeNet5 at the same time

	FBACC	UPSET
Rate	69.78%	10.00%
Iterations	437.68	25
PSNR	6.36	9.31

Attacking two target models makes it difficult to generate a perturbation that can simultaneously fool two target models, so the fooling rate is lower than attacking a separate target model. However, compared with the UPSET method, FBACC method has a great improvement, and the fooling rate is about 60% higher than the UPSET method, reaching 69.78%.

5 Conclusion and outlook

This paper reproduces the C&W method and the UPSET method, and analyzes the shortcomings of these methods, and then proposes targeted solutions for these shortcomings. This paper designs a fast, efficient black-box attack method, FBACC, that can attack multiple target models at the same time. This paper provides a new attack

method for the defense of adversarial attacks, which can be used to test the effect of the defense method.

Although the FBACC method can perform an adversarial black-box attack against a convolutional deep learning network classifier and fools the classifier to classify the adversarial pictures into a specific classification. However, every time the original image is changed, the perturbation needs to be regenerated, which is undoubtedly inefficient. If an attack method capable of generating a perturbation that is suitable for various pictures can be proposed, the practical application of the method will have great prospects.

Funding Statement: This work is supported by the National Key R & D Program of China (2017YFB0802703), Research on the education mode for complicate skill students in new media with cross specialty integration (22150117092), Major Scientific and Technological Special Project of Guizhou Province (20183001), Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ014), Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ019) and Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ022).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Akhtar, N.; Mian, A. (2018): Threat of adversarial attacks on deep learning in computer vision: a survey. *IEEE Access*, vol. 6, pp. 14410-14430.

Avcibas, I.; Memon, N. D.; Bülent, S. (2003): Steganalysis using image quality metrics. *IEEE Transactions on Image Processing*, vol. 12, pp. 221-229.

Carlini, N.; Wagner, D. (2016): Towards evaluating the robustness of neural networks. https://arxiv.org/abs/1608.04644.

Davis, J. J.; Clark, A. J. (2011): Data preprocessing for anomaly based network intrusion detection: a review. *Computers & Security*, vol. 30, no. 6-7, pp. 353-375

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A. et al. (2017): Robust physical-world attacks on deep learning models.

https://arxiv.org/abs/1707.08945.

Finlay, C.; Pooladian, A. A.; Oberman, A. M. (2019): The log barrier adversarial attack: making effective use of decision boundary information.

https://arxiv.org/abs/1903.10396.

Ghazi, M. M.; Ekenel, H. K. (2016): A comprehensive analysis of deep learning based representation for face recognition. https://arxiv.org/abs/1606.02894.

Goodfellow, I., Shlens, J., Szegedy, C. (2015): Explaining and harnessing adversarial examples. https://arxiv.org/abs/1412.6572.

Krizhevsky, A.; Sutskever, I.; Hinton, G. (2012): ImageNet classification with deep convolutional neural networks. *Advances Neural Information Processing Systems*, vol. 1, pp. 1097-1105.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. (1998): Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324.

Litjens, G.; Kooi, T.; Bejnordi, B. E.; Setio, A. A. A.; Ciompi, F. et al. (2017): A survey on deep learning in medical image analysis.

https://arxiv.org/abs/1702.05747.

Nill, N. (1985): A visual model weighted cosine transform for image compression and quality assessment. *IEEE Transactions on Communications*, vol. 33, no. 6, pp. 551-557.

Qasem, S. N.; Nazar, A.; Qamar, A.; Shamshirband, S.; Karim, A. (2019): A learning based brain tumor detection system. *Computers, Materials & Continua*, vol. 59, no. 3, pp. 713-727.

Sarkar, S.; Bansal, A.; Mahbub, U.; Chellappa, R. (2017): UPSET and ANGRI: breaking high performance image classifiers. https://arxiv.org/abs/1707.01159.

Su, J.; Vargas, D. V.; Kouichi, S. (2019): One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828-841.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, J. et al. (2013): Intriguing properties of neural networks. https://arxiv.org/abs/1312.6199.

Trahanias, P. E.; Karakos, D.; Venetsanopoulos, A. N. (1996): Directional processing of color images: theory and experimental results. *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 868-880.

Watson, A. B. (1993): *Digital Images and Human Vision*. MIT Press.

Wei, X.; Liang, S.; Cao, X.; Zhu, J. (2018): Transferable adversarial attacks for image and video object detection. https://arxiv.org/abs/1811.12641.

Zhao, G. D.; Zhang, Y. W.; Shi, Y. Q.; Lan, H. Y.; Yang, Q. (2019): The application of BP neural networks to analysis the national vulnerability. *Computers, Materials & Continua*, vol. 58, no. 2, pp. 421-436