Information Sciences 550 (2021) 285–296

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Black-box adversarial attacks by manipulating image attributes

Xingxing Wei *, Ying Guo, Bo Li

School of Computer Science and Engineering, Beihang University, Beijing, China

ARTICLE INFO

Article history:
Received 8 July 2020
Received in revised form 7 October 2020
Accepted 16 October 2020
Available online 4 November 2020

Keywords:
Adversarial attack
Adversarial attributes
Black-box setting

ABSTRACT

Although there exist various adversarial attacking methods, most of them are performed by generating adversarial noises. Inspired by the fact that people usually set different camera parameters to obtain diverse visual styles when taking a picture, we propose the adversarial attributes, which generate adversarial examples by manipulating the image attributes like brightness, contrast, sharpness, chroma to simulate the imaging process. This task is accomplished under the black-box setting, where only the predicted probabilities are known. We formulate this process into an optimization problem. After efficiently solving this problem, the optimal adversarial attributes are obtained with limited queries. To guarantee the realistic effect of adversarial examples, we bound the attribute changes using L_p norm versus different p values. Besides, we also give a formal explanation for the adversarial attributes based on the linear nature of Deep Neural Networks (DNNs). Extensive experiments are conducted on two public datasets, including CIFAR-10 and ImageNet with respective to four representative DNNs like VGG16, AlexNet, Inception v3 and Resnet50. The results show that at most 97.79% of images in CIFAR-10 test dataset and 98.01% of the ImageNet images can be successfully perturbed to at least one wrong class with only $\leqslant$300 queries per image on average.

1. Introduction

Although Deep Neural Networks (DNNs) have achieved great success in many applications [31,3,36], it is shown that DNNs are vulnerable to adversarial examples. Up to now, many kinds of adversarial attacks have been proposed. For example, C&W [2] and Deepfool [16] generate adversarial noises via optimization mechanisms. One-pixel attack [26] modifies only one pixel in an image to perform black-box attacks. UAP [15] outputs a universal adversarial perturbation to adapt many different images. Wei et al. [32] proposed the sparse adversarial perturbations to add on the selected key frames in a video. Adversarial examples not only appear in the digital environment but also in the physical world. Eykholt et al. [6] designed the Robust Physical Perturbations, which attach some graffiti on the "stop" traffic sign to fool the autonomous drive. Komkov and Petiushko [11] proposed the AdvHat to attack the ArcFace face recognition system.

In essence, all the above methods perform attacks by generating adversarial noises. Because humans' imperceptibility is a crucial indicator to measure the adversarial examples, the L_2 or L_∞ are usually utilized to bound the amplitude. In a word, such attacking methods aim to make the adversarial noises/perturbations as small as possible while keeping fooling the classifiers. We argue that the tiny noises/perturbations are just one way to guarantee the crypticity of adversarial examples. Whether there exist other kinds of adversarial attacks that don't arouse human's attention, either?.

* Corresponding author.
E-mail addresses: xxwei@buaa.edu.cn (X. Wei), yingguo@buaa.edu.cn (Y. Guo), boli@buaa.edu.cn (B. Li).

https://doi.org/10.1016/j.ins.2020.10.028
0020-0255/© 2020 Elsevier Inc. All rights reserved.

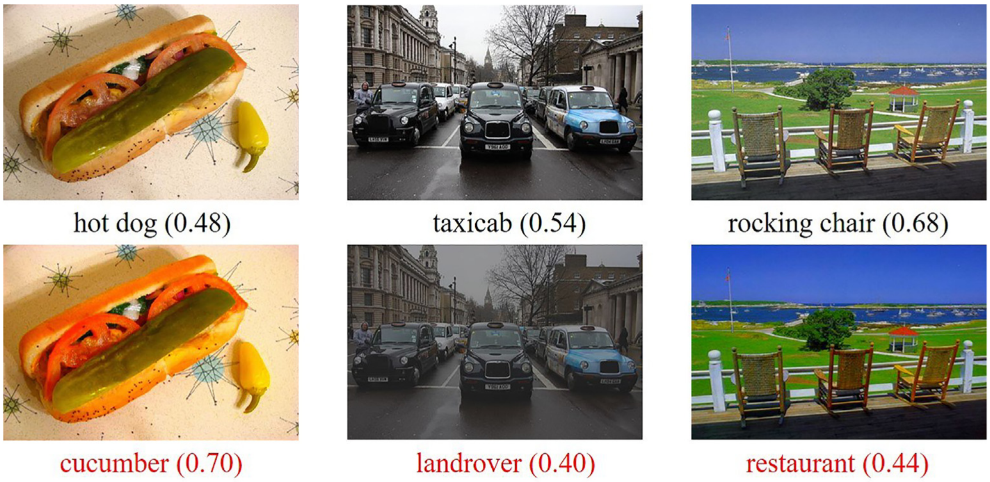

Fig. 1. Three ImageNet results achieved by adversarial attributes against Resnet50. For each column, the image above is the benign image, and the image below is the adversarial image by manipulating the attributes. The black text denotes the predicted correct class label and its probability. The red text denotes the predicted wrong class label and its probability. Here four attributes are jointly adjusted. They are brightness, contrast, chroma, and sharpness. Please see the images in the color mode. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We find that when taking a picture, people often set different camera parameters in consideration of diverse imaging effects and shooting environments, which makes the attributes of images taken for the same object be different. For the given images, these attributes refer to their brightness, contrast, sharpness, chroma, saturation, and so on. Fig. 2 shows such pictures taken with Canon camera under various brightness and contrast parameters.[1] When we take pictures for the same object using different camera parameters, can the well-trained models still get the correct results?.

To answer these two questions, in this paper, we propose the adversarial attributes, which generate adversarial examples by manipulating the images' brightness, contrast, sharpness, and chroma, etc (see Fig. 1). We believe that when we slightly adjust these attributes, the modified image usually does not arouse humans' suspicion. Because, as stated above, there exist no "ground-truth" attributes for an image, people often use different camera settings which makes the images have different attributes to present various visual effects according to their needs. Even using the "auto-mode" in the camera, different cameras will not show the consistent image attributes for the same object. This is a common phenomenon in our daily life.

We conduct the attacks in the black-box setting, where only the predicted probabilities obtained from the target model are known. This task is formulated as an optimization problem with constraint. The objective function is to find the optimal attribute changes to make the classifiers predict wrong labels, and the constraint is to bound the L_p norm of the change vector into a narrow range. We here set $p = 2$ and $p = \infty$, and conduct experiments, respectively. In the method section, we will introduce actually $p = 2$ and $p = \infty$ represent different adjusting strategies. $p = \infty$ means adjusting different image attributes independently, while $p = 2$ implies jointly adjusting the image attributes. The optimization problem is finally solved by Evolutionary Computation. Specifically, Differential Evolution (DE) is used. In DE, only the predicted probabilities are involved, which is consistent with the black-box setting.

In summary, this paper has the following contributions:

- We propose the adversarial attributes, a new way to perform reliable adversarial attacks while still having good crypticity. To our best knowledge, we are the first one to explore the usage of image attributes in the black-box adversarial attacks.
- We formulate the adversarial attribute attacks into an optimization problem with constraints (we propose two constraints: L_∞ and L_2 that represent different adjusting strategies), and present an effective solution (Differential Evolution) to solve it. More importantly, we give a formal explanation about why adversarial attributes can work in theory.
- We conduct a series of experiments, and the results show that at most 97.79% of images in CIFAR-10 test dataset and 98.01% of the ImageNet images can be successfully perturbed to at least one wrong class with only $\leqslant 300$ queries per image on average.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. We present the proposed adversarial attributes framework in Section 3. Section 4 reports all experimental results. Finally, we summarize the conclusions in Section 5.

[1] https://support.usa.canon.com/kb/index?page=content&id=ART172218.

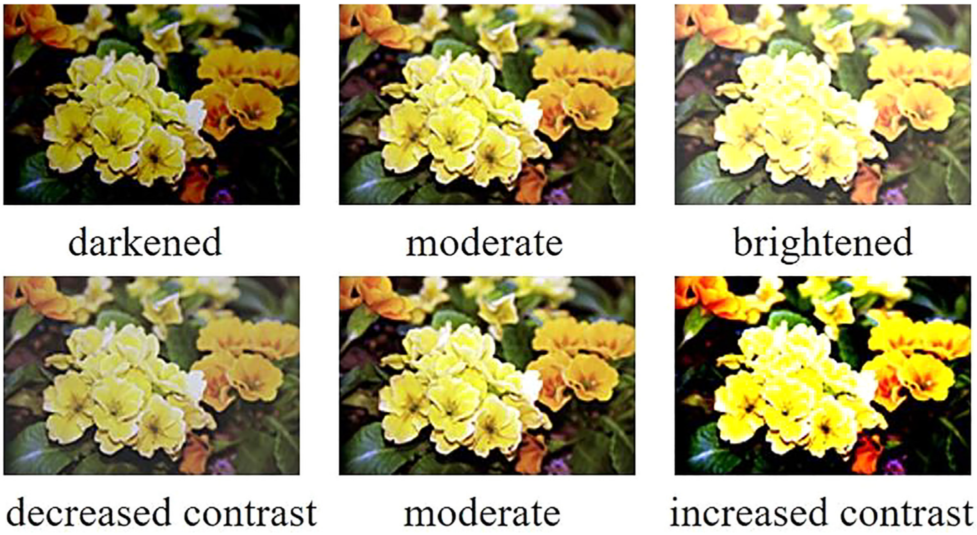

Fig. 2. Pictures taken in different settings using Canon camera's ScanGear mode. The first line is the result of adjusting the brightness, and the second line is the picture of adjusting the contrast. Please see the images in the color mode. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2. Related work

The related work comes from two aspects: adversarial attacks and adversarial defense.

2.1. Adversarial attacks

Deep Neural Networks (DNNs) are vulnerable to adversarial examples [28]. Up to now, many kinds of adversarial attacks have been proposed. These methods usually belong to the white-box attacks and black-box attacks.

In the white-box attacks, FGSM [7] is a classic method, which performs one-step attacks based on the gradient. Ada-FGSM [21] is an iterative attack that adaptively allocates step size of noises according to gradient information at each step. Besides, C&W [2] and Deepfool [16] generate adversarial noises via optimization mechanisms. Compared with FGSM, the optimization-based methods show better attacking ability. Adversarial examples can also be generated by the GAN [33]. Some follow-up works have been applied to attack other tasks [30,13]. Wei et al. [32] extended the adversarial attacks to the video data, and proposed the sparse adversarial perturbations to add on the selected key frames in a video. In addition, UAP [15] outputs a universal adversarial perturbation to adapt many different images.

In the black-box attacks, researchers usually utilize the transferability of adversarial attacks. Specifically, adversarial examples are generated based on one DNN model, and attack another DNN model. For example, to improve the transferability of FGSM, MI-FGSM [4] integrates the momentum to update the gradient direction. Results show MI-FGSM can indeed achieve better performance in the black-box setting. Effective black-box attacks can also be obtained by querying the target model. One-pixel attack [26] continuously selects and modifies one pixel in an image, and then queries the target model until the model is fooled. Customized Adversarial Boundary (CAB) [20] attack uses the current noise to model the sensitivity of each pixel and polish the adversarial noise of each image with a customized sampling setting. Shi et al. [22] used Curls iteration and Whey optimization to improve attack effect of iterative methods under black-box scenario.

Recent researches have carried out attacks from a more diverse perspective. Some researchers have shifted their focus from the digital to the physical environment. For example, Eykholt et al. [6] designed the Robust Physical Perturbations, which attach some graffiti on the "stop" traffic sign to fool the autonomous drive. Komkov and Petiushko [11] proposed AdvHat to attack the ArcFace face recognition system. Others argue that human perception is the determinant of disturbation imperceptibility. Zhao et al. [37] perturbed the image under the constraint of perceptual color (PerC) distance. SemanticAdv [18] changes the semantic attributes and introduces semantic interference naturally(e.g., expression or hair color of a portrait image) to craft the adversarial examples.

In essence, all the above pixel-wise methods perform attacks by generating adversarial noises and perception-level methods require more complex implementation mechanisms. Our adversarial attribute is different from them, and is a new way to finish the attacks.

2.2. Adversarial defense

To improve the DNNs' robustness, many adversarial defense methods are also proposed. Some methods defend adversarial examples by enhancing the model itself. Adversarial training [29] is one representative method, in which the adversarial examples are added into the training set to re-train the DNN model. B.S. et al. [1] proposed a single-step adversarial training method with dropout scheduling to improve the robustness of the model obtained from the adversarial training. Adversarial Training with Transferable Adversarial Examples(ATTA) [38] enhances the robustness of trained models and significantly improves the training efficiency by accumulating adversarial perturbations through epochs. Pang et al. [17] proposed a reverse corss-entropy loss to train the DNN model to detect adversarial examples. Some other methods defend adversarial examples by adding a pre-processing module to denoise the adversarial noises. Specifically, Jia et al. designed the ComDefend [10], which utilizes the compression to destroy the adversarial noises. Liao et al. presented the HGD [14]. They trained the denoise network guided by minimizing high-level representation. Other similar methods can be found in [24,35], and so on.

From the above descriptions, we can see these defense methods are constructed based on removing adversarial noises. In our method, we generate adversarial examples by manipulating adversarial attributes, which is totally different from adversarial noises. Therefore, in theory, the current defense methods cannot work well for adversarial attributes.

3. Methodology

3.1. Overview

The aim of this paper is to generate adversarial examples by manipulating the images' attributes like brightness, contrast, sharpness, and chroma, etc. If these attributes' changes are small, they will still be realistic, and will not arouse humans' suspicion. We conduct the attacks in the black-box setting, where only the predicted probabilities obtained from the target model are known. This task is formulated as an optimization problem with constraints. The objective function is to find the optimal attribute changes to make the classifiers predict wrong labels, and the constraint is to bound the L_p norm of the change vector into a narrow range. We here choose $p = 2$ and $p = \infty$ to bound the amplitude. The optimization problem is finally solved by Evolutionary Computation. Specifically, Differential Evolution (DE) is used, because DE doesn't use the gradient information, and therefore doesn't require the detailed structure and weights of the target model, which is consistent with the black-box setting.

3.2. Adversarial attributes

Assume $\mathbf{x} \in \mathbb{N}^d$ is a given image, where d is the number of pixels. Let t denote its ground-truth class label, and $\hat{t}$ denote a target class label different from t. $f(\cdot)$ represents the DNN model, and $f_t(\mathbf{x})$ is the predicted probability versus the ground-truth label t. $\mathbf{r}$ is the adversarial noise. In the existing methods, $\mathbf{r}$ is directly added on the image $\mathbf{x}$. They are searching for the optimal $\mathbf{r}^*$. In this way, generating adversarial examples in case of the targeted attack can be formalized as an optimization problem with constraints as follows [26]:

$$\mathbf{r}^* = \arg\max_{\mathbf{r}} \; f_{\hat{t}}(\mathbf{x} + \mathbf{r}), \quad s.t. \quad \|\mathbf{r}\|_p \leqslant \epsilon, \tag{1}$$

where p can be set to 0, 1, 2 or ∞. ϵ is a small value to bound the amplitude of the adversarial noises.

In our method, we manipulate the image attributes instead of the noises to generate adversarial examples. Therefore, Eq. (1) is modified as follows:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \; f_{\hat{t}}(g(\mathbf{x}; \boldsymbol{\theta})), \quad s.t. \quad \|\boldsymbol{\theta}\|_p \leqslant \epsilon, \tag{2}$$

where $g(\mathbf{x}; \boldsymbol{\theta})$ is the function that can manipulate image attributes, and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_i, \ldots, \theta_n)$ is the parameter to control the degree of attribute changes. θ_i is corresponding to the *i-th* attribute. n is the number of used attributes. For the un-targeted attack, the above formula is modified as:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; f_t(g(\mathbf{x}; \boldsymbol{\theta})), \quad s.t. \quad \|\boldsymbol{\theta}\|_p \leqslant \epsilon. \tag{3}$$

The function $g(\mathbf{x}; \boldsymbol{\theta})$ is defined as follows:

$$g(\mathbf{x}; \theta_i) = \theta_i * \mathbf{x} + (1 - \theta_i) * \bar{\mathbf{x}}_i, \quad i = 1, \ldots, n, \tag{4}$$

In [8], the authors state that brightness, contrast, saturation, tint, and sharpness can all be controlled with this unified formula. $\bar{\mathbf{x}}_i$ is a degenerate version of the image $\mathbf{x}$. For different image attributes, the degenerate image is different. For example, for sharpness, a blurred version of the original image is set as the degenerate image. Readers can refer to [8] for more details.

From Eq. (4), we can see that θ_i is linear with the output $g(\mathbf{x})$, a major $|\theta_i|$ will lead to a big change for the corresponding attribute. Therefore, in Eq. (2), we can apply the bound $\|\cdot\|_p$ on $\boldsymbol{\theta}$ to restrict the amplitude of image attribute changes. We here choose $p = 2$ and $p = \infty$. Actually, these two values represent different adjusting strategies.

When $p = \infty$, $\|\boldsymbol{\theta}\|_\infty = \underset{1 \leqslant i \leqslant n}{max} |\theta_i| \leqslant \epsilon$ is equal to $|\theta_i| \leqslant \epsilon$ for $i = 1, \ldots, n$. That means we can independently adjust each attribute to obtain the optimal $\boldsymbol{\theta}^*$ as long as θ_i changes in the range $-\epsilon \leqslant \theta_i \leqslant \epsilon$.

When $p = 2$, $\|\boldsymbol{\theta}\|_2 = \sqrt{\theta_1^2 + \ldots + \theta_n^2} \leqslant \epsilon$. Under this constraint, adjusting one attribute will affect other attributes. That means this case jointly adjusts all the attributes. In the experiments, we will compare these two strategies, and evaluate their own advantages.

Our setting is the black-box attacks. In this case, the DNN model $f(\cdot)$ is unknown, and we can only obtain the predicted probability when feeding an image to $f(\cdot)$. Therefore, the Stochastic Gradient Descent (SGD) algorithm cannot be used to solve Eq. (2). Instead, we utilize the Differential Evolution (DE) algorithm [25]. More details on using DE to find the appropriate image attribute parameters to achieve the goal of a successful attack are described in Section 3.3.

3.3. Differential evolution

DE performs population selection to explore the solution space efficiently. Because DE does not use the gradient information, it can be utilized on a broader range of optimization problems compared to gradient-based methods. In this process, each individual in the population corresponds to a solution vector, and an optimization target is given first as a comparison mechanism of individuals in the population. In each iteration, the candidate population is generated by crossover and mutation of the current population. Then DE compares the current population with the candidate population, leaving better individuals to form a new population.

In our case, we take the targeted attack as an example to illustrate the process of finding appropriate image attribute parameters by DE algorithm. $\boldsymbol{P}(k) = \left\{\boldsymbol{P}_i(k) | \theta_j^L \leqslant \boldsymbol{P}_{ij}(k) \leqslant \theta_j^U, 1 \leqslant i \leqslant NP, 1 \leqslant j \leqslant n\right\}$ represents the population, where NP is the population size and n is the number of genes per individual. $\boldsymbol{P}_{ij}(k)$ is the j-th gene value of the i-th individual in the k-th population. Specifically, each individual in the population refers to a tuple of image attribute change parameters, $\boldsymbol{P}_{ij}(k)$ corresponds to the j-th attribute. $\left(\theta_j^L, \theta_j^U\right)$ is the change range of the j-th image attribute. Eq. (2) is taken as the optimization objective, and the crossover way of generating candidate population is:

$$\begin{aligned} \boldsymbol{C}_i(k) = \boldsymbol{P}_{\gamma^*}(k) + \alpha\big(\boldsymbol{P}_{\gamma_1}(k) - \boldsymbol{P}_{\gamma_2}(k)\big) \\ \gamma^* \neq \gamma_1 \neq \gamma_2 \end{aligned} \tag{5}$$

where $\boldsymbol{C}_i(k)$ is the i-th individual in the k-th candidate population. α is the scale parameter set to be 0.5. γ^* is the index number of the best individual in the k-th population and γ_1, γ_2 are random numbers. It uses the difference vector of two individuals randomly selected from the population as the random change source of the third individual, and generates the mutation individual by weighting the difference vector and summing it with a predetermined target individual. Afterwards, we compare $\boldsymbol{C}_i(k)$ and $\boldsymbol{P}_i(k)$, and select the individual with higher $f_{\hat{t}}(g(\mathbf{x}; \boldsymbol{\theta}))$ as $\boldsymbol{P}_i(k+1)$, where $\boldsymbol{\theta} \in \{\boldsymbol{P}_i(k), \boldsymbol{C}_i(k)\}$. The evolution process stops when the label obtained by changing the image through $\boldsymbol{P}_{\gamma^*}(k)$ equals to $\hat{t}$, or the maximum number of iterations is reached. Through continuous evolution, it keeps good individuals, eliminates inferior ones, and guides the search to approach the optimal solution.

During this period, the population comparison mechanism obtains the predicted probability of the image by querying the model, and crossover depends on the random selection of individuals, neither of which needs the gradient information of the model. Therefore, the process is carried out under the complete black-box setting. After obtaining the optimal $\mathbf{r}^*$ via DE, we can generate the final adversarial examples. The modified attributes are called as adversarial attributes.

3.4. The explanation

In this section, we will give the reason why adversarial attributes can work to attack the DNN models. This explanation is based on the linear nature of DNN models.

In [7], Goodfellow et al. gave a linear explanation of adversarial examples. They began from a linear model. Suppose an adversarial example $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is the adversarial noise, and bounded by $\|\boldsymbol{\eta}\| \leqslant \epsilon$. For the linear model, the output is:

$$\boldsymbol{w}^\top \tilde{\mathbf{x}} = \boldsymbol{w}^\top \mathbf{x} + \boldsymbol{w}^\top \boldsymbol{\eta}. \tag{6}$$

If $\boldsymbol{w}$ has n dimensions and the average magnitude of an element of the weight vector is m, then the activation will grow by ϵmn [7]. When facing the high dimensional problems, an infinitesimal change of the input will add up to one massive change to the output. And thus the linear model will output the wrong result. They think that the DNN model consists of a variety of approximate linear modules, such as convolution, relu, etc. Therefore, the linear explanation can be used for the DNN model.

For our adversarial attributes, we can also use the linear nature of DNN to explain the underlying reason. We also begin from a liner model. The difference is that the adversarial example is generated via $\tilde{\mathbf{x}} = \theta_i \mathbf{x} + (1-\theta_i)\overline{\mathbf{x}}_i$. For the linear model, the output is $\boldsymbol{w}^\top \tilde{\mathbf{x}} = \theta_i \boldsymbol{w}^\top \mathbf{x} + (1-\theta_i)\boldsymbol{w}^\top \overline{\mathbf{x}}_i$. The formula is modified as follows:

$$\begin{aligned}\boldsymbol{w}^\top \tilde{\mathbf{x}} &= \theta_i \boldsymbol{w}^\top \mathbf{x} + (1-\theta_i)\boldsymbol{w}^\top \overline{\mathbf{x}}_i \\ &\propto \boldsymbol{w}^\top \mathbf{x} + \boldsymbol{w}^\top((1-\theta_i)/\theta_i)\overline{\mathbf{x}}_i = \boldsymbol{w}^\top \mathbf{x} + \boldsymbol{w}^\top \lambda_i \overline{\mathbf{x}}_i,\end{aligned} \tag{7}$$

where $\lambda_i = (1-\theta_i)/\theta_i$ is a small value. We can see that $\lambda_i \overline{\mathbf{x}}_i$ in Eq. (7) plays the same role with $\boldsymbol{\eta}$ in Eq. (6). For the given attribute, $\overline{\mathbf{x}}_i$ is a constant. We let the average magnitude of $\overline{\mathbf{x}}_i$ is D. Then the activation will grow by $\lambda_i nmD$. Similarly, when facing the high dimensional problems, an infinitesimal change of the input will add up to one large change to the output. This can explain that when we slightly modify any image attribute, the DNN model will have a big change in the output.

In terms of Eq. (6) and Eq. (7), the formulation of manipulating image attributes is a little similar to adding noise, however, there are some differences between them. Firstly, setting different camera parameters will make the image have different attribute characteristics. This is a common operation in our daily life. Therefore, the image obtained by adversarial attributes is natural. Secondly, the adversarial attribute is a uniform global perturbation for each pixel because the attribute change for each pixel is the same, while the random added noise is different for each pixel. Therefore it is a local operation. Overall, the adversarial attribute is a different attacking way compared with adversarial noises.

3.5. Implementation

In this section, we introduce some specific implementations in our method.

We select the `ImageEnhance` package in `PIL` to implement the $g(\mathbf{x};\boldsymbol{\theta})$, Eq. (4) is used in `ImageEnhance` to accomplish attribute adjustment. Four attributes are chosen to verify our method, they are brightness, contrast, chroma, and sharpness. Differential Evolution (DE) algorithm is implemented via the corresponding API in `SciPy` package. To solve Eq. (2) with L_∞ norm, we independently input the bound of each attribute into the DE algorithm. To solve Eq. (2) with L_2 norm, we firstly independently input the bound of each attribute, and then perform the rejecting sample, i.e., rejecting the attribute values that cannot meet the restriction $\|\boldsymbol{\theta}\|_2 \leqslant \epsilon$.

4. Experiments and results

4.1. Datasets

We use two public datasets to conduct experiments. The first one is CIFAR-10, and the second one is ImageNet.

CIFAR-10[2]: The image size is 32 × 32, and the number of classes is 10. There are a total of 50,000 images in the training set, and 10,000 images in the testing set. In our experiment, we only use 10,000 testing images to perform black-box attacks.

ImageNet[3]: There are 1,000 object categories. In the ILSVRC 2012, the validation data contains 50,000 images with labels. In our experiment, we choose a subset of the validation set. Specifically, we randomly choose 5 images for each category. In this way, we construct the testing set with 5,000 images to perform black-box attacks.

4.2. Target models

We choose four representative DNNs as our target models. They are VGG16 [23], AlexNet [12], Resnet50 [9] and Inception v3 [27]. For the ImageNet dataset, we directly use the corresponding APIs in `torchvision` package. For the CIFAR-10 dataset, we retrain the four networks to obtain the final parameters. In the experiments, these four models only give the predicted probability for a given image, and their structures and weights are not aware for the adversary.

4.3. Metrics

We use two metrics: Fooling rate and Query times.

Fooling rate: is defined as the percentage of adversarial examples that succeed in attacking the threat models in the black-box setting out of all the testing images.

Query times: is defined as the number of query times to the target model in order to successfully generate the adversarial examples.

[2] https://www.cs.toronto.edu/ kriz/cifar.html

[3] http://image-net.org/challenges/LSVRC/2012/.

4.4. Results and analysis

4.4.1. Performance comparisons

Firstly, we report the performance of our method on the CIFAR-10 and ImageNet against VGG16, AlexNet, Resnet50 and Inception v3, respectively. The results in the un-targeted attacking setting are listed in rows 2 to 7 of Table 1, where every three rows list the Fooling rate and Query times under the L_∞ and the L_2 constraint of the corresponding dataset. In the table, the numbers in the brackets denote the query times. From these results, we can see: (1) the proposed adversarial attributes can achieve at most 98.01% fooling rate in ImageNet dataset, and 97.79% fooling rate in CIFAR-10 dataset. This demonstrates the effectiveness of our method. (2) Compared with the L_∞ constraint, L_2 constraint usually shows a better fooling rate against the used three DNN models. This is because L_2 constraint has larger searching space than L_∞ constraint when they meet the same maximum perturbation. (3) Under L_∞ constraint, the query times are closely relevant to the image sizes. For example, the query times on CIFAR-10 are far below that on ImageNet. However, under the L_2 constraint, query times are not relevant to the image sizes. There are almost 200 queries for different networks and images.

The comparisons between our method and other black-box attacking methods are listed in Table 2 (the threat model is VGG16). Here we choose two classic methods. MI-FGSM [4] is based on the transferability to perform black-box attacks. It belongs to the transferability-based methods. One-pixel attack [26] generates adversarial examples by querying threat models. It needs the returned predicted probability, and therefore one-pixel attack belongs to the score-based methods. Table 2 shows our method achieves the best fooling rates compared with other attacks. It outperforms MI-FGSM with more than 10% improvement on CIFAR-10 and almost 5% improvement on ImageNet.

4.4.2. Ablation study

To better understand our method, we report the comparison results when manipulating the single attribute on CIFAR-10. The results are given in Table 3. We can see that if we only change one attribute, the fooling rates are very low. However, when we adjust four attributes simultaneously, the fooling rates show a quick rise. This contrast further verifies that our method is reasonable to adjust different attributes jointly.

4.4.3. The study of adversarial labels

In Fig. 3, we evaluate the number of image categories versus different fooling rates. The histograms are illustrated to measure this indicator on ImageNet. From the figure, we see that for the AlexNet, more than 90% of image categories are located in the bar of 100% fooling rate, which represents that AlexNet is vulnerable to adversarial attributes. While the histograms of VGG16, Resnet50 and Inception v3 are uniform, meaning they are relatively robust to adversarial attributes. Based on the

Table 1
Fooling rate and Query times on CIFAR-10 dataset and ImageNet dataset.

		VGG16	AlexNet	Resnet50	Inception_v3
un-targeted		CIFAR-10			
	L_∞	68.38%(74)	94.81%(64)	80.90%(70)	69.16%(82)
	L_2	70.64%(154)	71.67%(206)	83.63%(117)	97.79%(108)
		ImageNet			
	L_∞	43.97%(266)	87.11%(227)	42.70%(256)	40.69%(254)
	L_2	51.96%(178)	98.01%(136)	45.66%(197)	78.25%(157)
targeted		CIFAR-10			
	L_∞	18.00%(171)	36.57%(168)	23.49%(167)	20.54%(143)
	L_2	44.85%(122)	64.67%(107)	48.12%(111)	45.03%(113)

Table 2
Comparisons with the state-of-the-art black-box attacks.

Methods	One-pixel	MI-FGSM	Ours-L_2
CIFAR-10	31.40%	60.24%	**70.64%**
ImageNet	16.04%	47.18%	**51.96%**

Table 3
Comparison results versus single attribute.

Methods	VGG16	AlexNet	Resnet50	Inception_v3
Chroma	10.00%	16.86%	6.98%	6.16%
Brightness	27.14%	50.74%	24.90%	23.15%
Sharpness	5.04%	17.80%	7.05%	5.93%
Contrast	16.26%	38.24%	16.55%	15.56%
Together	**68.38%**	**94.81%**	**80.90%**	**69.16%**

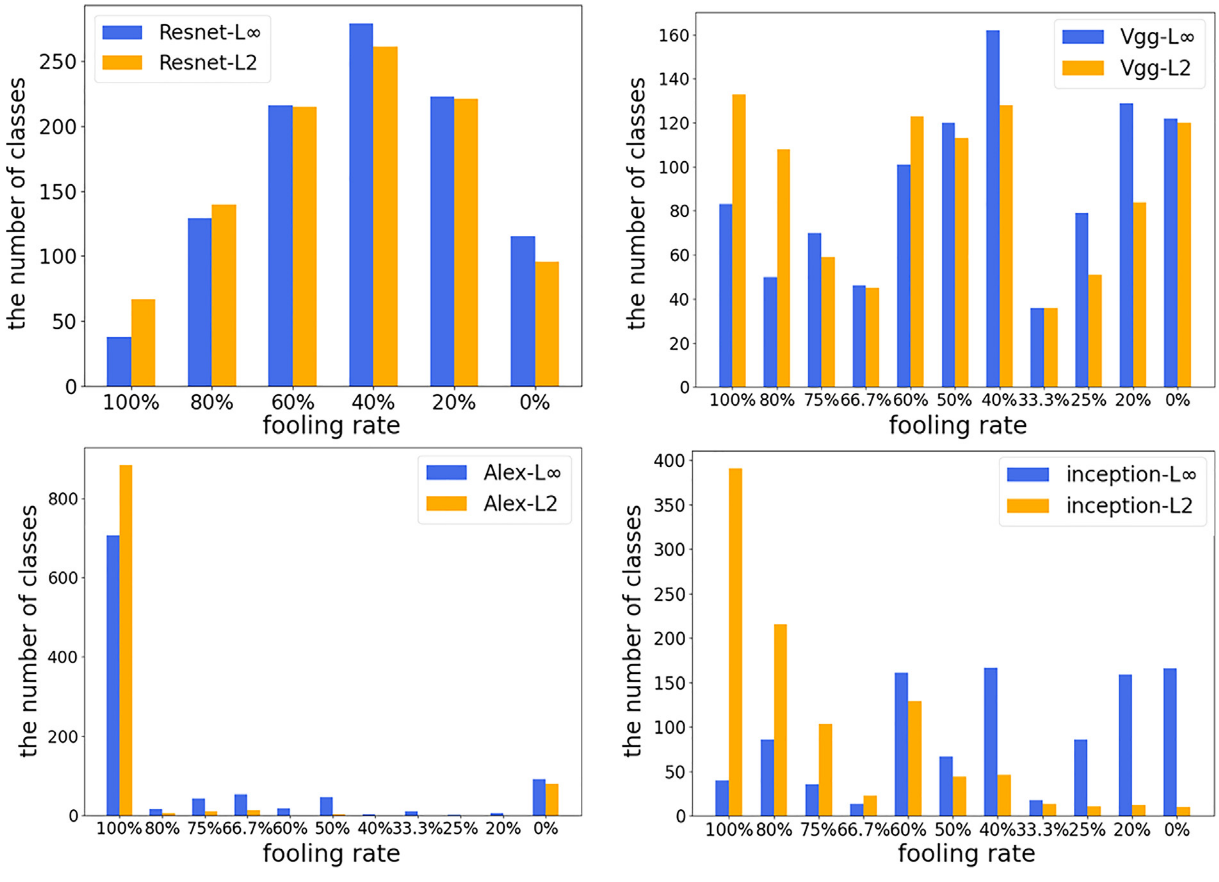

Fig. 3. The statistics of image categories versus different fooling rates. The results are achieved on ImageNet dataset against Resnet50, VGG16, AlexNet and Inception v3 under two constraints.

statistics, we select some original categories with 100% fooling rate to see the relationship with their corresponding adversarial labels. In Fig. 4, eight such examples are given. For each graph, the center node denotes the original label, and the leaf nodes denote the adversarial labels that have been perturbed into.

From the figure, we find one phenomenon, i.e., the adversarial labels are similar to the original labels. For example, in the second sub-figure of top row, the center node is `night snake`, the leaf nodes are `Indian cobra`, `horned viper`, etc. These

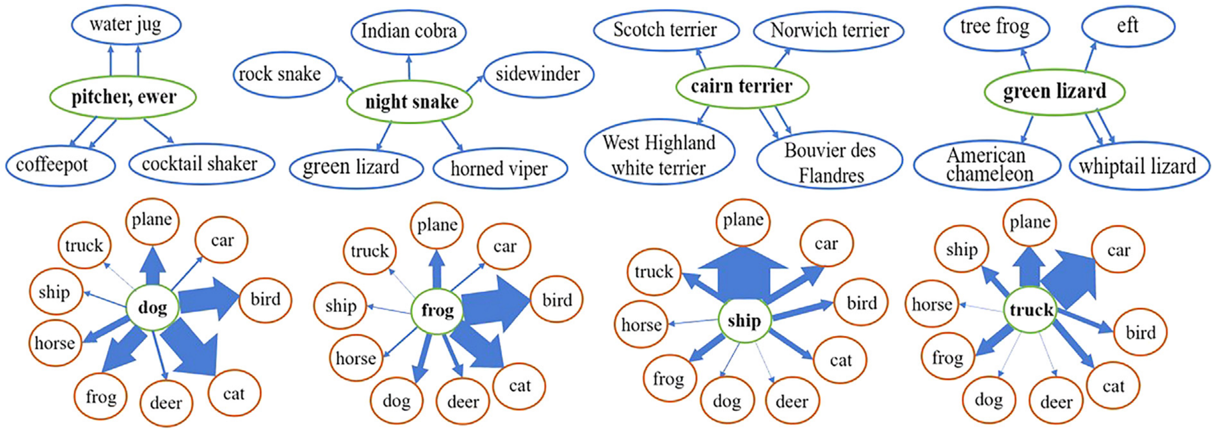

Fig. 4. The mis-classified results of un-targeted attack. For each graph, the center node denotes the original label, and the leaf nodes denote the adversarial labels that images are perturbed into. The top row is the results of ImageNet. Because each category has five images (see Datasets section), every center node has five edges. We merge the coincident leaf nodes, and retain the edges. The bottom row is the results of CIFAR-10. The arrow's width represents the number of images that are fooled from the center label to leaf labels.

adversarial labels belong to the same snake class, but different fine-grained subclass. In the first sub-figure of bottom row, the center node is `dog` and the most adversarial labels are `cat`. They are both within the animal class. This is different from the results of adversarial noises. In the traditional attacking methods, the un-targeted attacking labels have no obvious fixed relation with the original labels. But in our method, the adversarial labels show different properties. This is because, in the un-targeted attack, we aim to minimize the confidence of ground-truth label. In this way, the label with the second highest confidence becomes the adversarial label. This label is usually very similar to the ground-truth label.

4.4.4. Robustness of adversarial attributes

The robustness of adversarial attributes is also tested. We select two representative defense algorithms: JPEG compression [5] and Spatial smoothing [34]. The adversarial examples processed by these defense methods are then inputted into DNN models to check the drop of fooling rates. We here compare with FGSM versus the robustness. The final results are given in Fig. 5. For each sub-figure, we list the fooling rate drop after defense. From the figure, we see that adversarial examples of FGSM cannot resist JPEG compression and Spatial smoothing. For Resent50, the drop is 31.01% at most. However, our method shows better robustness for these defense methods. Among them, L_2 constraint is more robust than L_∞ constraint. Because for all the four models, the fooling rate drops of L_2 are almost zero. Note that FGSM is a white-box attack, therefore, the fooling rate of FGSM is higher than our black-box attacking method. Because we focus on the robustness, the fooling rate drop is the key indicator. From the view of the fooling rate drop, the robustness of our method is pretty good.

We also use adversarial training as a defense method to test the robustness of the proposed attack method. Table 4 lists the comparison results of Fooling rates and Query times on CIFAR-10 dataset under L_2 constraint before and after adversarial

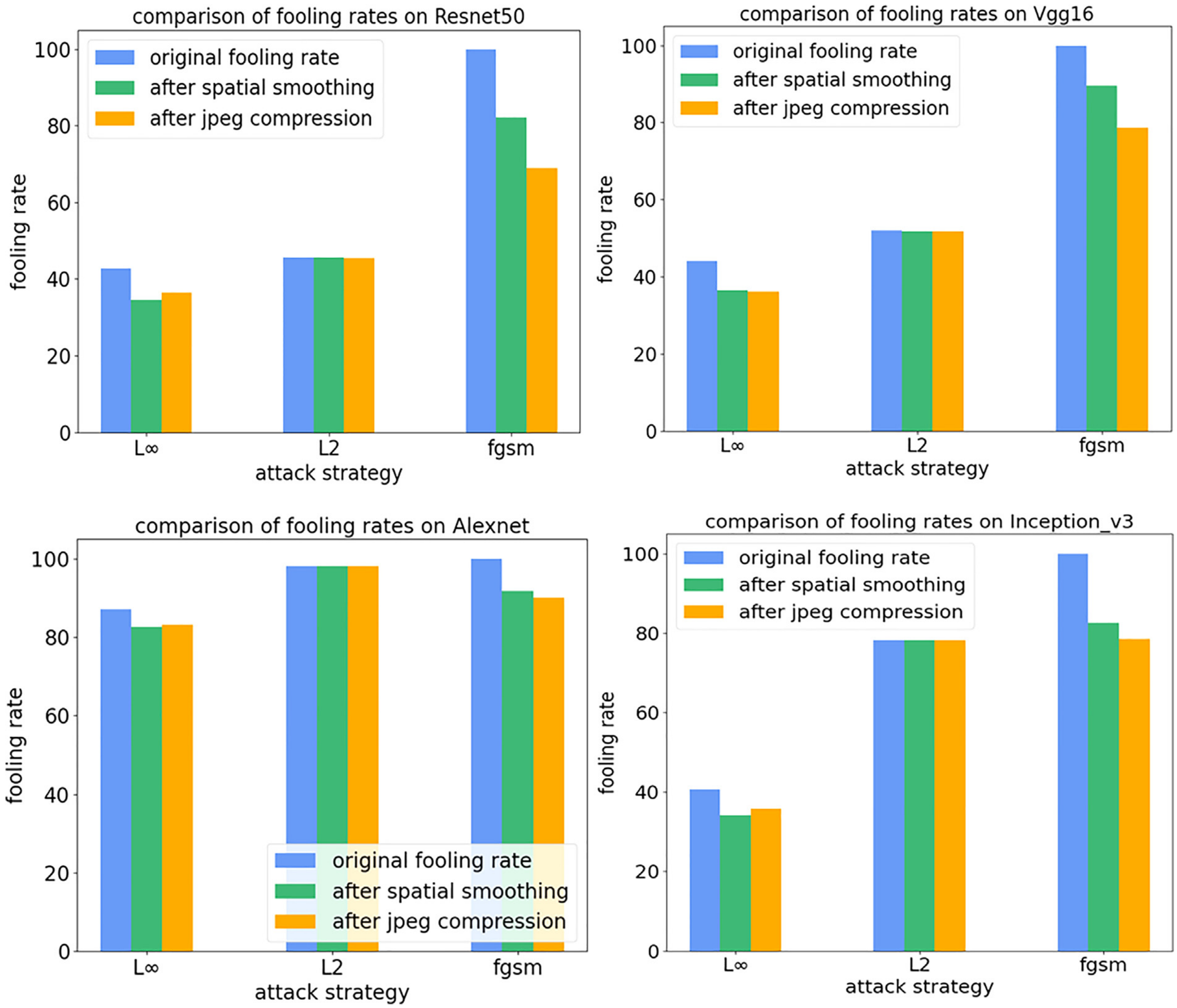

Fig. 5. The fooling rate comparison before and after defense methods. We here choose FGSM as a baseline. Note that FGSM is a white-box attack, therefore, the fooling rate of FGSM is higher than our black-box attacking method. Because we focus on the robustness, the fooling rate drop is the key indicator.

Table 4
Comparison results before and after adversarial training.

	VGG16	AlexNet	Resnet50
adv_training	78.23%(179)	72.70%(249)	67.77%(234)
original	70.64%(154)	71.67%(206)	83.63%(117)

training. It can be seen from the table that the Fooling rate only slightly decreases in Resnet50 after adversarial training, and there is no significant change in VGG16 and Alexnet. Although the Query times needed increase slightly, the range is not large. Therefore, our adversarial attributes can effectively resist the defense of adversarial training.

4.4.5. Qualitative results

We give some qualitative results in Fig. 6, where the first two rows are under L_2 constraint and the last two rows are under L_∞ constraint. In each constraint, the top row denotes the clean images, and the bottom row denotes the adversarial examples with adversarial attributes. For each pair, the ground-truth labels (black texts) and the adversarial labels (red texts) are given. From these images, we see the adversarial examples look very normal. To some extent, they are more beautiful and aesthetic than the clean images. For example, the fourth adversarial image of L_2 constraint is more artistic than the original image. Therefore, they will not arouse people's suspicion. These results demonstrate adversarial attributes are indeed an alternative way to generate adversarial examples besides the widely used adversarial noises.

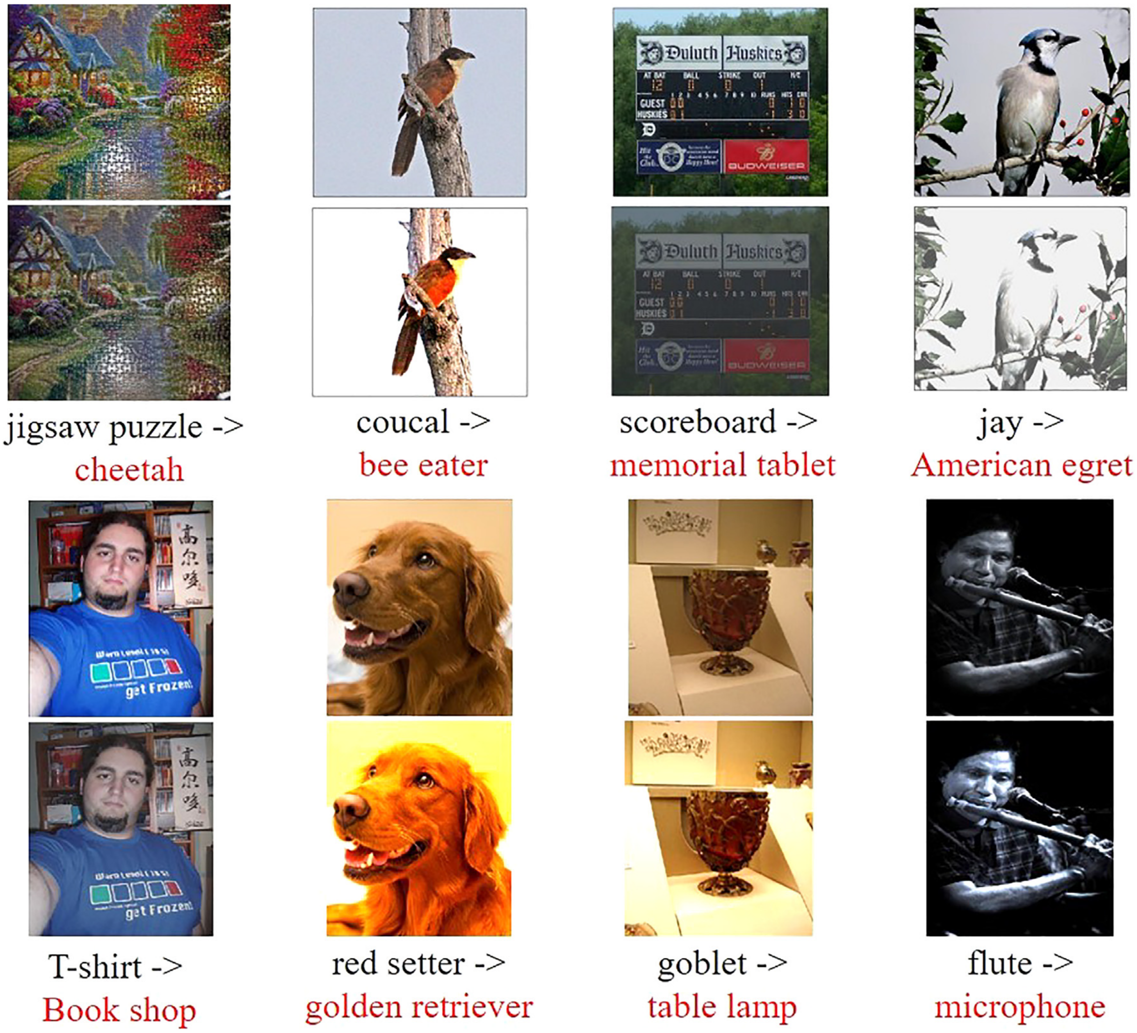

Fig. 6. Qualitative results are generated under L_2 and L_∞ constraints in un-targeted attack. The first two rows are under L_2 constraint and the last two rows are under L_∞ constraint. In each constraint, the top row denotes the clean images, and the bottom row denotes the adversarial examples with adversarial attributes. For each pair, the ground-truth labels (black texts) and the adversarial labels (red texts) are given. Please see the images in the color mode. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

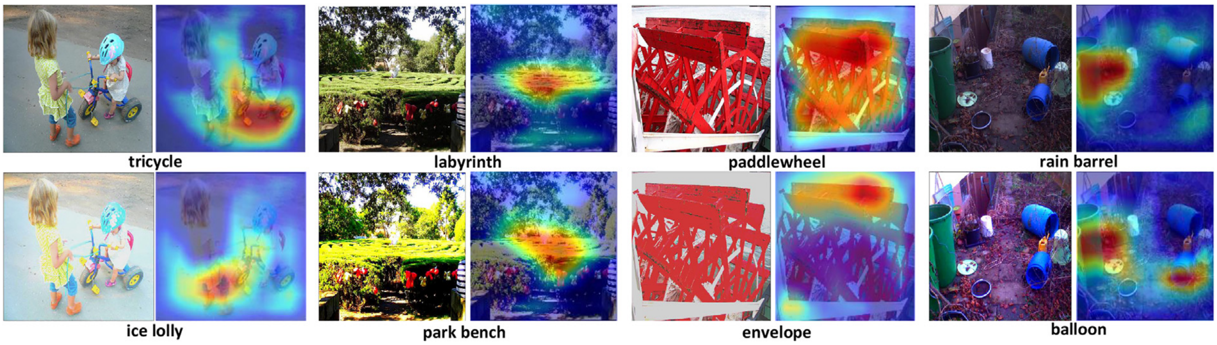

Fig. 7. The heatmaps of clean images and their corresponding adversarial examples. For each pair, the top row is the clean image and its ground-truth label; the bottom row is the adversarial examples and its targeted adversarial label. The heatmaps are generated using Grad-Cam [19]. Please see the images in. the color mode. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.4.6. Targeted attacks

Now we perform the targeted attacks using our method, i.e., solving the Eq. (2). For simplicity, we here give the results on CIFAR-10 dataset under L_2 and L_∞ constraint, respectively. To perform targeted attacks, we randomly choose another label ($\neq$ ground-truth label) in CIFAR-10 as the targeted labels. The results are listed in the last three lines of Table 1, where we can see that (1) the fooling rates of targeted attacks are below compared with un-targeted attack shown in lines 2 to 7 of Table 1. This is expected because targeted attack is more difficult than un-targeted attack. (2) L_2 norm shows better fooling rate than L_∞ norm on all the four DNN models. The best fooling rate is achieved on attacking AlexNet (64.67%). It is consistent with the un-targeted attacks, i.e., AlexNet is more sensitive to the adversarial attributes than VGG16, Resnet50 and Inception v3. (3) the query times of targeted attacks are almost the same with that of un-targeted attacks. This is beyond our expectation. The targeted attacks usually need more query times to meet the goals. We think this may benefit from the great power of differential evolution algorithm. It can quickly find the optimal solution of Eq. (2).

To understand the mechanism of adversarial attributes, we generate adversarial examples under the targeted attacks and then illustrate their heatmaps using [19]. Four examples are given in Fig. 7. From the figure, we see that the attentions of adversarial examples are all changed compared with the attentions of clean images. For example, in the first pair, the area with the largest response of the clean image lies in the tricycle. However, the area with the largest response of the adversarial image lies in the girl's leg. These heatmaps show that adversarial attributes fool the DNN models by modifying the areas of the largest response, which is consistent with the adversarial noises.

5. Conclusion

In this paper, we proposed the adversarial attributes, an alternative way to generate adversarial examples. Our method manipulated the image attributes to perform the black-box attacks. To this end, we utilized Differential Evolution to solve the optimal changing value for each attribute. To make the adversarial examples realistic, we bounded the attribute changes using L_∞ and L_2 constraint. Experiments showed that L_2 constraint usually had better fooling rates and better robustness than L_∞ constraint. Another feature of adversarial attributes was that the generated adversarial labels were usually relevant with the ground-truth labels. Qualitative results told us the adversarial examples were normal, and to some extent, they were more beautiful and artistic than the clean images. Therefore, they would not arouse people's suspicion. In the future, how to better select the changing value for each attribute is an interesting problem, and it is a good choice to utilize the reinforcement learning.

CRediT authorship contribution statement

Xingxing Wei: Methodology, Writing - original draft. **Ying Guo:** Software, Validation, Data curation. **Bo Li:** Conceptualization, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Supported by National Natural Science Foundation of China (No.62076018, 61806109), National Key R&D Program of China (Grant No.2020AAA0104002), and CCF-Tencent Open Research Fund.

References

[1] B.S. Vivek, R. Venkatesh Babu. Single-step adversarial training with dropout scheduling, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 947–956, June 2020..

[2] Nicholas Carlini, David Wagner. Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), pages 39–57. IEEE, 2017..

[3] Lixi Deng, Jingjing Chen, Qianru Sun, Xiangnan He, Sheng Tang, Zhaoyan Ming, Yongdong Zhang, Tat Seng Chua. Mixed-dish recognition with contextual relation networks, in: Proceedings of the 27th ACM International Conference on Multimedia, pages 112–120. ACM, 2019..

[4] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, Jianguo Li. Boosting adversarial attacks with momentum, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9185–9193, 2018..

[5] Gintare Karolina Dziugaite, Zoubin Ghahramani, Daniel M Roy. A study of the effect of jpg compression on adversarial images. preprint arXiv:1608.00853, 2016..

[6] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, Dawn Song. Robust physical-world attacks on deep learning visual classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1625–1634, 2018..

[7] Ian J Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014..

[8] P. Haeberli, D. Voorhies, Image processing by linear interpolation and extrapolation, IRIS Universe Magazine 28 (1994) 8–9.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[10] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, Hassan Foroosh, Comdefend: An efficient image compression model to defend adversarial examples, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6084–6092.

[11] Stepan Komkov, Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. arXiv preprint arXiv:1908.08705, 2019..

[12] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pages 1097–1105, 2012..

[13] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy Chowdhury, Ananthram Swami. Adversarial perturbations against real-time video classification systems. arXiv preprint arXiv:1807.00458, 2018..

[14] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Hu. Xiaolin, Jun Zhu, Defense against adversarial attacks using high-level representation guided denoiser, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1778–1787.

[15] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard. Universal adversarial perturbations, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1765–1773, 2017..

[16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.

[17] Tianyu Pang, Chao Du, Jun Zhu. Robust deep learning via reverse cross-entropy training and thresholding test. arXiv preprint arXiv:1706.00633, 3, 2017..

[18] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, Bo Li. Semanticadv: Generating adversarial examples via attribute-conditional image editing. CoRR, abs/1906.07927, 2019..

[19] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[20] Yucheng Shi, Yahong Han, Qi Tian. Polishing decision-based adversarial noise with a customized sampling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1027–1035, June 2020..

[21] Yucheng Shi, Yahong Han, Quanxin Zhang, Xiaohui Kuang, Adaptive iterative attack towards explainable adversarial robustness, Pattern Recogn. 105 (2020) 107309.

[22] Yucheng Shi, Siyu Wang, Yahong Han. Curls & whey: Boosting black-box adversarial attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6512–6520, June 2019..

[23] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014..

[24] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. arXiv preprint arXiv:1710.10766, 2017..

[25] Rainer Storn, Kenneth Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[26] Su. Jiawei, Danilo Vasconcellos Vargas, Kouichi Sakurai, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. 23 (5) (2019) 828–841.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013..

[29] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204, 2017..

[30] Xingxing Wei, Siyuan Liang, Xiaochun Cao, Jun Zhu. Transferable adversarial attacks for image and video object detection. arXiv preprint arXiv:1811.12641, 2018..

[31] Xingxing Wei, Jun Zhu, Sitong Feng, Hang Su. Video-to-video translation with global temporal consistency, in: 2018 ACM Multimedia Conference on Multimedia Conference, pages 18–25. ACM, 2018..

[32] Xingxing Wei, Jun Zhu, Sh.a. Yuan, Su. Hang, Sparse adversarial perturbations for videos, Proc. AAAI Conf. Artif. Intell. 33 (2019) 8973–8980.

[33] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, Dawn Song. Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610, 2018..

[34] Weilin Xu, David Evans, Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155, 2017..

[35] Wanting Yu, Hongyi Yu, Lingyun Jiang, Mengli Zhang, Kai Qiao, Linyuan Wang, Bin Yan. Had-gan: A human-perception auxiliary defense gan model to defend adversarial examples. arXiv preprint arXiv:1909.07558, 2019..

[36] Rowan Zellers, Yonatan Bisk, Ali Farhadi, Yejin Choi, From recognition to cognition: Visual commonsense reasoning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6720–6731.

[37] Zhengyu Zhao, Zhuoran Liu, Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1039–1048, June 2020..

[38] Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, Atul Prakash. Efficient adversarial training with transferable adversarial examples, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1181–1190, June 2020..