Journal of Contemporary Educational Research, 2024, Volume 8, Issue 11
http://ojs.bbwpublisher.com/index.php/JCER
ISSN Online: 2208-8474
ISSN Print: 2208-8466

A Study on Filter-Based Adversarial Image Classification Models

Zhongcheng Zhao*

The Second Branch of The Middle School Affiliated to Beijing Jiaotong University, Beijing 100081, China

***Corresponding author:** Zhongcheng Zhao, zhaozhongchengeric@yeah.net

Abstract: In view of the fact that adversarial examples can lead to high-confidence erroneous outputs of deep neural networks, this study aims to improve the safety of deep neural networks by distinguishing adversarial examples. A classification model based on filter residual network structure is used to accurately classify adversarial examples. The filter-based classification model includes residual network feature extraction and classification modules, which are iteratively optimized by an adversarial training strategy. Three mainstream adversarial attack methods are improved, and adversarial samples are generated on the Mini-ImageNet dataset. Subsequently, these samples are used to attack the EfficientNet and the filter-based classification model respectively, and the attack effects are compared. Experimental results show that the filter-based classification model has high classification accuracy when dealing with Mini-ImageNet adversarial examples. Adversarial training can effectively enhance the robustness of deep neural network models.

Keywords: Adversarial example; Image classification; Adversarial attack

Online publication: November 27, 2024

1. Introduction

With the rapid development of deep learning technology, its applications in many fields such as computer vision [1], speech recognition [2], and natural language processing [3] have widely penetrated our daily life and industrial production. These technologies are revolutionizing industries such as autonomous driving [4], security monitoring [5], and health care [6]. However, the sensitivity of deep neural networks to adversarial attacks has also attracted much attention. Research shows that electronic data may suffer from adversarial attacks during network transmission, which poses a threat to the security of artificial intelligence systems. For example, in image classification tasks, attackers can generate adversarial samples to mislead the classifier into wrong classification results [7].

To cope with this challenge, researchers around the world have developed a variety of defense strategies, which can be mainly divided into two categories. One is aimed at enhancing the robustness of the model to maintain the correct output even under adversarial attacks. The other is to defend by identifying whether the

input sample has been tampered with.

In order to improve the robustness of the model, researchers mainly adopt two methods. One is to reduce the attack effect of adversarial examples in the spatial domain through preprocessing, such as JPEG compression, image rotation, adding noise, and other technologies. The other is to improve the resistance of the network to adversarial samples by adjusting the network architecture or optimizing the training data. Goodfellow *et al.* [8] proposed the method of adding adversarial samples in the training process to improve the robustness of the network against specific attacks. Zantedeschi *et al.* [9] proposed a strategy of using a restricted modified Linear Unit (ReLU) function to reduce adversarial perturbations.

However, due to the difficulty of defending adversarial examples solely by improving their robustness, many researchers have begun to turn to detection-based defense methods. Li *et al.* [10] adopted a cascade method of Principal Component Analysis (PCA) combined with multiple classifiers to identify adversarial samples. In this process, the image must pass the test of all the classifiers in turn, and only the image that passes all the classifiers is considered as the sample without adversarial attack. AdaBoost plays a key role in this process by adjusting the sample weights, misclassified samples are weighted more and correctly classified samples are weighted less, the next base classifier is trained. A new weak classifier is introduced in each iteration until a preset error threshold is reached. Nevertheless, the performance of this strong classifier based on cascade is not satisfactory in actual detection.

Zhang *et al.* [11] explored a new method to improve the detection ability of the deep neural network by adding a special sub-network. This “detector” sub-network can effectively identify adversarial samples on small-scale data sets through adversarial training. However, in the face of large size and small disturbance of the sample, the accuracy of its detection will be significantly reduced.

Lu *et al.* [12] used the difference between the activation patterns of non-adversarial samples and adversarial samples in the neural network for detection. Their method judges the authenticity of an image by comparing the consistency between the original image and the depth image on a small-scale dataset. Others also explored two attack methods, FGSM [13] and DeepFool [14]. The disturbance level of FGSM is determined by the network gradient and the disturbance factor, and the strength of the disturbance can be controlled by adjusting the disturbance factor. However, DeepFool is an attack method based on linear approximation, which generates perturbations by calculating the shortest linear distance between the sample and the class boundary, which makes it difficult for some detection models to effectively identify adversarial samples.

Currently, many open-source adversarial example generation methods, such as AdvGAN and DeepFool, are typically trained on the National Institute of Standards and Technology Hybrid Database (MNIST) [14] and CIFAR-10 [15] datasets. In these methods, the image is usually converted to a single channel before the attack to generate grayscale adversarial samples. This transformation leads to a reduction in the dimensionality of the adversarial examples, which may not be enough to effectively fool the model. Therefore, using these examples to train the adversarial example detection model may not effectively improve the performance of the model. If these single-channel samples are channeled, for example by data stacking, this may change the adversarial effect, as the perturbation is also replicated three times.

To address these issues, this study improves the AdvGAN and G-ATN algorithms by adjusting the feedforward network structure and increasing the normalization of the input image. In the FGSM algorithm, the influence of the adversarial factors on the disturbance generation and model recognition accuracy is studied, and the better effect of the adversarial factors is selected. At the same time, the input channel is set to three channels, and the Mini-ImageNet dataset is used to generate adversarial samples. In order

to enhance the confusion of adversarial samples, the ResNet50 pre-training model is used to screen the generated samples. In addition, based on the generated adversarial examples, a classifier is developed to identify whether an image in a dataset is an adversarial example.

2. Generating adversarial examples

2.1. AdvGAN adversarial attacks

The AdvGAN algorithm, for networks with a single channel input, achieved a 98% attack success rate on the MNIST dataset. The network structure is shown in **Figure 1** [16]. The basic flow of the algorithm is as follows: the original MNIST sample is taken as input, and the generator *G* generates pixel perturbations. The perturbed sample is denoted as *x+G(x)*, which is the result of adding the original sample to the perturbation produced by the generator. This result is then fed into the discriminator *D* (ResNet50 pre-trained model). If the classification label of the discriminator for the superposition result is inconsistent with the classification label of the original sample, the adversarial sample *x** is output. If they agree, continue to adjust the generator and increase the pixel perturbation.

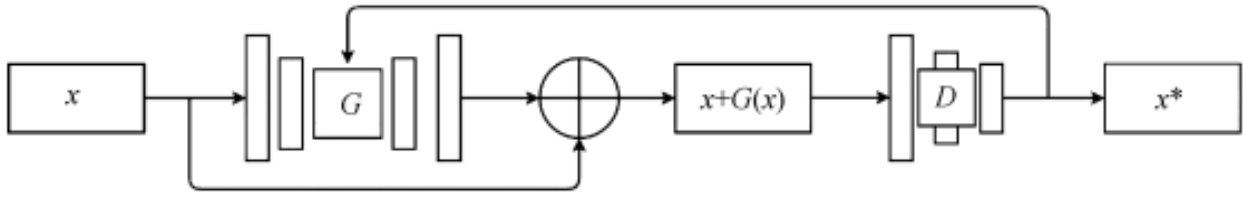

Figure 1. Block diagram of AdvGAN attack method

The specific construction and output results of the generator *G* are presented in **Figure 2**. Based on this model, this study uses the Mini-Image Net dataset, adjusts the input channel of the generator module, and adds a residual processing unit to screen out a more effective image perturbation *R(z)*. In addition, a normalization step of the input image is added to facilitate the extraction of image features and accelerate the generation of perturbations.

The structure of the generator includes three parts: encoder, residual processing, and decoder. The encoder consists of three convolutional layers, which are responsible for processing the original sample data and extracting image features. The residual processing part contains four residual blocks, where the image is combined with the original features by reflection padding to generate more optimal perturbed data. The decoder consists of three deconvolution layers, which aim to restore the dimensions of the original samples. For the discriminator, the ResNet50 pre-trained model is used to generate adversarial examples of the Mini-ImageNet dataset. The loss function used by the classifier is defined as in Equation (1).

$$L_{\text{AdvGAN}} = E_x \log D(x) + E_x \log\left(1 - D\left(x + G(x)\right)\right) \quad (1)$$

Ex in Equation (1) represents the expected value of sample *x*. From Equation (1), it can be seen that the loss function aims to correctly distinguish the original sample data *x* and the data with disturbance *x+G(x)* through discriminator *D*, so that the final generated adversarial sample data is more similar to the original sample data.

Adversarial perturbations are generated by the generator module of the AdvGAN model. The generator

module extracts image features through a convolutional layer and combines them with the original features using the reflection filling of the residual blocks to generate high-quality perturbed data. Subsequently, adversarial perturbations are obtained by restoring the dimensions of the original samples through the decoder. After stacking the original image with these perturbations, it is input into the classification module of the attack model to obtain the classification label of the adversarial sample. If the classification label is not consistent with the label of the original sample, the output image indicates that it is an adversarial sample. If the labels agree, the image is sent back to the generator module to continue generating perturbations that are able to fool the classifier.

2.2. FGSM adversarial attacks

In FGSM (Fast Gradient Sign Method) adversarial attack algorithm, by adding network gradient-based perturbation to the original image, Goodfellow *et al.* [8] generated adversarial samples that can make the model misclassify. The perturbation formula is as follows.

$$\ell = \varepsilon \text{sign}\left(\nabla_x J(x, y, \theta)\right) \tag{2}$$

Where x represents the original sample, θ is the weight parameter of the model, ε is the perturbation coefficient, y is the true class of x, *sign* is the sign function, J is the loss function, and ∇_x is the partial derivative of the loss function with respect to x. By calculating the partial derivative of the loss function J with respect to the original sample, the result is input into the sign function, and the direction of the image gradient is determined according to the properties of the sign function. The core idea of the algorithm is that when the disturbance is completely consistent with the gradient direction, the loss function (error value) will increase, which will affect the classification result. The sign function is used to ensure that the perturbation is in the same direction as the gradient. ε is an artificial parameter, similar to a weight parameter, that controls the amount of attack noise. The larger the value of ε is, the more intense the attack is, and the attack noise is also more perceptible to the naked eye. Therefore, choosing an appropriate value of ε is crucial for the success of the attack. In order to test the influence of different ε values, the researchers selected 10,000 sample data, made adversarial samples according to different perturbation coefficients, and input these FGSM adversarial samples into the pre-trained model (ResNet-50), and obtained the test accuracy under different perturbation coefficients, as shown in **Figure 2**.

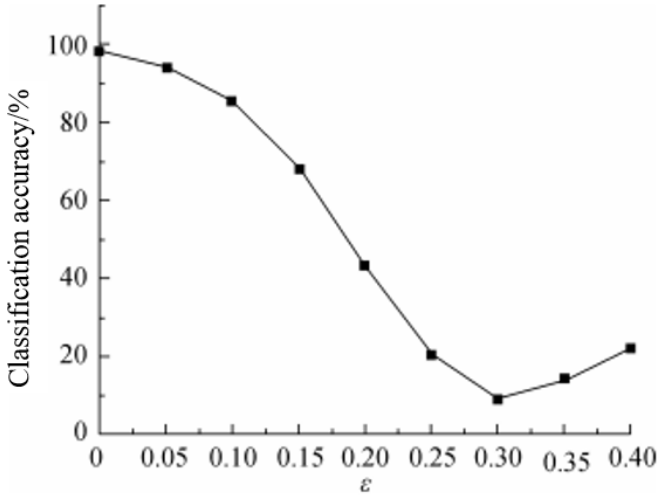

Figure 2. Influence of disturbance factor on accuracy

From the experimental results in **Figure 2**, the classification accuracy of the ResNet50 model on 10,000 sample data is 98.1% without adversarial attacks. With the increase in the degree of disturbance, the classification accuracy of the ResNet50 model for the adversarial samples shows a trend of first decreasing and then increasing. When the ε value is 0.3, the attack success rate of the adversarial samples generated by the FGSM method reaches 90.1%. Therefore, the generated adversarial samples with different perturbation coefficients are visualized, and the results are shown in **Figure 3**. The theoretical basis for choosing the value of ε is: to prevent the loss value from exceeding the set threshold, so as to preserve the characteristics of the original image.

The adversarial attack improves the loss value by increasing the gradient. By adding the calculated gradient direction to the input image, the loss value of the modified image is higher than that of the original image when it passes through the classification network, resulting in the model misclassifying the input image. In order to ensure the success rate of the attack, this study chooses 0.3 as the perturbation coefficient to make adversarial samples, and the generated adversarial results are presented in **Figure 4**.

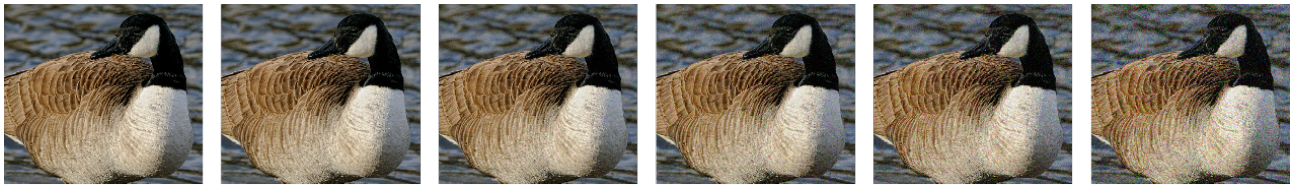

Figure 3. The results generated by different Epsilon

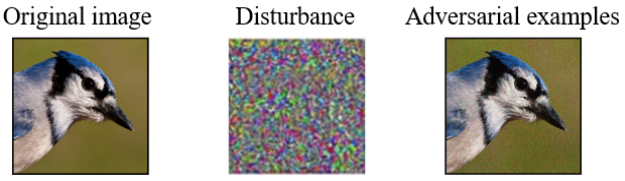

Figure 4. The adversarial samples generated by FGSM and classification results of ResNet50 pre-trained model

2.3. G-ATN adversarial attack

The G-ATN adversarial attack method is similar to the FGSM method in the use of gradient, but the unique feature of G-ATN is that it takes the original image and its gradient as the input to generate the perturbation, so that the perturbation can change according to the characteristics of different input images. In contrast, the perturbation in the FGSM method is mainly determined by the gradient of the network model as well as the chosen perturbation coefficients.

In this study, we modified the original network structure by changing the single-channel input to a three-channel and adding the image gradient as an additional input to generate the perturbation to enhance the influence of image characteristics on the perturbation. The structure of G-ATN is shown in **Figure 5**.

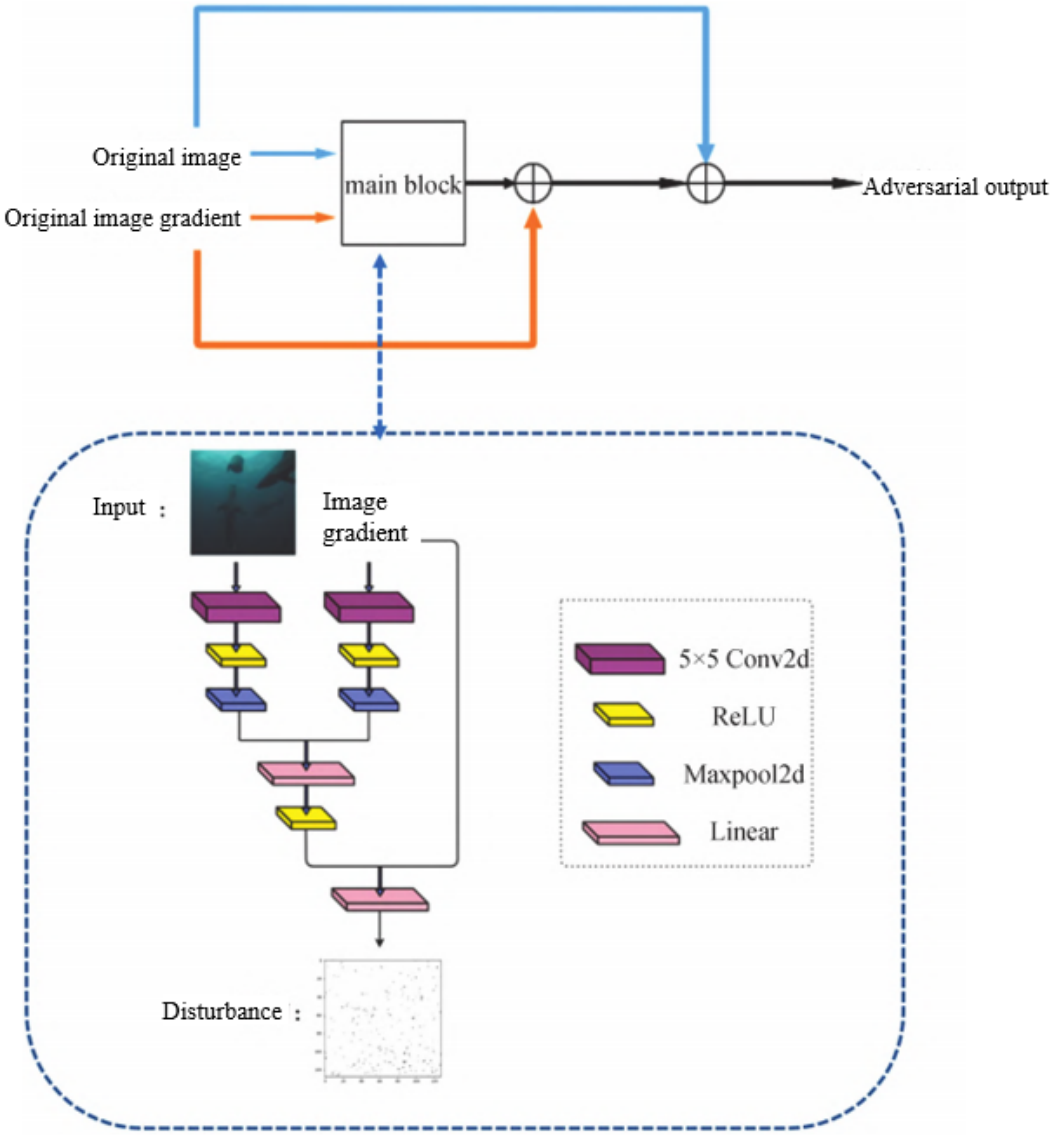

Figure 5. Block diagram of G-ATN

This method takes the original image and image gradient as input and obtains the corresponding disturbance value by generating the main block. Specifically, the generation disturbance module first uses a 5x5 convolutional layer to extract the features of the original image and its gradient. Then, the feature image is divided into blocks through Max pooling, and the maximum value of each feature block is calculated to obtain the pooled image. G-ATN further takes the gradient of the original image as the additional input of the disturbance value, which enhances the influence of the gradient on the disturbance, so that the disturbances generated by different images are different due to their different characteristics and gradients. The generated disturbance is superimposed with the original image to obtain the output, and it is sent to the discriminator to check whether the classification label is consistent with the label of the original image. If not, it is judged as an adversarial sample. The specific perturbation generation results are shown in **Figure 6**.

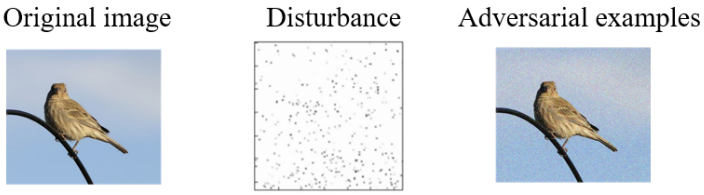

Figure 6. The adversarial samples generated by G-ATN and classification results of ResNet50 pre-trained model

Figure 6 shows that the adversarial perturbations generated by the G-ATN adversarial attack method will present differences depending on the input images. Adding the gradient of the input image as an additional input can improve the success rate of the adversarial attack, but there may be some identifiable noise points in the output image.

3. Filter-based classifiers

3.1. Filter-based classifier network structure

The filter-based classification model consists of two parts: feature extraction and classification. In the process of generating adversarial samples, the pre-trained model of ResNet50 is selected as the target network for the three attack methods. Therefore, the network structure of ResNet50 is used to extract image features in the feature extraction part.

In a regular neural network architecture, the output of each layer is usually directly translated into the input of the next layer. However, in the residual network design, the output of each layer is not only passed to the next layer, but also skipped to the next two or three layers. **Figure 7** shows a detailed example of the structure of the residual network.

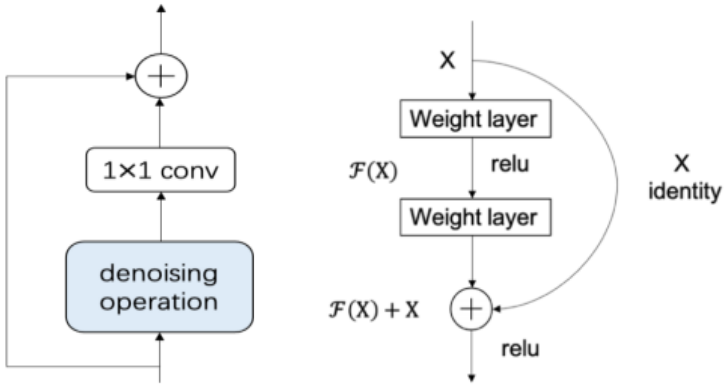

Figure 7. Filter-based classifier network structure

In **Figure 7**, *F(X)* is the residual, and *X* is the input value of the whole residual block. The operation directly leading from the first layer of the residual block to the last two layers of the Network is called a shortcut connection, also known as Highway Network. *X* is input in the residual block, and through the two-layer network, *F(X)* is added to the shortcut connection, and then activated by the ReLU function, and finally the output result of the residual block is obtained. If *H(X)* is expressed as the final output result, *H(X)* can be expressed as *H(X)=F(X)+X*. From this, we can see that *F(X)* is the difference between the final output of the network and the input.

Gaussian filter is used to remove image noise in the denoising module. Gaussian filter can effectively suppress the noise obeying normal distribution. Most of the noise in the image is Gaussian distribution, so the Gaussian filter is widely used in the image field. Another advantage of the Gaussian filter is that it can not only suppress the noise generated by the target detection data set in the process of collection but also blur the generated adversarial samples, which can play a certain inhibitory role in the disturbance added to the samples. For this consideration, the Gaussian filter is used as a filter in the denoising module.

Gaussian filters, also known as Gaussian smoothing operators, are used to "blur" an image and remove details and noise. In this sense, it is similar to the mean filter, but it represents a different kernel in the shape of a Gaussian ("bell-shaped") hump. The principle of the Gaussian filter is that the pixel values of the whole image are weighted, and each pixel value is obtained by the weighted average of the pixel values in its neighborhood.

A Gaussian distribution in one dimension has the following form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

In two dimensions, isotropic (i.e., circular-symmetric) Gaussians have the following form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4)$$

3.2. Loss functions

Distinguishing adversarial examples from non-adversarial examples is a binary classification problem. For binary classification problems, labels [0,1] are used to represent the original and adversarial examples respectively. According to the output of the sigmoid function, the probability that the current sample label is 1 and 0 can be represented. In this paper, the input sample data is represented by the input sample data, and the label of the sample data is represented by y. The probability that sample x is correctly predicted as label y is shown in Equation (5).

$$P(y|x) = \hat{y}^y (1 - \hat{y})^{1-y} \quad (5)$$

Where, $\hat{y} = P(y = 1 \mid x)$ represents the predicted probability of the model when the label is 1; $1\text{-}\hat{y} = P(y = 0 \mid x)$ represents the model prediction probability when the label is 0. Equation (3) shows that when $y = 0$, the prediction probability is $P(y \mid x) = \hat{y}^0 (1 - \hat{y})^{1-0} = 1 \times (1 - \hat{y})^{1-0} = P(y = 0 \mid x)$ and when y =1, the prediction probability is $P(y \mid x) = \hat{y}^1 \times (1 - \hat{y})^{1-1} = P(y = 1 \mid x)$. Therefore, Equation (5) can express the probability that a single sample is an adversarial sample and a non-adversarial sample. However, in the process of model training, in order to make the model better focus on misclassified samples, it is also necessary to make the model pay attention to misclassified sample data. Therefore, this study added an exponential loss function $L(y, f(x)) = e^{-yf(x)}$, where y represents the true label of the sample and $f(x)$ represents the prediction result of the model for the sample. Using this loss function can make the model produce an exponential growth loss value for the misclassified samples. Finally, based on Equation (5), the exponential function is added. The corresponding weights W_1, $W_2$of the two loss functions are given to obtain the loss function of the filter-based classifier, as shown in Equation (6).

$$L = -\{W_1[y\log\hat{y} + (1 - y)\log(1 - \hat{y})] + W_2\exp[-yf(x)]\} \quad (6)$$

3.3. Choosing an optimizer

When building classification models, the choice of optimizer has a significant and direct impact on the performance of the model. The role of the optimizer is to adjust the model parameters to reduce the value of the loss function and improve the accuracy of the model on the training data. Different optimizers adopt their own unique methods to update the weights and biases of the model, which will affect the training efficiency, convergence, and final performance of the model. This study compares the performance of multiple optimizers, including the Adaptive Moment Estimation optimizer (Adam), the hierarchical adaptive moment optimizer (LAMB) suitable for batch training, and the Stochastic gradient Descent (SGD). We conduct comparative experiments on the accuracy of the model using different optimizers while keeping the loss function and learning rate consistent. The experimental results are shown in **Table 1**. According to the data in **Table 1**, when classifying the adversarial samples generated by AdvGAN and FGSM, the classification model using Adam optimizer has high accuracy. However, when classifying the adversarial samples generated by G-ATN, the classification model selected by the LAMB optimizer shows relatively higher accuracy.

Table 1. Accuracy of the model using different optimizers

Optimizer	Adversarial algorithm	Accuracy (%)
Adam	AdvGAN	94.3
	FGSM	90.3
	G-ATN	60.3
LAMB	AdvGAN	74.5
	FGSM	79.6
	G-ATN	91.6
SGD	AdvGAN	79.6
	FGSM	89.2
	G-ATN	80.1

3.4. Model training

Early data processing plays an important role in classification. The processing results directly determine the prediction and generalization ability of the model. This paper carries out comparative experiments and obtains the importance of early data processing through comparison. The original data images were subjected to adversarial attacks according to three adversarial attack algorithms to generate corresponding adversarial samples, and a total of 800 images (without the rest of the processing) were obtained. Based on the input of the filter classification model, the data set was divided into a training set and a test set according to the ratio of 8:2. In the experimental-related parameter settings, the learning rate was adjusted by equal interval, that is, StepLR, the initial value was 1x10-3, and was decreased according to the training rounds. SGD was used as the initial optimizer and the batch size was set to 4. Shuffle is set to True to avoid the influence of data input order on network training. The experimental results are shown in **Figure 8**.

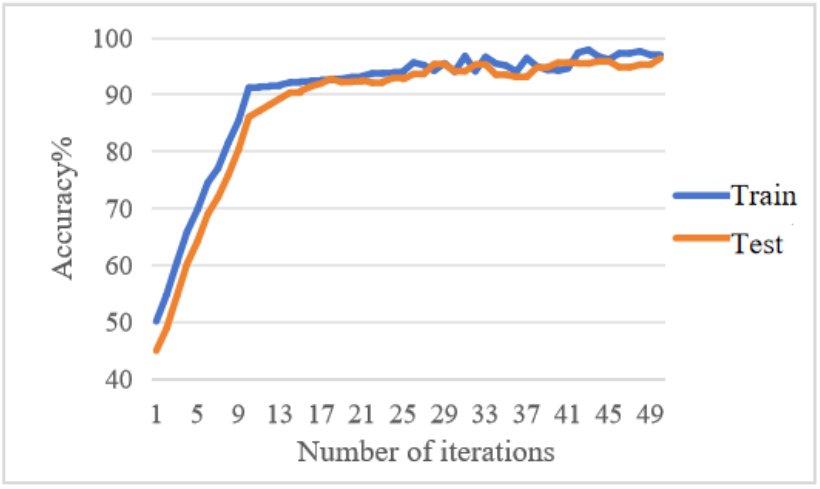

Figure 8. Comparison results of data processing experiments

4. Analysis of experimental results

With the development of Convolutional neural Networks (ConvNets), EfficientNet came out in 2019, which proposed a more general idea for the optimization of current classification networks. The compound model scaling algorithm is proposed to improve the index by comprehensively optimizing the network depth, network width, and resolution [17]. EfficientNet achieves 84.3% Top-1 on the ImageNetILSVRC2012 dataset Accuracy [18]. Therefore, three kinds of adversarial samples and non-adversarial attack samples (related noise samples and original samples) generated by AdvGAN, FGSM, and G-ATN adversarial attack methods are sent into the EfficientNetb7_ns and filter-based classification models respectively, and the accuracy of model classification is obtained, as shown in **Table 2**.

Table 2. EfficientNet classification results

Adversarial Attack Algorithm	Number of adversarial examples	Number of non-adversarial examples	The model correctly identifies the quantity	Model recognition accuracy
AdvGAN	7800	1800	1797	18.72%
FGSM	7800	1800	3401	35.43%
G-ATN	7800	1800	2818	29.36%

According to the data in the table, the adversarial samples we made successfully caused confusion to the EfficientNet classification model, and more than 60% of the adversarial samples generated by the three adversarial attack algorithms were not recognized by EfficientNet. Comparing the model recognition accuracy in **Table 3** and **Table 2**, it can be seen that the filter-based classification model adopted in this paper achieves 89.45%, 93.72%, and 80.97% in recognition accuracy respectively, which exceeds the accuracy of the EfficientNet classification network in identifying adversarial samples and non-adversarial samples. Therefore, the filter-based classification model adopted in this paper shows a strong ability to distinguish adversarial samples from non-adversarial samples.

Table 3. Results based on the filter-based classification

Adversarial Attack Algorithm	Number of adversarial examples (number of classes)	Number of non-adversarial examples (number of classes)	Correctly identify the number of adversarial samples	Correctly identify the original number of samples	Model recognition accuracy
AdvGAN	7800 (13 classes)	1800 (13 classes)	6977	1610	89.45%
FGSM	7800 (13 classes)	1800 (13 classes)	7310	1687	93.72%
G-ATN	7800 (13 classes)	1800 (13 classes)	6315	1457	80.97%

5. Conclusion and prospects

In order to accurately identify adversarial samples in image classification, this study used a filter-based classification model, which is based on a residual network architecture. Adversarial samples are created by adding subtle perturbations to the original data. These samples are able to mislead deep neural networks to produce erroneous outputs with a high degree of confidence, posing a challenge to network security. The model consists of two main parts: residual network feature extraction and classification module. The model uses an adversarial training method for cyclic training and evaluates a variety of popular adversarial attack techniques on the Mini-ImageNet dataset. Experimental data show that the filter-based classifier performs significantly better than the EfficientNet model in identifying adversarial samples, highlighting the excellent performance of RC-Net in distinguishing adversarial and non-adversarial samples.

In addition, this study provides strategies to reduce the negative impact of adversarial samples on cybersecurity in the field of image recognition, which makes it possible for security-sensitive applications such as autonomous driving, financial fraud detection, facial recognition, and malware detection. This paper emphasizes the importance of constructing a robust model for adversarial attacks, which lays a foundation for future research on tracing the source of adversarial samples, and further improves the security and reliability of artificial intelligence systems in practical applications.

Disclosure statement

The author declares no conflict of interest.

References

[1] Li C, Cao YN, Peng YK, 2022, Research on Automatic Driving Target Detection Based on YOLOv5s. Journal of Physics: Conference Series, 2171(1): 012047.

[2] Sainath TN, He YZ, Li B, et al., 2020, A Streaming on Device End-to-End Model Surpassing Server-Side Conventional Model Quality and Latency, International Conference on Acoustics, Speech and Signal Processing, Barcelona, 6059–6063.

[3] Wang Q, 2021, Research on Image Recognition Technology Based on Convolution Neural Network, International Conference on Information Systems and Computer Aided Education, Dalian, 2628–2632.

[4] Chen JF, Zhang WH, 2020, Research and Application of Deep Learning in Image Recognition. Electronics World, 2020(19): 48–49.

[5] Krizhevsky A, Sutskever Hinton GE, 2017, ImageNet Classification with Deep Convolutional Neural Networks. J

Communications of the ACM, 60(6): 84–90.

[6] Chen QY, Huang Y, Sun R, et al., 2020, An Efficient Accelerator for Multiple Convolutions from the Sparsity Perspective. IEEE Transactions on Very Large Scale Integration (VLS) Systems, 28(6): 1540–1544.

[7] Chen X, Weng J, Deng XL, et al., 2023, Feature Distillation in Deep Attention Network Against Adversarial Examples. IEEE Transactions on Neural Networks and Learning Systems, 34(7): 3691–3705.

[8] Goodfellow IJ, Shlens J, Szegedy C, 2023, Explaining and Harnessing Adversarial Examples. arXiv. https://doi.org/10.48550/arXiv.1412.6572

[9] Zantedeschi V, Nicolae MI, Rawat A, 2017, Efficient Defenses Against Adversarial Attacks. arXiv. https://doi.org/10.48550/arXiv.1707.06728

[10] Li X, Li FX, 2017, Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics, International Conference on Computer Vision, Venice, 5775–5783.

[11] Zhang CL, Ye ZC, Wang Y, et al., 2018, Detecting Adversarial Perturbations with Saliency, International Conference on Signal and Image Processing, Shenzhen, 271–275.

[12] Lu JJ, Issaranon T, Forsyth D, 2017, SafetyNet: Detecting and Rejecting Adversarial Examples Robustly, International Conference on Computer Vision, Venice, 446–454.

[13] Linardos P, Little S, McGuinness K, 2019, MediaEval 2019: Concealed FGSM Perturbations for Privacy Preservation. arXiv. https://doi.org/10.48550/arXiv.1910.11603

[14] Moosavidezfoolis M, Fawzi A, Frossard P, 2023, DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. arXiv. https://doi.org/10.48550/arXiv.1511.04599

[15] McCaffrey J, 2014, Test Run: Distorting the MNIST Image Data Set. MSDN Magazine, 29(7): 66–69.

[16] Wu DX, Xu J, Liu H, 2020, Analysis of the Influence of Stylized-CIFAR10 Dataset on ResNet, in Chen XF, Yan HY, Yan QB, et al., Machine learning for Cybersecurity, Springer, Cham, 416–426.

[17] Mangla P, Jandial S, Varshney S, et al., 2023, AdvGAN+: Harnessing Latent Layers for Adversary Generation. arXiv. https://doi.org/10.48550/arXiv.1908.00706

[18] Jabra MB, Koubaa A, Benidira B, et al., 2021, COVID-19 Diagnosis in Chest X-Rays Using Deep Learning and Majority Voting. Applied Sciences, 11(6): 2884.