Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/41/20230737

Robustness analysis of Visual Transformer based on adversarial attacks

Yuqi Fan[1], Fuzhi He[2], and Zibang Nie[3,4]

[1]College of Computer and Control Engineering, Northeast Forestry University, Harbin, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin, China
[3]Dundee International Institute of Central South University, Central South University, Changsha, China

[4]Gardenia@csu.edu.cn

Abstract. As model architectures deepen, large models (e.g., Visual Transformer) perform increasingly well on vision tasks. Adversarial attack is an important test to measure the robustness of the model. By adding noise information to the data, it interferes with the discriminative ability of the model. Previous studies have found that adversarial attacks significantly impact small models (e.g., VGG16, ResNet18), while further tests are needed for interference on large models. This paper conducts three experiments to examine the performance of Visual Transformer (ViT) models against adversarial attacks. In Experiment 1, this paper uses three different attack methods (FGSM, I-FGSM, and MI-FGSM) to test the performance of ViT and some small models. In Experiment 2, this paper tests whether the ViT could distinguish noisy data successfully attacked on the small models. In Experiment 3, this paper examines the defense performance of the ViT, and VGG16 retrained on noisy data. The results show that (1) compared to small models, ViT does have a more vital ability to resist noisy data; (2) the performance improvement of ViT could be better than that of the small model after retraining.

Keywords: Adversarial Attack, Large Model, Vision Transformer.

1. Introduction

With the high demand for deep learning, more and more models are moving toward more extensive parameters and architectures. However, the emergence of adversarial example attacks causes broad worry about the reliability of large models. Therefore, the research to adapt models to potential test data attacks is becoming increasingly significant.

Researchers have been working to improve the algorithms of training and the application of models for a long time. Nowadays, models are not only designed to process image datasets but also text-type datasets. Therefore, there is an obvious truth that while the model scale is becoming more extensive and complex, the methods to attack models are becoming more varied.

Research on adversarial attack help models to be more potent in defending from data noise and purposive misleading on datasets. In recent years, research on adversarial attacks has been systematically summarized by Akhtar et al., and Machado et al. contributed to putting forward some

DOI: 10.54254/2755-2721/41/20230737

metric principles about defense from adversarial attacks [1,2]. They also summarize the development of adversarial learning on image classification [3]. Zhou et al. surveyed examples of adversarial attacks and divided them into four categories [3, 4]. Besides, many research conclusions can be connected to the experiments. Kurakin et al. put forward a transitive attack [3]. It emphasizes that an attack dataset designed for one model can mislead other models simultaneously, which means adversarial attacks are universal [5]. In 2018, adversarial training was put forward by Ma et al. It shows that the combination of noise and original data can improve the accuracy of the test [6]. As mentioned in the former paragraph, the research about adversarial attacks is closely connected with the scale of the model and the fast increase of parameters in a model shown in recent years. Therefore, some research introduces the connection between model scale and adversarial attack. In 2016, Nicolas Papernot and other experts used 'Distillation' to establish a connection between a larger model and a smaller model to transfer data. The result shows distillation can help the more minor model defense attack [7]. As for the scale's effect, it can be found in Shafahi's research that a large model performs better when facing an adversarial attack. However, the increasing model scale cannot perform better when defending against attack at any time. Gilmer et al. put forward this opinion in 2018 [8].

Adversarial attacks play an essential role in computer science. Hence, this paper aims to find the performances of different models, especially for large and small models, when they are attacked with variable noises. This paper also concentrates on searching for features that affect models' performance when being attacked, such as the connections between the scale of models and their ability to defend against attacks.

The core idea of the experiment is to apply gradient information to add noises to the dataset to test the robustness of models. Experiments are divided into three parts. The CIFAR-10 dataset is applied in all tasks, and epsilons of our test are set in the range of 0.05 to 0.3. Besides, task models contain LeNet, VGG, ResNet, and ViT. In the first task, some models are trained with a standard training dataset and tested with a dataset attacked by three attack methods (FGSM, I-FGSM, and MI-FGSM). This paper collects the wrong classification results of several models tested with an FGSM attack in the second task. After that, these outcomes are used to test other models for accuracy. In the final task, this paper performs adversarial training on VIT and VGG16 by adding noise to the training dataset. These models are tested by FGSM attack. After some experiments, this paper finds some probable conclusions:

1. The model performs better when the number of parameters in this model is more than other models.

2. When adding noise to the training dataset, smaller models significantly improve more than larger models because a larger model catches deeper data characteristics while the noise is set at the surface level of a dataset.

2. Background

2.1. ViT model

ViT is an essential model because of its outstanding performance in many computer vision tasks. So far, many experiments have been conducted on ViT [9, 10]. The processing flow of a ViT model usually consists of four main parts: patch embeddings, learnable embedding, position embeddings, and Transformer encoder. The Transformer encoder consists of alternating layers of Multi-head Self-attention and Multi-layer Perception blocks among these four parts. The embedded input vector comprises image block embedding, class vector, and position coding. Multi-head Self-attention Block comprises a multi-head attention mechanism, layer normalization, and skip connection, while Multi-layer Perception Block comprises a feedforward network (FFN), Layer normalization, and skip connection. The output image is represented by the layer normalization and classification head [11].

DOI: 10.54254/2755-2721/41/20230737

2.2. Adversarial Attacks

Adversarial attacks specifically refer to adding a slight perturbation to the adversarial example, which can be seen by human eyes, but will affect the model, so the DNN model will output wrong results [12]. There are many common Attack methods, such as Fast Gradient Sign Method (FGSM), Iterative FGSM (I-FGSM), Momentum Iterative Fast Gradient Sign Method (MI-FGSM), Carlini & Wagner (C&W attack), and DeepFool attack. The following is a brief introduction to these ways.

Gradient-based methods, but only one step is required, such as FGSM [13]. FGSM is a single attack method based on gradient optimization, originally used to speed up the generation of adversarial samples in adversarial training. Linearizing the loss function around the current value of the model parameter x can obtain the optimal max-norm constrained perturbation [12].

Iterative methods, such as I-FGSM, MI-FGSM and DeepFool. I-FGSM is an improved way of FGSM. The smaller step size is applied multiple times. Each pixel grows (or decreases) based on the adversarial example in the previous step, and the pixel value of the intermediate result is clipped to ensure that the adversarial example generated in each step falls into the legal range [13, 14]. DeepFool is based on the gradient iteration method to generate the least disturbance and can have a high attack accuracy. The sample robustness and model robustness mirrors are defined for the first time, and they can accurately calculate the perturbation of depth classifiers to large-scale data sets, thus reliably quantifying the robustness of classifiers [15].

Momentum iterative gradient-based methods, such as MI-FGSM. It is a momentum iterated gradient based method [13]. I-FGSM method has weak transferability, to solve this problem, momentum is incorporated into the iterative attack I-FGSM, which can be used to stabilize the update direction and thus improve transferability [16].

Optimization-based methods. These methods can calculate the differences between original samples and noise samples, but the transferability of these methods could be stronger [13]. C&W attack is an example. It has both high attack accuracy and low counter disturbance. It can be realized that the model classification error is invisible to the human eye [17].

3. Methods

To test the defense capability and the robustness of the large model, this paper conducts three experiments about adversarial attacks. In Experiment 1, to check whether the large model has a particular advantage in resisting adversarial attacks, this paper attacked some related small and ViT models with different degrees, respectively. In Experiment 2, this paper puts the successful attack samples of one model into another model and observes the classification ability of other models for these noise samples. In Experiment 3, this paper processes partial data in the training set with noise and re-trained the VGG16 and ViT models. After that, this paper tests the two models against adversarial attacks and observes if the performance of the models against noise is improved. In Experiment 1, this paper uses three attack tests: FGSM attacks, I-FGSM attacks and MI-FGSM attacks. In Experiments 2 and 3, only FGSM is used.

Fast Gradient Sign Method (FGSM). FGSM is one of the simplest image adversarial attack techniques, which is a direct and effective algorithm for generating adversarial examples. As shown in equation (1), let θ be the model parameters, $\boldsymbol{X}$ be the image input, $\boldsymbol{y_{true}}$ be the corresponding label, and $J(\theta, X, y_{true})$ be the loss function. This equation can be linearized around the current value of θ loss function, to obtain $\varepsilon sign(\nabla_x J(\theta, X, y_{true}))$ max-norm perturbation.

$$X^{adv} = X + \varepsilon \cdot sign\big(J(\theta, X, y_{true})\big) \tag{1}$$

In FGSM algorithm, the perturbed signal values are searched in the ε neighbourhood of the original image so that the adversarial sample $\boldsymbol{X^{adv}}$ is classified as an incorrect label t. FGSM learns input image features during network training and obtains classification probabilities through SoftMax or Sigmoid layers. The resulting classification probabilities and true labels are then used to calculate the loss value, return the loss value, and calculate the gradient, which is gradient backpropagation. When the changed

DOI: 10.54254/2755-2721/41/20230737

image is fed into the classification network, add the calculated gradient direction to the input image to make the loss more significant than that of the whole image.

Iterative Fast Gradient Sign Method (I-FGSM). I-FGSM is an upgraded version of FGSM. As shown in equation (1), FGSM adds one step of perturbation along the direction of increasing gradient, so each pixel changes only once. As an improvement, in I-FGSM, multiple small perturbations are added in the direction of gradient increase in an iterative manner, and the gradient direction is re-calculated after each small step. I-FGSM can achieve more accurate migration, but the computational complexity is also increased.

$$X_{N+1}^{adv} = Clip_{X,\varepsilon}\left\{X_N^{adv} + \alpha \cdot sign\left(\nabla_x J\left(\theta, X_N^{adv}, y_{true}\right)\right)\right\} \tag{2}$$

The calculation formula of I-FGSM is given in equation (2), where $X_0^{adv} = X$. $\boldsymbol{X}$ is the original image, and X_N^{adv} is the adversarial sample processed by the FGSM algorithm N times. $Clip_{X,\varepsilon}(A)$ clips each element $A_{i,j}$ of the input vector $\boldsymbol{A}$ to between $[X_{i,j} - \varepsilon, X_{i,j} + \varepsilon]$. $sign(\nabla_x J\left(\theta, X_N^{adv}, y_{true}\right))$ is consistent with the expression evaluation in FGSM, and α represents the magnitude of the pixel change in each iteration. If the alpha set to 1, in each iteration, pixel update only $sign(\nabla_x J\left(\theta, X_N^{adv}, y_{true}\right))$ or $-sign(\nabla_x J\left(\theta, X_N^{adv}, y_{true}\right))$.

In I-FGSM, each pixel grows (or decreases) α on the basis of the confrontation sample in the previous step, and then implements the clipping operation so that each pixel of the original sample ($X_{i,j}$) changes in the ε neighbourhood. When each pixel changes less than ε, I-FGSM is easy to produce the corresponding noise samples, otherwise, I - FGSM will degenerate into FGSM.

Momentum Iterative Fast Gradient Sign Method (MI-FGSM). MI-FGSM uses the momentum to generate noise examples and benefit from them. The momentum method is a technique that achieves acceleration by accumulating a velocity vector along the gradient direction during iterations. The momentum method also shows its effectiveness for stable updates by remembering previous gradients and traversing poor local minima or maxima. As shown in equation (3) and (4), the momentum is integrated into the I-FGSM to stabilize the update direction and avoid local maxima of the difference.

$$g_{N+1} = \mu g_N + \frac{\nabla_x J\left(\theta, X_N^{adv}, y_{true}\right)}{\| \nabla_x J\left(\theta, X_N^{adv}, y_{true}\right) \|_1} \tag{3}$$

$$X_{N+1}^{adv} = X_N^{adv} + \alpha \cdot sign(g_{N+1}) \tag{4}$$

The parameter g_N is the cumulative gradient of the first N iterations and $g_0 = 0$ is initialized. Specifically, the gradient of the previous iteration is collected using the decay factor μ defined in equation (3). If μ is set to 0, MI-FGSM will degenerate to I-FGSM. At each iteration, the current gradient $\nabla_x J(\theta, X, y_{true})$ is normalized by its own L_1 distance, since the magnitude of the gradient varies from iteration to iteration.

4. Experimental results and analysis

This paper conducts three experiments about adversarial attacks to test the defense capability and robustness of the large model. This paper chooses cifar-10 as the dataset for this experiment. The cifar-10 dataset consists of 50,000 training images and 10,000 test images. Each image is a 32x32 RGB image. The dataset has ten classes: truck, plane, car, bird, cat, deer, dog, frog, horse, and cat, and has high complexity and a strong representation. For all the experiments, this paper pre-processed the input data to the size of 224x244 when training the model. Before training, all models except the LeNet were loaded with version parameters pre-trained on the IMAGENET dataset.

4.1. Adversarial attack test

In Experiment 1, to check whether the large model has a particular advantage in resisting adversarial attacks, this paper uses test data with different levels of noise contamination to test the model's

DOI: 10.54254/2755-2721/41/20230737

classification accuracy. This paper trains LeNet, VGG16, ResNet18, ResNet34, and ViT. LeNet and VGG16 are trained for ten epochs, while ResNet18, ResNet34, and ViT are trained for five epochs. In the test phase, this paper does not attack the data misclassified by the model but only adds noise to the correctly classified data. The ε values are 0.05, 0.1, 0.15, 0.2, 0.25 and 0.3, respectively. It must be noted that when using I-FGSM attacks, the iteration number N is set to 10, and α is set to ε⁄N. When using MI-FGSM attacks, the iteration number N and the parameter α are ten and ε⁄N, and the decay factor μ is set to 1. Tables 1 to 3 show the results of the Experiment 1.

Table 1. Classification ACC of different models against different levels of FGSM.

Epsilon	0	0.05	0.1	0.15	0.2	0.25	0.3
LeNet	0.6142	0.1145	0.0738	0.0543	0.0392	0.0319	0.0256
VGG16	0.9337	0.2282	0.2121	0.1905	0.1761	0.1639	0.1571
ResNet18	0.9532	0.2358	0.2062	0.1879	0.1722	0.1619	0.1602
ResNet34	0.9609	0.2949	0.2618	0.2169	0.1858	0.1679	0.1554
ViT	**0.9651**	**0.3551**	**0.2777**	**0.2353**	**0.2051**	**0.1817**	**0.1734**

Table 2. Classification ACC of different models against different levels of I-FGSM.

Epsilon	0	0.05	0.1	0.15	0.2	0.25	0.3
LeNet	0.6142	0.2112	0.1919	0.1724	0.1551	0.1426	0.1305
VGG16	0.9337	0.3458	0.3116	0.2897	0.2677	0.2572	0.2462
ResNet18	0.9532	0.3159	0.2936	0.2805	0.2698	0.2610	0.2563
ResNet34	0.9609	0.3739	0.3547	0.3450	0.3363	0.3264	0.3258
ViT	**0.9651**	**0.5463**	**0.5145**	**0.4890**	**0.4597**	**0.4381**	**0.4182**

Table 3. Classification ACC of different models against different levels of MI-FGSM.

Epsilon	0	0.05	0.1	0.15	0.2	0.25	0.3
LeNet	0.6142	0.2116	0.1956	0.1743	0.1611	0.1448	0.1321
VGG16	0.9337	0.3463	0.3065	0.2862	0.2678	0.2571	0.2498
ResNet18	0.9532	0.3168	0.2944	0.2834	0.2704	0.2600	0.2559
ResNet34	0.9609	0.3754	0.3575	0.3451	0.3353	0.3266	0.3191
ViT	**0.9651**	**0.5450**	**0.5159**	**0.4892**	**0.4615**	**0.4368**	**0.4117**

Tables 1 to 3 show that the FGSM, I-FGSM, and MI-FGSM are practical for large and miniature models. With the increase of the complexity of the model, the ability of the model to defend against attacks also improves. It can be observed that ViT achieves the best classification accuracy for different values of ε, which means ViT has the most vital ability to deal with different degrees of attacks.

Therefore, this paper speculates that increasing the model's complexity can improve the model's ability to resist data noise attacks. With the deepening of the number of layers, the model is better at extracting deep features of the data. This is beneficial for the model to recognize the data by analyzing the deep features, ignoring the influence of noise to a certain extent. From this, it can be preliminarily inferred that the robustness of the model with many layers is better. However, these attack methods still show high lethality in large models, so the data pollution problem also exists in large models. Simply increasing the complexity of the model to resist attacks is still very limited.

4.2. Comparative attack test

In Experiment 2, to further demonstrate that the large model outperforms the small model in identifying contaminated data, this paper did the following tests: this paper inputs the data successfully attacked A model (that is, the noisy data misclassified by A model) into different B models to test the

DOI: 10.54254/2755-2721/41/20230737

classification ability of B model on these noisy data. LeNet, ResNet18, ResNet34, and ViT are used as models A and B, respectively. Tables 4 to 7 show the results of the Experiment 2. The column with a value of 0 means that the model (A model) misclassified these noisy data, and this paper fed these data directly into the other models (B models). In this experiment, only the FGSM attack mode is used.

Table 4. Classification ACC of other models for noisy data misclassified by Lenet.

Epsilon	LeNet	ResNet18	ResNet34	ViT
0.05	0	0.326	0.369	**0.580**
0.1	0	0.311	0.355	**0.549**
0.15	0	0.278	0.318	**0.524**
0.2	0	0.256	0.284	**0.487**
0.25	0	0.234	0.264	**0.458**
0.3	0	0.218	0.250	**0.442**

Table 5. Classification ACC of other models for noisy data misclassified by ResNet18.

Epsilon	LeNet	ResNet18	ResNet34	ViT
0.05	0.195	0	0.210	**0.487**
0.1	0.203	0	0.179	**0.483**
0.15	0.205	0	0.143	**0.473**
0.2	0.210	0	0.120	**0.449**
0.25	0.212	0	0.111	**0.427**
0.3	0.204	0	0.102	**0.404**

Table 6. Classification ACC of other models for noisy data misclassified by ResNet34.

Epsilon	LeNet	ResNet18	ResNet34	ViT
0.05	0.183	0.148	0	**0.446**
0.1	0.192	0.135	0	**0.451**
0.15	0.200	0.123	0	**0.449**
0.2	0.204	0.112	0	**0.440**
0.25	0.211	0.100	0	**0.421**
0.3	0.213	0.090	0	**0.396**

Table 7. Classification ACC of other models for noisy data misclassified by ViT.

Epsilon	LeNet	ResNet18	ResNet34	ViT
0.05	0.167	0.203	**0.212**	0
0.1	0.182	0.179	**0.193**	0
0.15	**0.186**	0.159	0.164	0
0.2	**0.192**	0.141	0.139	0
0.25	**0.185**	0.131	0.126	0
0.3	**0.186**	0.122	0.115	0

Tables 4 to 7 show that ViT can correctly classify noisy data misclassified by small models (e.g., LeNet, ResNet18, and ResNet34). LeNet is the model with the simplest structure in this experiment. ViT has about 50% reclassification accuracy for noisy data misclassified by LeNet. However, as the complexity of the model increases (from LeNet to ResNet34), the reclassification accuracy of ViT also decreases. Table 7 shows that the small models need to play a better role in correcting the large model.

DOI: 10.54254/2755-2721/41/20230737

This paper speculates that those noisy samples that can successfully attack small models have limited effectiveness against large models. The reason is similar to what this paper concluded in Experiment 1. Large models have a more vital ability to extract deep features, which helps the model identify data categories from a higher dimensional space. Therefore, for the attack samples acting on the small models, the large models can be identified, and this inspires us that large models may play a specific role in assisting the discrimination of miniature models. At the same time, it can be observed that although the accuracy of the LeNet model is low in reclassification, it shows some stability. That is, the accuracy does not decrease significantly with the increase of epsilon. This may be because when the model is simple enough, the extracted information is limited and not sensitive to different noise levels.

4.3. Adversarial training

In Experiments 1 and 2, this paper observes that the performance of the large model is better than that of the small models when dealing with the noise attack. However, the performance degradation after the attack is still very significant. The natural idea is to train the model with some noisy data and let the model learn the features of these noisy data to improve its resistance. In Experiment 3, this paper retrains seven ViT models. This paper randomly samples 5% of the 50,000 images from the cifar10 training set and adds different noise levels to form a new training set (45,000 raw images and +5000 noisy images). Six ViT models are retrained for different noise values (epsilon is 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30, respectively). At the same time, for each level of noise, 2% of the training data is extracted for noise processing and mixed and added to the original training set (34000 original data +6000 data containing various types of noise) to train the seventh model. All ViT models are trained with epoch five and learning rate 1e-4. This paper next uses FGSM attacks to test these seven ViT models. The specific parameter setting is consistent with the FGSM test part in Experiment 1. Table 8 shows the results. The data of the ViT-base is from Experiment 1.

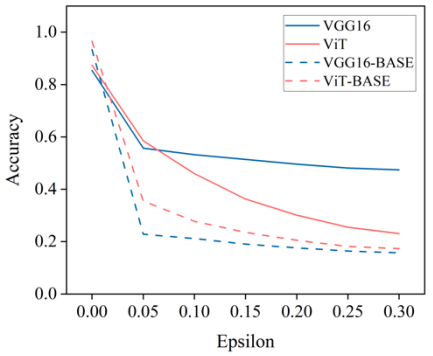
Table 8. Adversarial attack testing of ViT-Base and new ViT models retrained with different levels of noise.

	0.0	0.05	0.10	0.15	0.20	0.25	0.30
ViT-base	**0.965**	0.355	0.278	0.235	0.205	0.182	0.173
ViT (ε: 0.05)	0.874	**0.585**	**0.460**	0.363	0.301	0.255	0.231
ViT (ε: 0.10)	0.825	0.506	0.447	0.405	0.370	0.344	0.327
ViT (ε: 0.15)	0.867	0.492	0.452	0.425	0.403	0.384	0.372
ViT (ε: 0.20)	0.841	0.466	0.446	0.428	0.416	0.404	0.392
ViT (ε: 0.25)	0.850	0.428	0.410	0.399	0.390	0.379	0.372
ViT (ε: 0.30)	0.873	0.442	0.434	**0.431**	**0.427**	**0.424**	**0.427**
ViT (ε: 0.05-0.30)	0.702	0.420	0.403	0.395	0.385	0.378	0.374

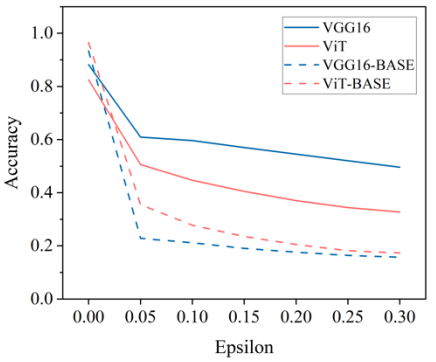
Table 8 shows that the classification accuracy of the retrained ViT model is higher than that of the model before training (ViT-base). However, the classification accuracy is decreased when not under attack. It can be shown that the strategy has a specific effect. Although the accuracy when Epsilon=0 is sacrificed, the resistance to different degrees of attack is improved. This suggests adding noise to the training set to improve the model's defense. Notably, this paper expects that the retrained model would handle all attack levels well when adding all noise levels to the training set (as this paper did when training the seventh ViT model). However, this is not the case, as the seventh model (0.05~0.30 for the adversarial sample retrain) does not perform best at all attack levels. This suggests that the defense performance may not be improved to the best by widely adding all kinds of noise for retraining but may be improved by explicitly adding a specific type of noise (just as the model retrained with only 0.03 noise data shows relatively good performance in this test).

DOI: 10.54254/2755-2721/41/20230737

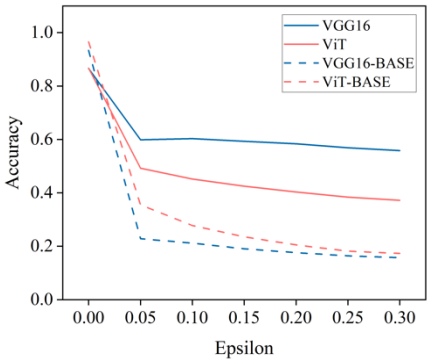
Next, this paper also did the same experiment on the VGG16 model, and all VGG16 models are trained with epoch 10, learning rate 0.001, and momentum 0.9. Figure 1 shows the data comparison of VGG16 and ViT. Figure 1(a)-(f) shows the performance of VGG16 and ViT in adversarial attack tests after retraining with different noise levels except the mixed noise, respectively. Figure 2 shows the performance of VGG16 and ViT in adversarial attack testing after retraining with mixed noise. The data of ViT-base and VGG16-base are from Experiment 1.



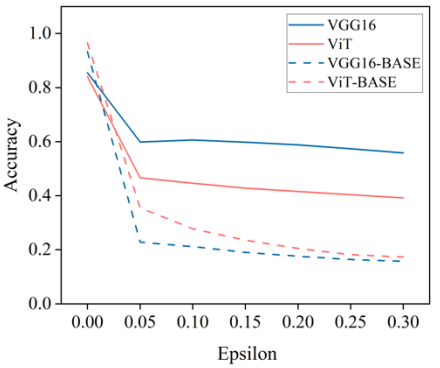
(a) Epsilon: 0.05 for adversarial training



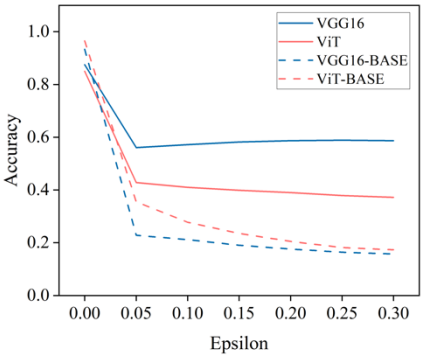
(b) Epsilon: 0.10 for adversarial training



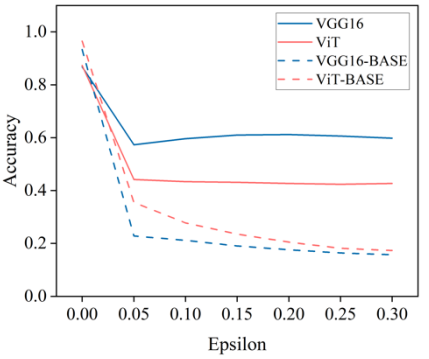
(c) Epsilon: 0.15 for adversarial training



(d) Epsilon: 0.20 for adversarial training



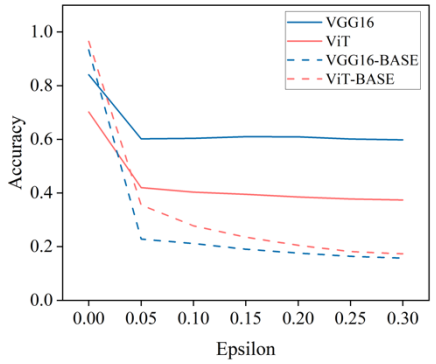
(e) Epsilon: 0.25 for adversarial training



(f) Epsilon: 0.30 for adversarial training

Figure 1. Performance comparison between ViT and VGG16 after retraining. Figure (a)-(f) show the performance of VGG16 and ViT in adversarial attack tests after retraining with different levels of noise except the mixed noise, respectively.

DOI: 10.54254/2755-2721/41/20230737



Epsilon:0.05~0.30 for the adversarial sample retrain VGG16 and ViT

Figure 2. Performance comparison between ViT and VGG16 after retraining. Figure2 shows the performance of VGG16 and ViT in adversarial attack tests after retraining with mixed noise.

It can be seen performance of VGG16 against noise attacks is also improved after retraining. However, it is worth mentioning that no matter what level of noise is used for retraining, VGG16 has higher performance improvement than ViT after retraining. The defense strategy of adding noise to the training set is more effective for small models like VGG16. This paper analyses that FGSM attacks the noise on the surface features of the data. However, in Experiments 1 and 2, this paper believes that large models are better at extracting deep features and, thus, are somewhat resistant to noise. However, in Experiment 3, this paper hopes that the model can learn the information of these noises during the training phase to improve the defense performance. At this point, the features of the large model to extract deep features become a defect, making it difficult for the large model to learn some surface noise information. This makes the defense performance improvement of the large model after retraining limited even worse than that of the small model. Thus, in adversarial attacks, large models are not necessarily superior to small models in every aspect. In the direct resistance to noise data, large models have certain advantages. However, when this paper wants to improve the model's performance by retraining with noise, the improvement of the large model is likely to be lower than that of the small model.

5. Discussion

Large models have more complex architectures and more parameters than small models, so they are better at extracting deep features of the data. Using the information of deep features, the large model makes it easier to distinguish the actual category of the data while ignoring noise interference to a certain extent. It shows that the large model has better robustness compared with the small models tested in the experiments. This advantage of large models is reflected in Experiments 1 and 2. However, it is essential to note that this feature of large models only sometimes provides an all-around advantage. In Experiment 3, this paper adds noisy data to the training set to improve the defense ability of the trained model. The experimental results show that, after retraining, the large model performs worse than some small models in adversarial attacks. This paper believes this is because some adversarial attack methods focus more on attacking shallow features. When resisting adversarial attacks, the large model extracts the features to resist the noise information of shallow features well, but small models cannot. However, when this paper wants the model to learn this noisy information during the training phase, the complex architecture of large models needs to pay attention to learning shallow features, which makes the model unable to learn noise information. The defense strategy of retraining the model with noisy data may be more suitable for small models. For large models, other methods may be needed to improve the defense performance, which is worth exploring by scholars.

There are two main limitations to the experiment. First, more than the comparative experiments are required. In Experiments 2 and 3, this paper only uses the FGSM attack method for testing, and

DOI: 10.54254/2755-2721/41/20230737

the test experiments on I-FGSM and MI-FGSM should be supplemented. In Experiment 3, this paper should tune more parameters to see how the model's performance changes, like adding different amounts of noisy data for the adversarial training. Second, this paper only tests the ViT model for experiments with large models. More tests on different large models, such as the Clip model and the InternImage model, are needed to draw more general conclusions. This is also the future work.

6. Conclusion

This paper conducts three experiments to explore the performance of large models in adversarial attacks. Experiments 1 and 2 show that compared with small models, large models have a more vital ability to resist noise attacks. However, in Experiment 3, it is found that when noise data is added to the training set, the improvement of the resistance to noise attack of the large model is not as good as that of the small model. This paper analyses that this is caused by the noise information in shallow layers that large models cannot quickly learn. Thus, large models do not outperform small models in every way. In future work, this paper will add more complete experiments with more attack methods and experimental parameters and continue to try adversarial attack tests on other large models such as Clip and InternImage.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] N Akhtar, A Mian, N Kardan and M Shah, "Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey," in IEEE Access, vol. 9, pp. 155161-155196, 2021, doi: 10.1109/ACCESS.2021.3127960.

[2] Machado G R, Silva E and Goldschmidt R R. 2021. Adversarial Machine Learning in Image Classification: A Survey Toward the Defender's Perspective. ACM Comput. Surv. 55, 1, Article 8 (January 2023), 38 pages. https://doi.org/10.1145/3485133

[3] Teng L et al. (2022) A survey on adversarial attacks In computer vision: Taxonomy, visualization and future directions. *Computers & Security*, Volume 121.

[4] Zhou Y, Han M, Liu L, He J and Gao X. "The Adversarial Attacks Threats on Computer Vision: A Survey," 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), Monterey, CA, USA, 2019, pp. 25-30, doi: 10.1109/MASSW.2019.00012.

[5] Yampolskiy, R V. (Ed.). (2018). Artificial Intelligence Safety and Security (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781351251389

[6] Aleksander M et al. (2019) Towards Deep Learning Models Resistant to Adversarial Attacks. Available at: https://arxiv.org/abs/1706.06083.

[7] Nicolas P et al. (2016) Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, 2016, pp. 582-597, doi: 10.1109/SP.2016.41.

[8] Nicolas P et al. (2016) The Limitations of Deep Learning in Adversarial Settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Saarbruecken, Germany, 2016, pp. 372-387, doi: 10.1109/EuroSP.2016.36.

[9] Li X and Zhang B -B (2023) FV-ViT: Vision Transformer for Finger Vein Recognition in IEEE Access, vol. 11,2023, pp. 75451-75461, doi: 10.1109/ACCESS.2023.3297212.

[10] Dierickx P, Van Damme A, Dupuis N and Delaby O (2023) Comparison Between CNN, ViT and CCT for Channel Frequency Response Interpretation and Application to G. Fast in IEEE Access, vol. 11, pp. 24039-24052, 2023, doi: 10.1109/ACCESS.2023.3247877.

[11] Dosovitskiy A, Beyer L, Kolesnikov A et al (2020) An Image is Worth 16x16 Words: Transformers for Image Recognition at *Scale.ArXiv, abs/2010.11929*.

DOI: 10.54254/2755-2721/41/20230737

[12] Lyu C, Huang K and Liang H N (2014) Explaining and Harnessing Adversarial Examples. Retrieved from https://doi.org/10.48550/arXiv.1412.6572

[13] Dong Y et al. (2018) Boosting Adversarial Attacks with Momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 9185-9193, doi: 10.1109/CVPR.2018.00957.

[14] Kurakin A, Goodfellow I and Bengio S (2016) Adversarial Examples In *The Physical World* Retrieved from https://doi.org/10.48550/arXiv.1607.02533.

[15] Moosavi-Dezfooli S -M, Fawzi A and Frossard P (2016) DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2574-2582, doi: 10.1109/CVPR.2016.282.

[16] Wang G, Yan H and Wei X(2022) Enhancing Transferability of Adversarial Examples with Spatial Momentum Retrieved from https://doi.org/10.48550/arXiv.2203.13479.

[17] Carlini N and Wagner D (2017) Towards Evaluating the Robustness of Neural Networks *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, 2017, pp. 39-57, doi: 10.1109/SP.2017.49.