Wuhan University
Journal of Natural Sciences

2023, Vol.28 No.1, 035-044

Article ID 1007-1202(2023)01-0035-10
DOI https://doi.org/10.1051/wujns/2023281035

Adversarial Example Generation Method Based on Sensitive Features

WEN Zerui[1], SHEN Zhidong[1,3†], SUN Hui[2], QI Baiwen[2]

1. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430079, Hubei, China;

2. Zhongnan Hospital, Wuhan University, Wuhan 430072, Hubei, China;

3. Engineering Research Center of Cyberspace, Yunnan University, Kunming 650504, Yunnan, China

Abstract: As deep learning models have made remarkable strides in numerous fields, a variety of adversarial attack methods have emerged to interfere with deep learning models. Adversarial examples apply a minute perturbation to the original image, which is inconceivable to the human but produces a massive error in the deep learning model. Existing attack methods have achieved good results when the network structure is known. However, in the case of unknown network structures, the effectiveness of the attacks still needs to be improved. Therefore, transfer-based attacks are now very popular because of their convenience and practicality, allowing adversarial samples generated on known models to be used in attacks on unknown models. In this paper, we extract sensitive features by Grad-CAM and propose two single-step attacks methods and a multi-step attack method to corrupt sensitive features. In two single-step attacks, one corrupts the features extracted from a single model and the other corrupts the features extracted from multiple models. In multi-step attack, our method improves the existing attack method, thus enhancing the adversarial sample transferability to achieve better results on unknown models. Our method is also validated on CIFAR-10 and MINST, and achieves a 1%-3% improvement in transferability.

Key words: deep learning model; adversarial example; transferability; sensitive characteristics; AI security

CLC number: TP 391.4

0 Introduction

Neural networks are now widely used in a variety of fields, including automatic driving[1,2], medical treatment[3,4], biology[5,6], and finance[7], where they have produced remarkable results. However, we must consider the security issue posed by deep neural networks. Similar to the problem in the traditional field of security, we must carefully examine user input. In the SQL(Structured Query Language) injection attack and XSS(Cross Site Scripting) attack, users can inject malicious code into the entry field to dump database and server content. The situation is identical with regard to deep neural networks. Confronted with abnormal inputs, we must pay close attention to the functionality of deep neural networks. There have been early occurrences of attacks of

Received date: 2022-09-06

Foundation item: Supported by the Key R&D Projects in Hubei Province (2022BAA041 and 2021BCA124) and the Open Foundation of Engineering Research Center of Cyberspace (KJAQ202112002)

Biography: WEN Zerui, male, Master candidate, research direction: AI security and adversarial attack. E-mail: zeruiwen2018@163.com

† To whom correspondence should be addressed. E-mail: shenzd@whu.edu.cn

this type. Some attackers were able to evade system detection, for instance, in malicious email detection and intrusion detection systems employing deep learning models, due to the characteristics of the models. The attack on contemporary deep neural networks will inevitably result in a breach of privacy, identity theft, and numerous other grave issues.

The development of attack enhances the defense. If we can discover more efficient attack methods, it will undoubtedly benefit defensive abilities. Adversarial examples apply a minute perturbation to the original image, which is inconceivable to the human eye but induces massive errors in the deep learning models. There are two types of adversarial example methods: black-box attacks and white-box attacks. White-box attacks such as FGSM (fast gradient sign method)[8], BIM(basic iterative method)[9] and C&W(Carlini &Wagner)[10] can be very effective when the model is known. In black-box attack, there are two prevalent methods: query and migration. The former must repeatedly access the deep neural networks. In the real world, many APIs(Application Programming Interface) of online models are not free, making them costly and easily detected. Transferability is the superior option. Szegedy *et al*[8] have discovered that adversarial examples are transferable. Additionally, adversarial examples produced by one model pose a threat to other models. By showcasing transferability, we are able to conduct black-box attacks without visiting the model, making our attacks more covert. In transfer-based attack, data transformation is a common method in migratory attacks, just like DI^2-FGSM(Diversity Input)[11].

In this work, our approach seeks to explore whether better transferability can be achieved by focusing on destroying sensitive features of the image, starting from the image itself. In the real world, it is less likely that we are aware of the models our target employs. Therefore, black box attacks are more significant in reality. Our method focuses on enhancing transferability in order to achieve superior results on unexplored deep neural networks. We were motivated by the human attention mechanism. Attention mechanism is that humans will selectively focus on a portion of all information while ignoring the rest of the visible information. For example, people will pay more attention on cat's face when they try to distinguish between cats and dogs. As shown in Fig. 1, the same mechanism applies to deep neural networks for classified tasks. All models focus on the face of the cat, when all they try to classify this image. If the most sensitive features are attacked, it is more likely that all models will provide an incorrect answer.

The most closely related work is the ATA(Attention-guided Transfer Attack) method[12]. It features attention transferability. However, it is inadequate for the conditions with limited local algorithmic power since the immediacy is highly important. Meanwhile, it fails to recognize that the sensitivity characteristics generated by various models are also distinctive. The differences may hinder transferability. As shown in Fig. 1, although all the models focus on the face of the cat, there are distinctions. ResNet-50 is more focused on the body of the cat. Shufflennet v2 is more attentive to other details of the cat.

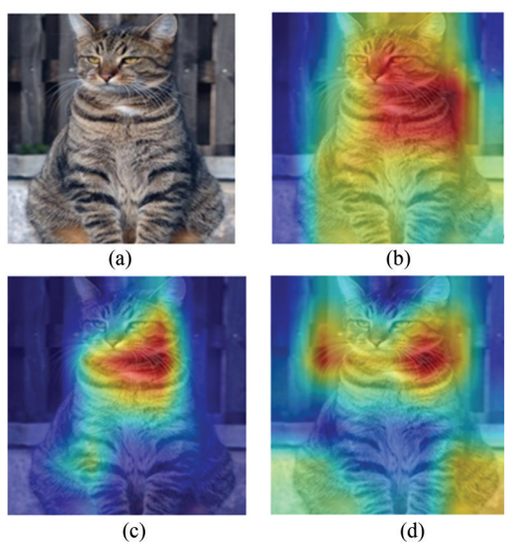

Fig. 1 Grad-Cam feature maps generated under various networks

(a) is an original image belonging to the 281st ImageNet class; (b) is generated through ResNet-50; (c) is generated through Shufflennet v2; (d) is generated through VGG19

Due to the variability of attention areas, firstly, we propose a single-step attack based on vulnerable characteristics from single network. In existing work, few people consider the effects of single-step attacks. However, single-step attacks are fast and have a low number of accesses to the model. As for the method of extracting sensitive features, we refer to the ATA method using Grad-CAM(Class Activation Maps)[13], which extracts the areas of interest of the network for a given image. Then, we propose a one-step attack based on multiple sensitive characteristics from different networks. Finally, the idea of multi-feature fusion is applied to multi-step

attacks, where the current ATA approach only considers sensitive features from one model. In conclusion, we enhance the ATA method by incorporating sensitive characteristics from other classification models. We can outperform the original ATA method by 1%-3% on the CIFAR-10 and MINST dataset.

1 Related Work

The adversarial example attacks can be divided into black-box and white-box attacks based on whether the attacker knows the model structure or not. For black-box attacks, transfer-based attacks are more commonly used because they do not require access to the target model. Next, we introduce representative methods for each of these types of attacks and the most advanced methods that related to our work.

1.1 White-Box Attack

Since we understand the architecture of deep neural networks, it is obvious that white box attacks are more efficient and dangerous. FGSM is a well-established white-box attack method[8]. In FGSM, the loss function is maximized by taking steps in the opposite direction of the gradient of the loss function. BIM is the iterative version of FGSM[9], whereas super parameters must be manually set in FGSM. In addition, when loss function comes to nonlinear functions, we do not know if the loss function will increase or decrease. BIM solves the aforementioned issues. It divides the FGSM attack into multiple turns in order to achieve better outcomes.

1.2 Black-Box Attack

Query-based attacks use input to learn the gradient information of the model[14], while in reality, the effect of Query-based attacks is satisfying[15].

1.3 Transfer-Based Attack

DI^2-FGSM[11] is transfer-based attack using data transformation. It applies a combinatorial transformation to the data with probability p before generating an adversarial sample. Wu *et al* [16] further used adversarial transformations as new transformations to enhance transferability. Like mentioned before, ATA method[12] features attention transferability. Our work also enhances ATA with sensitive features from different models.

2 Background

In this section, we destruct sensitive features to improve transferability. First, we discuss how we acquire sensitivity features. In addition, we present our baseline FGSM and BIM. Finally, we discuss the ATA method and several aspects that ATA fails to notice.

2.1 Grad-CAM

Grad-CAM[13] can be described as formula (1) and (2). As revealed in formula (1), for specified class c, we find the partial derivative of class c for each feature map k. Then we apply global average pooling to get α_k^c which represents the importance of feature map k in judging class c. In order to visualize the result, we need to apply ReLU activation function to liner combination. ReLU can filter the pixel that has negative impact on judging class c.

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \tag{1}$$

$$L_{\text{Grad-CAM}}^c = \text{ReLU}\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}} \tag{2}$$

2.2 FGSM

FGSM[8] is a classic white-box attack method. It is based on the gradient. Conventionally, we move toward the gradient in order to decrease the loss function. In FGSM method, we move toward the gradient in step ε so that the loss function will increase. The greater the ε is, the easier our adversarial examples are to be detected. The method can be formulated as follows:

$$x^{\text{adv}} = x + \varepsilon\ \text{sign}\left(\nabla_x J\left(\theta, x, y\right)\right) \tag{3}$$

where x^{adv} represents adversarial examples, x is original data. ε is perturbation budget, θ stands for network parameter and y is label of the original data. sign() is symbolic functions to get the sign of gradient.

2.3 BIM

In FGSM, we need to set ε mutually which may cause undesirable effect in non-linear condition. Then BIM[9] is proposed to solve this problem. In BIM, we divide an FGSM step into multiple steps. In each step, we perform similar tasks just like we did in FGSM. We attempt to increase the loss function based on previous iteration. The method can be formulated as follows:

$$X_0^{\text{adv}} = X, X_{N+1}^{\text{adv}} = \text{Clip}_{X,\epsilon}\left\{X_N^{\text{adv}} + \alpha\text{sign}\left(\nabla_x J\left(X_N^{\text{adv}}, y_{\text{true}}\right)\right)\right\} \tag{4}$$

where y_{true} is the true label of origin data, X is original data, α is perturbation budget which satisfies $0<\alpha<\epsilon$, ϵ is also a perturbation budget.

2.4 ATA

ATA[12] is a black-box method with good transferability[19]. It is superior to C&W[10], JSMA(Jacobian-based Saliency Map Attack)[17], BIM[9], and TAP(Transferable Adversarial Perturbations)[18] in terms of transferability. It modifies the loss function in consideration of the sensitive features. We use L2 distance to denote the distance between two images. The greater the L2 distance is, the more differences there are between the two images. The following is the new loss function:

$$J_{\text{Grad}-\text{CAM}}\left(x^n, x^{\text{ori}}\right) = l\left(f\left(x^n\right), t\right) + \lambda \cdot \left(L^{y^{\text{ori}}}_{\text{Grad}-\text{CAM}}\left(x^n\right) - L^{y^{\text{ori}}}_{\text{Grad}-\text{CAM}}\left(x^{\text{ori}}\right)\right)_2^2 \quad (5)$$

Then we can solve the following optimization issues:

$$\text{maximize}\, J_{\text{Grad}-\text{CAM}}\left(x^n, x^{\text{ori}}\right) \text{ subject to } \left(x^p - x\right)_2 \leqslant \varepsilon \quad (6)$$

As a black-box attack, it means the adversarial examples generated by ATA has outstanding transferability. ATA generates adversarial examples on ResNet V2, Inception V3 , Inception V4 and Network with both Res and Inception blocks, respectively. Then adversarial examples generated by the local model are used to attack other models, for example, using adversarial examples generated on ResNet V2 to attack Inception V3 and Inception V4, which can also gain fine results. ATA outperforms other excellent methods like C&W[10] and TAP[18].

However, the ATA method does not account for sensitive features produced by other methods. In addition, ATA requires a massive number of calculations. We can apply this method to more realistic scenarios if we can devise a single step method to generate adversarial examples quickly with limited local computing power.

3 Generating Adversarial Examples Using Sensitive Features

We will introduce three new methods: one-step attack based on sensitive features from single model, one-step attack based on sensitive features fusion and multi-step attack based on sensitive features fusion, step by step. One-step attacks are more suitable under conditions where speed is crucial. Due to the fact that there are times when we require adversarial examples as quickly as possible, such as during military operations, one-step attack is also needed in real-world situation. Under the condition that we have sufficient time to plan our attacks, a multi-step attack is preferable. In general, classifier f takes input picture x, then a probability vector will be produced showcasing the most likely class of x. In adversarial example, we try to add perturbation on original picture x in order to deceive the classifier. When the new image x^p is the input of classifier f, the classifier will make wrong decision. Meanwhile, the distance between x and x^p should not be too far or our x^p is easy to be detected. Formula (7) needs to be satisfied:

$$\min \left\| x - x^p \right\|_\infty \quad \text{and} f\left(x + p\right) = c^{\text{new}} \neq c^{\text{ori}} \quad (7)$$

where c^{new} is the new prediction of the adversarial example, c^{ori} is the prediction of the origin clean image.

So the key is to solve the following optimization problem:

$$f\left(x + p\right) = c^{\text{new}} \neq c^{\text{ori}} \text{ subject to } \min \left\| x - x^p \right\|_\infty < \epsilon \quad (8)$$

3.1 One-Step Attack

Firstly, we introduce one-step attack based on sensitive features of single model. In one-step attack, we send original images into the network to acquire the sensitive features. Then we only reserve sensitive section and send x' into network again. We produce adversarial examples according to new parameters of the network. Algorithm 1 describes the process of generating x'. The mask in one-step attack is as the same size of the input data. In sensitive region, mask contains 1, otherwise, contains 0. Then we calculate the Hadamard product of the input images and the mask. We can reserve the sensitive features.

Algorithm 1 Generate x' sensitive features of single model

Input: x: original image, θ: parameters of the network.
Output: x'
mask = grad_CAM(θ)
# Reserving sensitive section
$x' = x$ *mask
return x'

The aforementioned method only takes into account sensitive features generated by one model. Then, we propose a one-step attack method that combines multiple sensitive features generated by various models. Algorithm 2 describes the process of generating x' using multiple sensitive features.

The α, β, γ are the probability that selects the pixels of three different section. The probability equal to 1 denotes that we choose every pixel in the area. The probability equal to 0 denotes that no pixels should be se-

Algorithm 2 Generate x' using multiple sensitive features generated by various models

Input: x :original image, θ_1: parameters of the network1, θ_2: parameters of the network2, θ_3: parameters of the network3
Output: x'
$\text{mask}_1 = \text{grad\_CAM}(\theta_1)$
$\text{mask}_2 = \text{grad\_CAM}(\theta_2)$
$\text{mask}_3 = \text{grad\_CAM}(\theta_3)$
Reserving sensitive section
$x' = x * (\alpha * (\text{mask}_1 \cap \text{mask}_2 \cap \text{mask}_3) \cup \beta * (\text{mask}_{2,3} \cap \text{mask}_{1,3} \cap \text{mask}_{1,2}) \cup \gamma * (\text{mask}_{1\text{only}} \cap \text{mask}_{2\text{only}} \cap \text{mask}_{3\text{only}})$
return x'

lected in the section. Clearly, if α,β,γ are too large, the x' will be too similar to the original image x, which can be formulated as follows:

$$\lim_{\alpha,\beta,\gamma \to 1} x' = x \tag{9}$$

After generating the x', we feed it into the neural network again. The attack is based on new neural network parameters. The concept of attack is comparable to that of FGSM, which moves in the opposite direction of the gradient. Formula (10) is the method of attack.

$$x^p = x + \varepsilon \cdot \text{sign}\left(\nabla_{x'} J\left(x', y\right)\right) \tag{10}$$

3.2 Multi-Step Attack

In the multi-step attack, we modify the loss function by factoring in the sensitive features generated by different models. The objective of the new loss function is to destroy sensitive features based on the focus of the three models.

$$\begin{aligned} J_{\text{Grad-CAM}}\left(x^n, x^{\text{ori}}\right) &= l\left(f\left(x^n\right), t\right) \\ &+ \alpha \cdot \left\| L_{\theta_1}^{y^{\text{ori}}}\left(x^n\right) - L_{\theta_1}^{y^{\text{ori}}}\left(x^{\text{ori}}\right) \right\|_2^2 \\ &+ \beta \cdot \left\| L_{\theta_2}^{y^{\text{ori}}}\left(x^n\right) - L_{\theta_2}^{y^{\text{ori}}}\left(x^{\text{ori}}\right) \right\|_2^2 \\ &+ \gamma \cdot \left\| L_{\theta_3}^{y^{\text{ori}}}\left(x^n\right) - L_{\theta_3}^{y^{\text{ori}}}\left(x^{\text{ori}}\right) \right\|_2^2 \end{aligned} \tag{11}$$

The α,β,γ enable us to assign weight so that we can decide which sensitive feature should be pay more attention to.

BIM can be utilized to address the optimization issue. We increase the loss function with each iteration in order to destroy the sensitive features and deceive the classifier. BIM can be regarded as a multi-step version of FGSM. Algorithm 3 is our multi-step method. Figure 2 shows the flowchart of this method.

Algorithm 3 Multi-step attack

Input: x: original image, θ_1: parameters of the network1, θ_2: parameters of the network2, θ_3: parameters of the network3, ϵ: budget
Output: x^p
$\epsilon' = \dfrac{\epsilon}{\text{iter}}$
$x^0 = x$
for $k = 0$ **to** $\text{iter} - 1$ **do**
$x^{k+1} = \text{clip}_{x,\epsilon}\left\{x^k + \epsilon' \text{sign}\left(\dfrac{\partial J_{\text{Grad-CAM}}\left(x^k, x, \theta_1, \theta_2, \theta_3\right)}{\partial x}\right)\right\}$
end for
return $x^p = x^{\text{iter}}$

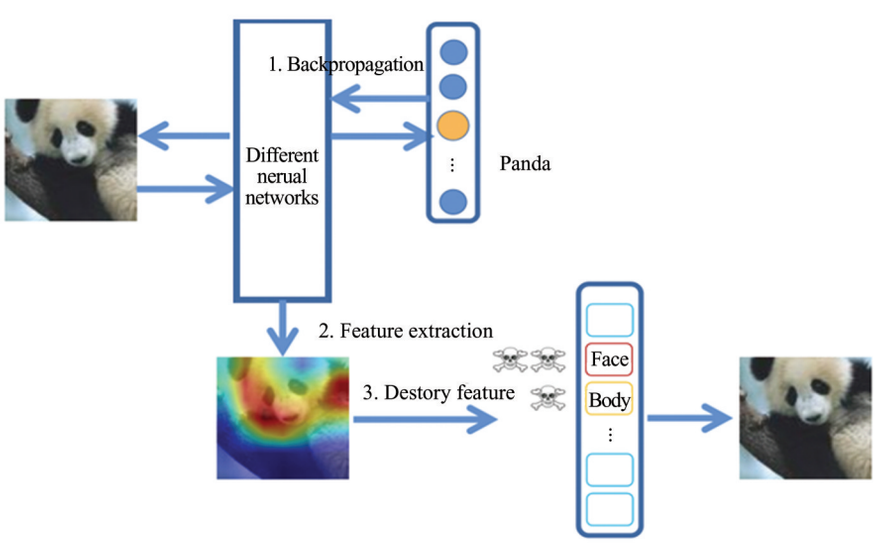

Fig. 2 The process of the multi-attack based on feature fusion

4 Experiment

Experimentally, we demonstrate that adversarial examples generated by our method are more transferable. Meanwhile, the differences between adversarial examples and original image of our method remain constant with other methods.

4.1 Setup

Dataset: We use CIFAR-10[19] and MINST[20] as our Dataset. CIFAR-10 includes 60 000 images with a resolution of 32×32×3. There are ten categories. These were used as the training set, which consisted of 50 000 items, and the test set consisted of 10 000 items. MINST also included 60 000 images but with a resolution of 28×28×1. These were used as the training set, which consisted of 50 000 items, and the test set consisted of 10 000 items.

Models: As local and attack models, we selected VGG19[21], ResNet-50[22], Inception[23], SENet[24] FcaNet[25] and a transformer-based model SimpleViT[26] . These models include both the most traditional models and the most advanced models, such as SENet and FcaNet. The use of these models as attack and local models is highly relevant; if our adversarial examples exhibit good transferability in these models, it is likely that good transferability will also be exhibited in a realistic black box. All of these models are trained with Adam optimizer with learning rate 0.001.

Baseline: In one-step attacks, FGSM is used as the baseline. In a multi-step attack, BIM and ATA serve as baselines. Our main goal is to design a more transferable method in order to achieve better results in black-box settings. Szegedy *et al*[8] have discovered that adversarial examples are transferable. Although white-box attacks like FGSM is not designed for black-box settings, the adversarial examples it produced can work in black-box settings with limited effect. Therefore, we also choose FGSM as baseline to show how effective our design is.

Metrics: We use the accuracy of model to evaluate the effectiveness of our attack. The lower this metric is, the more successful our attack proves to be. We will skip the data originally misclassified by the model in order to make sure that there are no other factors that lead to wrong outputs. To evaluate the size of the perturbation, we use the L2 distance between original images and adversarial examples. L2 distance can be described as Formula (11). x is the clean image and y is the adversarial example. Larger distance represents a larger generated adversarial perturbation.

$$\| x,y \|_2 = \sqrt{\sum_i^n \left(x_i - y_i \right)^2} \tag{12}$$

4.2 The Effectiveness of One-Step Attack

Tables 1 and 2 show the transferability of a one-step attack with sensitive local model features. Clearly, our method outperforms FGSM by 3%-8%. We tested it with hyperparameters of 0.1, 0.05, and 0.01, respectively. The local model is represented in the first column. Ours represents the method we propose, while the plain in the table represents the accuracy of attacked models when they are not attacked.

Table 1 Transferability of one-step attack with ResNet-50's sensitive features

%

Method	ε=0.1			ε=0.05			ε=0.01		
	SENet	VGG19	Inception	SENet	VGG19	Inception	SENet	VGG19	Inception
Plain	87	85.63	87.330	87	85.63	87.33	87	85.63	87.33
FGSM	39.53	41.09	38.720	46.47	48.34	46.94	55.50	57.03	58.09
Ours	**35.13**	**39.13**	**36.875**	**41.25**	**43.88**	**45.88**	**47.75**	**51.50**	**54.37**

Table 2 Transferability of one-step attack with Inception's sensitive features

%

Method	ε=0.1			ε=0.05			ε=0.01		
	SENet	VGG19	ResNet-50	SENet	VGG19	ResNet-50	SENet	VGG19	ResNet-50
Plain	87	85.63	86.37	87	85.63	86.37	87	85.63	86.37
FGSM	40.68	40.40	37.90	47.13	47.30	44.38	56.47	57.19	53.33
Ours	**36.13**	**37.38**	**32.37**	**40.75**	**43.33**	**39.12**	**48.63**	**50.25**	**46.87**

Next, we discuss transferability performance under various feature maps. We apply the feature maps generated by various models to the same local network and discover that transferability performs differently. As shown in Table 3, we have highlighted the items with the highest mobility performance with bold font. We initially hypothesized that the attack performance may vary due to the structure and performance of the local network, so we introduced the new FcaNet, whose performance on ImageNet and CIFAR-10 datasets was the best in this experiment. The FcaNet-generated feature maps did not aid in improving our performance. To be more specific, none of the network's feature maps achieved the best mobility. Consequently, the experiment demonstrates the necessity of feature fusion. In Table 3, the hyperparameter (perturbation budget) ε is set to 0.1.

Table 3 The effect of different network sensitive feature maps on transferability

%

Local network	Mask	Inception	SENet	VGG19	FcaNet
ResNet-50	Inception	35.630	35.875	37.13	**42.50**
	SENet	35.400	36.875	**36**	42.75
	ResNet-50	36.875	35.130	39.13	43.90
	FcaNet	**35**	**34.980**	36.73	42.80
Inception	Inception	**32.13**	35.73	37.15	44.03
	SENet	33.33	**33.43**	**35.75**	**43.37**
	ResNet-50	32.34	34.79	38.25	44.42
	FcaNet	33.45	36.23	35.98	43.50

The experiments for one-step feature fusion based attack mentioned in Section 3.2 are displayed in Tables 4 and 5. Whereas we obtain data for experiments involving multiple feature fusion by averaging the experiments five times. This is due to the fact that the parameter settings may lead to fluctuations in the experimental results. In terms of transferability, Tables 4 and 5 demonstrate that our new feature fusion method is significantly superior to the FGSM method and has a degree of improvement over single features. Few cases will exhibit transferability similar to the single-feature case.

Table 4 The effect of feature fusion on transferability with $\varepsilon = 0.1$

%

Local Network	Method	SENet	VGG19	Inception	FcaNet
ResNet-50	Plain	87	85.63	87.330	90.32
	FGSM	39.53	41.09	38.720	50.19
	Single features	35.13	39.13	36.875	43.90
	Feature fusion	**32.87**	**35.87**	**26.180**	**40.37**
Inception	Plain	87	85.63	86.37	90.32
	FGSM	40.68	40.40	37.90	50.69
	Single features	36.13	37.38	32.37	**44.03**
	Feature fusion	**32.03**	**33.50**	**30.90**	45.09

4.3 The Effectiveness of Multi-Step Attack

The experimental results of multi-step attacks based on feature fusion are presented in Tables 6 and 7. We can see that the ATA method performs better than BIM method, whereas our method performs even better than ATA method. It demonstrates an increase in the transferability of the adversarial examples we generate. The perturbation budget size adopted here is 0.01. When the perturbation is too great, the distance between the adversarial example and the original image becomes too great. Table 6 is the result on CIFAR-10 dataset. The transferability of our method is steadily better than that

Table 5 The effect of different network sensitive feature maps on transferability with $\varepsilon = 0.05$

%

Local network	Method	SENet	VGG19	Inception	FcaNet
ResNet-50	Plain	87	85.63	87.33	90.32
	FGSM	46.47	48.34	46.94	52.13
	Single features	**41.25**	43.88	45.88	46.23
	Feature fusion	41.62	**42.78**	**44.67**	**44.75**
Inception	Plain	87	85.63	86.37	90.32
	FGSM	47.13	47.30	44.38	52.44
	Single features	40.75	43.33	39.12	46.39
	Feature fusion	**39.50**	**42.30**	**38.87**	**45.62**

Table 6 The effect of multi-step attack on transferability of CIFAR-10

%

Local network	Mask	Plain	BIM	ATA	Ours
ResNet-50	SENet	87	46.37	45.250	**41.75**
	VGG19	85.63	49.50	46.375	**46.23**
	Inception	87.33	49.79	45.250	**44.85**
	FcaNet	90.32	48.06	47.780	**46.87**
Inception	SENet	87	47.75	42.78	**40.32**
	VGG19	85.63	46.53	41.50	**40.25**
	ResNet-50	86.37	41.31	39.75	**38.25**
	FcaNet	90.32	47.93	45.12	**43.50**

Table 7 The effect of multi-step attack on transferability of MINST

%

Local network	Mask	Plain	BIM	ATA	Ours
ResNet-50	SENet	99.04	73.77	62.07	**60.82**
	VGG19	99	82.33	**68.67**	69.53
	Inception	99.64	75.54	71.26	**68.75**
	ViT	98.40	45.04	29.87	**28.33**
Inception	SENet	99.04	83.84	61.23	**59.50**
	VGG19	99	81.29	75.64	**73.92**
	ResNet-50	98.87	82.84	78.33	**75.20**
	ViT	98.40	47.93	31.09	**29.78**

of ATA. Table 7 is the result on MINST dataset. All the models perform greatly in MINST dataset so we increase the perturbation budget to 0.2 to increase the effect of attack.

Attacked models also include a transformer-based model, ViT(Vision Transformer). Table 8 shows a discussion of adversarial examples ' invisibility. Invisibility means adversarial examples should not be discovered by the human eye. If human can notice the difference between original picture and adversarial examples, the attack will be detected. L2 distance can be used to measure the distance between the original image and the adversarial example. A greater distance indicates a larger perturbation. A larger perturbation means the adversarial example can be easily detected. Our method is approximately equivalent to the distance between the FGSM and the original graph. This proves that our method performs similarly to other methods in terms of concealment. In fact, if the perturbation budget ε is set to 0.01, the distance between the final adversarial example and the original image is not too far from the FGSM.

Table 8 The distance of different methods with $\varepsilon = 0.01$

Local network	Method	Distance
ResNet-50	FGSM	10.74
	BIM	10.76
	ATA	10.78
	Ours	10.76
Inception	FGSM	10.72
	BIM	10.77
	ATA	10.79
	Ours	10.77

5 Conclusion

Based on the fact that Grad-Cam is effective at extracting sensitive features from a network and that the sensitive features extracted by different models are primarily similar, this paper first proposes a one-step attack method against sensitive features . On the basis of the limitation that the one-step attack against sensitive features only considers the feature maps generated by the local model, a more transferability one-step attack is proposed by fusing the feature maps generated by various deep learning models. It is also demonstrated to be more transferable than FGSM and is guaranteed to be just effective. This paper also finds that the ATA multi-step attack highly resembles the ideas presented in this paper, and ATA also does not fuse multiple features. Therefore, we propose a multi-step attack method that combines multiple features. It is experimentally more transferable than the adversarial examples generated by ATA by 1% -3%.

References

[1] He Y, Zhao N, Yin H X. Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach[J]. *IEEE Transactions on Vehicular Technology*, 2018, **67**(1): 44-55.

[2] Zhao D B, Chen Y R, Lv L. Deep reinforcement learning with visual attention for vehicle classification[J]. *IEEE Transactions on Cognitive and Developmental Systems*, 2017, **9**(4): 356-367.

[3] Wang X, Yang W, Weinreb J, *et al*. Searching for prostate cancer by fully automated magnetic resonance imaging classification: deep learning versus non-deep learning[J]. *Scientific Reports*, 2017, **7**(1): 15415.

[4] Xiong H Y, Alipanahi B, Lee L J, *et al*. The human splicing code reveals new insights into the genetic determinants of disease[J]. *Science*, 2015, **347**(6218): 1254806.

[5] Ching T, Himmelstein D S, Beaulieu-Jones B K, *et al*. Opportunities and obstacles for deep learning in biology and medicine[J]. *Journal of the Royal Society Interface*, 2018, **15** (141): 20170387.

[6] Branson K. A deep (learning) dive into a cell[J]. *Nature Methods*, 2018, **15**(4): 253-254.

[7] Deng Y, Bao F, Kong Y Y, *et al*. Deep direct reinforcement learning for financial signal representation and trading[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **28**(3): 653-664.

[8] Szegedy C, Zaremba W, Sutskever I, *et al*. Intriguing properties of neural networks[EB/OL]. [2021-12-06]. http://www.arXiv:1312.6199.

[9] Kurakin A, Goodfellow I J, Bengio S. Adversarial examples in the physical world[EB/OL]. [2022-02-15]. http://www.arXiv:1607.02533.

[10] Carlini N, Wagner D. Towards evaluating the robustness of neural networks[C]//2017 *IEEE Symposium on Security and Privacy* (*SP*). Washingtong D C: IEEE, 2017: 39-57.

[11] Xie C H, Zhang Z S, Zhou Y Y, *et al*. Improving transferability of adversarial examples with input diversity[C]//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Washingtong D C: IEEE, 2019: 2725-2734.

[12] Wu W B, Su Y X, Chen X X, *et al*. Boosting the transferability of adversarial samples via attention[C]//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Washingtong D C: IEEE, 2020: 1158-1167.

[13] Selvaraju R R, Cogswell M, Das A, *et al*. Grad-CAM: Visual explanations from deep networks via gradient-based localization[C]//*Proceedings of the IEEE International Conference on Computer Vision*. Washingtong D C: IEEE, 2017: 618-626.

[14] Guo C, Gardner J R, You Y R, *et al*. Simple black-box adversarial attacks[EB/OL]. [2019-05-17]. https://doi. org/10.48550/arXiv.1905.07121.

[15] Dong Y P, Pang T Y, Su H, *et al*. Evading defenses to transferable adversarial examples by translation-invariant attacks [C]// *The IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*). Washingtong D C: IEEE, 2019: 4307-4316.

[16] Wu W B, Su Y X, Lyu M R, *et al*. Improving the transferability of adversarial samples with adversarial transformations [C]//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Washingtong D C: IEEE, 2021: 9020-9029.

[17] Papernot N, McDaniel P, Jha S, *et al*. The limitations of deep learning in adversarial settings[C]//2016 *IEEE European Symposium on Security and Privacy* (*EuroS&P*). Washington D C: IEEE, 2016: 372-387.

[18] Zhou W, Hou X, Chen Y, *et al*. Transferable adversarial perturbations[C]//*Proceedings of the European Conference on Computer Vision* (*ECCV*). Washingtong D C: IEEE, 2018: 452-467.

[19] Krizhevsky A. *Learning Multiple Layers of Features from Tiny Images*[D]. Tront: University of Tront, 2009.

[20] LeCun Y, Bottou L, Bengio Y, *et al*. Gradient-based learning

applied to document recognition[J]. *Proceedings of the IEEE*, 1998, **86**(11): 2278-2324.

[21] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[EB/OL]. [2022-09-15]. http://www.arXiv:1409.1556.

[22] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C]//*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washingtong D C: IEEE, 2016: 770-778.

[23] Szegedy C, Vanhoucke V, Ioffe S, *et al*. Rethinking the inception architecture for computer vision[C]//*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washington D C: IEEE, 2016: 2818-2826.

[24] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washingtong D C: IEEE, 2018: 7132-7141.

[25] Qin Z Q, Zhang P Y, Wu F, *et al*. FCAnet: Frequency channel attention networks[C]//*Proceedings of the IEEE/CVF International Conference on Computer Vision*. Washingtong D C: IEEE, 2021: 763-772.

[26] Beyer L, Zhai X, Kolesnikov A. Better plain ViT baselines for ImageNet-1k[EB/OL]. [2021-12-05]. http://www. arXiv: 2205.01580, 2022.

□