Plant Disease Classification and Adversarial Attack based CL-CondenseNetV2 and WT-MI-FGSM

Yong Li[1], Yufang Lu[2]*
School of Information Science and Engineering, Guilin University of Technology
Guilin, China

***Abstract*—In recent years, deep learning has been increasingly used to the detection of pests and diseases. Unfortunately, deep neural networks are particularly vulnerable when attacked by adversarial examples. Hence it is vital to explore the creation of intensely aggressive adversarial examples to increase neural network robustness. This paper proposes a wavelet transform and histogram equalization-based adversarial attack algorithm: WT-MI-FGSM. In order to verify the performance of the WT-MI-FGSM, we propose a plant pests and diseases identification method based on the coordinate attention mechanism and CondenseNetV2: CL-CondenseNetV2. The accuracy of CL- CondenseNetV2 on the PlantVillage dataset is 99.45%, which indicates that the improved CondenseNetV2 model has a more significant classification performance. In adversarial sample experiments using WT-MI-FGSM and CL-CondenseNetV2, experimental results show that when CL-CondenseNetV2 is attacked by the adversarial algorithm WT-MI-FGSM, the error rate reaches 89.8%, with a higher attack success rate than existing adversarial attack algorithms. In addition, the accuracy of CL-CondenseNetV2 is improved to 99.71% by adding the adversarial samples generated by WT-MI-FGSM to the training set and performing adversarial training. The experiments demonstrate that the adversarial examples caused by WT-MI-FGSM can improve the model's performance.**

***Keywords*—Adversarial examples; FGSM; plants diseases and pests; attention mechanism; CondenseNetV2**

I. Introduction

A country's agricultural sector is vital to its economic growth, with the potential to both stimulate and directly impact the national economy's development or stagnation. The stability of agriculture is intrinsically linked to social stability and national self-sufficiency; therefore, it is crucial to achieve the steady and sustainable development of agriculture. To achieve sustainable development in the agricultural field, we must first perform well in the prevention and control of crop diseases and insect pests, guarantee that the prevention and control measures are scientific and safe, successfully control diseases and insect pests, and promote eco-logical development in the agricultural field in a benevolent and sustainable direction [1].

Since the advent of deep learning, deep learning-based picture identification has been a popular topic in the image recognition community, finding widespread application in areas such as facial recognition, transportation, and healthcare. The most traditional network models are GoogLeNet, VggNet, and ResNet [2]-[4]. In recent years, the agricultural sector has also made extensive use of deep learning. Progress has been made in the identification of plant diseases and pests as a result of the extensive research undertaken by scientists. Based on the AlexNet network, LV and others use batch normalization, PRelu activation function, etc. to improve network convergence and avoid over-fitting. They also combine extended convolution and multi-scale convolution to improve network feature extraction ability, demonstrating that the algorithm of feature enhancement can effectively improve the network's feature extraction ability and recognition accuracy [5]. Pandian et al. utilized image enhancement technology based on image processing and deep learning to improve the crop disease data set. Additionally, they expanded and improved the data set with the antagonistic generating network and neural pattern transfer using migration learning technology. Using this method, the experimental findings show that the improved data set can reach higher precision [6]. Durmus used tomato photos from the PlantVillage dataset to train numerous deep neural networks, and the accuracy of networks such as SqueezeNet significantly improved due to this [7-8]. Using plant images in the visible spectrum, Lily proposed a straightforward and reliable method for diagnosing plant diseases [9]. In his research work, Kaur proposed the DAG-ResNet model and utilized it to discover a number of tomato illnesses [10]. The accuracy was 98.8%. ALVARE et al. integrate FasterR-CNN, SSD, RFCN, VggNet, ResNet, and other feature extractors to obtain a notable recognition and classification effect [11]. Wenliang Tang used conditional convolution, channel attention module, and knowledge distillation to improve the model [12]. The accuracy was 97.6%. Agriculture diseases and pests have a high degree of similarity, a more dispersed and intricate distribution, a greater difficulty in classification, and a greater need for classification networks. Based on the classic CondenseNetV2 model, this work introduces CL- CondenseNetV2, a method for identifying agriculture diseases and pests that combines the coordinate attention mechanism and CondenseNetV2. The model incorporates a flexible and lightweight coordinate attention mechanism and embeds the position information into the channel attention in order to detect and identify the target area with greater precision. As a result of these enhancements, the CL- CondenseNetV2 model now has an identification accuracy of 99.45%, allowing for highly precise detection of agricultural diseases and pests.

Although deep neural networks perform well in most classification tasks, they are vulnerable when faced with adversarial samples. Adversarial samples are a class of samples formed by intentionally adding subtle perturbations to a dataset, which can induce the network model to misclassify

*Corresponding Author

and threaten the model's safety. However, on the other hand, for model designers, adversarial samples can be used as an effective tool to evaluate the security and robustness of the model. They can effectively improve the correctness and security of the model classification through adversarial training. Adversarial attack algorithms can be classified into two categories according to the mainstream classification methods: black-box attack and white-box attack, and white-box attack algorithms include FGSM, DeepFool, C&W, etc. [13]-[15]. Black-box attack algorithms currently have single-pixel and local search attack algorithms, etc. [16].

However, there are fewer examples of improving the classification success of the model by adding adversarial examples for adversarial training. This paper proposes a new DNN called CL-CondenseNetV2 that adds a coordinate attention module to CondenseNetV2. A significant number of comparative studies have shown that our network model performs well. In addition, this paper introduces wavelet variation and histogram equalization in the image domain based on the MI-FGSM algorithm to propose a new adversarial attack algorithm WT-MI-FGSM. The adversarial examples generated by this algorithm can be used to train CL-CondenseNetV2. Training with adversarial examples will help us to improve the accuracy of the classification of plant diseases and the security and robustness of the model.

II. Adversarial Attack Algorithm

FGSM, a white-box attack technique built on the production of adversarial example gradients, is the most widely used adversarial assault algorithm today. By iteratively computing the gradient, Alexey et al introduced I-FGSM, which significantly enhances the fit of the adversarial sample to the model [17]. After introducing the momentum factor based on I-FGSM, Dong et al suggested the MI-FGSM approach, which greatly increased the success rate of black-box assaults and successfully enhanced the migrability of the generated adversarial samples [18]. The MI-FGSM algorithm function is shown in (1) and (2).

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(x_t^{adv}, y)}{\left\| \nabla_{x_t^{adv}} J(x_t^{adv}, y) \right\|_1} \tag{1}$$

$$x_{t+1}^{adv} = Clip_x^{\varepsilon} x_t^{adv} + \alpha \cdot sign(g_{t+1}) \tag{2}$$

In the formula, *sign*() is a symbolic function, and $\nabla_x J(x, y)$ represents a gradient. ε is the size of the neighborhood. x_{t+1}^{adv} is the adversarial example generated by iterating $t+1$ times. t represents the number of iterations. The step length can be obtained by $\alpha = \varepsilon / T$ .It ensures that the adversarial examples generated are in the neighborhood of x .Where μ is the decay factor of the momentum term, and g_{t+1} denotes the cumulative gradient iteration $t+1$ times. The role of the *Clip* function is to constrain the adversarial examples within the ε-neighborhood of the original image x to satisfy the Infinite norm constraint.

In this study, wavelet transform and histogram equalization are successfully integrated with MI-FGSM to create the WT-MI-FGSM, a more effective adversarial attack algorithm. The overall flow of WT-MI-FGSM is shown in Fig. 1 below. By means of an adversarial attack algorithm, the original example is utilized to produce an adversarial example. First, the wavelet transform and histogram equalization are performed on the origin example to obtain a 224×224×3 image. The loss function is then computed using the acquired images as input into the model. Iteratively updating the example along the gradient of the loss function is followed by the addition of perturbations. If the requirements are unmet, the iteration will continue until it succeeds. Finally, output confrontational examples. The WT-MI-FGSM algorithm function is shown in (3). Where D is the image enhancement function. It includes wavelet transform and histogram equalization.

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(D(x_t^{adv}), y)}{\left\| \nabla_{x_t^{adv}} J(D(x_t^{adv}), y) \right\|_1} \tag{3}$$

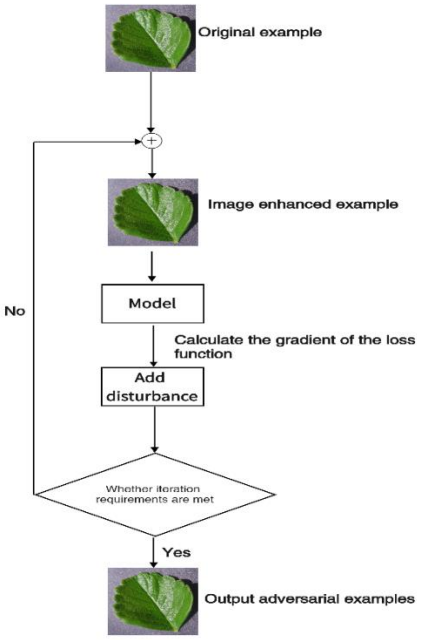

Fig. 1. WT-MI-FGSM attack concrete steps.

III. CL-CondenseNetV2 Model Design

A. CondenseNetV2

Huang Gao's team proposed DenseNet and CondenseNet in 2017, and DenseNet establishes a dense connection mechanism that allows each layer in the network to be directly connected to its preceding layer in the same block to achieve feature reuse, which enables DenseNet to reduce the total number of parameters and improve efficiency significantly [19]. CondenseNet introduces a full-dense connection and pruning mechanism based on DenseNet [20]. The full-dense connection enables the network to establish a dense connection between different blocks while combining with average pooling to achieve stitching between feature maps of various sizes, thus gaining more robust feature reuse. CondenseNet can achieve

almost the same accuracy as DenseNet with just 1/10 the training time thanks to the pruning mechanism, which enables the network to prune the irrelevant weights during the training phase and reduce network redundancy.

Nevertheless, features in DenseNet and CondenseNet will remain the same once they are formed, drastically ignoring the potential value of some features. CondenseNetV2, a powerful yet lightweight neural network based on CondenseNet, was suggested by Gao Huang in 2021 [21]. CondenseNetV2 introduces a sparse feature reactivation mechanism that enables the network to learn to choose a few potentially redundant features. The efficiency of the deep network's feature reuse is increased by concurrently cropping and updating these redundant features to make them better suited to feature learning. CondenseNetV2 achieves better performance than DenseNet and CondenseNet at a low computational cost, and achieves excellent performance on image classification and detection tasks. Table I shows the CondenseNetV2 network structure.

TABLE I. CONDENSENETV2 NETWORK STRUCTURE

Layers	Input	ConenseNetV2
Convolution	224×224	3×3 *Conv* , stride 2
Dense Block (1)	112×112	$\begin{bmatrix} 1\times1L-Conv \\ 3\times3G-Conv \\ SFR\ module \end{bmatrix}\times4$ (k=8)
Transition Layer (1)	112×112	2×2 average pool, stride2
Dense Block (2)	56×56	$\begin{bmatrix} 1\times1L-Conv \\ 3\times3G-Conv \\ SFR\ module \end{bmatrix}\times6$ (k=16)
Transition Layer (2)	56×56	2×2 average pool, stride2
Dense Block (3)	28×28	$\begin{bmatrix} 1\times1L-Conv \\ 3\times3G-Conv \\ SFR\ module \end{bmatrix}\times8$ (k=32)
Transition Layer (3)	28×28	2×2 average pool, stride2
Dense Block (4)	14×14	$\begin{bmatrix} 1\times1L-Conv \\ 3\times3G-Conv \\ SFR\ module \end{bmatrix}\times10$ (k=64)
Transition Layer (4)	14×14	2×2 average pool, stride2
Dense Block (5)	7×7	$\begin{bmatrix} 1\times1L-Conv \\ 3\times3G-Conv \\ SFR\ module \end{bmatrix}\times8$ (k=128)
Classification Layer	1×1	7×7 global average pool 1000D fully-connected, SoftMax

B. Introduction of Coordinate Attention Mechanism

Attention Mechanism is a unique structure within a machine learning model that is used to automatically calculate and learn the contribution of input data to output data. Common modules for attention mechanisms include SE, CBAM, etc. SE is only concerned with the weighting of channels [22]. Despite the fact that CBAM simultaneously considers the weight allocation of channels and spaces, redundant convolution pooling operations result in the loss of some useful information [23]. Coordinate Attention (CA) mechanism is an attention mechanism that can embed position information into channel attention introduced in this paper [24]. In comparison to SE, CBAM, and other attention mechanisms, this attention mechanism is not only capable of capturing cross-channel information, but also direction perception and position perception information, in addition to being lightweight and flexible. Coordinate attention operations consist of coordinate information embedding and coordinate attention generation. Coordinate attention module is shown in Fig. 2.

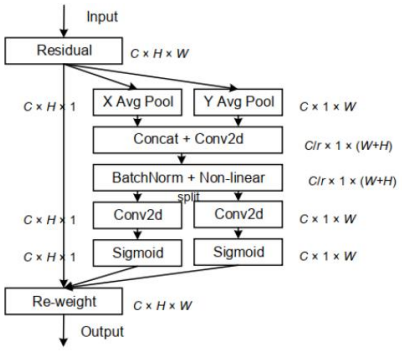

Fig. 2. Coordinate attention module.

1) Coordinate information embedding: Channel attention frequently employs two-dimensional global pooling to encode global spatial information, but this operation typically makes it challenging to save target location data. In order to prevent this, the Coordinate attention mechanism decomposes the two-dimensional pooling of channel attention into two one-dimensional feature coding processes and collects features along two spatial directions. Equation (4) and formula (5) represent the horizontal and vertical coordinate coding operations of input x.

$$z_c^h(h)=\frac{1}{W}\sum_{0\le i\le W}x_c(h,i) \tag{4}$$

$$z_c^w(w)=\frac{1}{H}\sum_{0\le i\le H}x_c(j,w) \tag{5}$$

Where x_c denotes the c th channel component of the input data, h and w respectively reflect the data's height and width. These two coding operations allow the attention module to capture the long-term dependency along one spatial direction and to store precise location information along the other spatial direction, thereby assisting the network in more precisely identifying the target information of pests and diseases on crops.

2) Coordinate attention generation: In order to make full use of the global receptive field obtained through the embedding operation of coordinate information and encode the accurate position information, first embed the coordinate information into the obtained features for concatenate operation, and then send them into the convolution module with the shared convolution core of 1×1, reduce its dimension to the original C/r, and then send the feature map F_1 after

batch normalization into the nonlinear activation function to obtain the feature map f in the form of $1\times(W+H)\times C/r$. Equation (6) is shown below.

$$f=\delta\left(F_1\left(\left[z^h,z^w\right]\right)\right) \tag{6}$$

Where $[\cdot,\cdot]$ is the splicing operation along the spatial dimension, δ is the nonlinear activation function, F_1 is the convolution change function, and f is the intermediate feature map that encodes the spatial information in the horizontal and vertical directions.

After the above operations, f is decomposed into f^h and f^w along the spatial dimension, and 1×1 convolution and nonlinear activation operations are performed on them to obtain the attention weights g^h and g^w of the feature map in the horizontal and vertical coordinate directions respectively. Equation (7) and (8) are as follows.

$$g^h=\sigma\left(F_h\left(f^h\right)\right) \tag{7}$$

$$g^w=\sigma\left(F_w\left(f^w\right)\right) \tag{8}$$

Where σ is the sigmoid activation function, F^h and F^w represent the convolution change function of the characteristic components f^h and f^w, respectively.

Finally, expand g^h and g^w, calculate the coordinate attention mask by matrix multiplication, and act on the input to get the output Y of the attention module:

$$y_c(i,j)=x_c(i,j)\times g_c^h(i)\times g_c^w(j) \tag{9}$$

C. CL-CondenseNetV2

The above coordinate attention is added to the CondenseNetV2 network to obtain the basic structure of CL-CondenseNetV2 as shown in Fig. 3. In each layer of the proposed network, LGC is first used to select important features for feature learning, and after obtaining new features, the SFR module is used to reactivate the previous features. On this basis, we added the coordinate attention module. The coordinate attention module improves the recognition accuracy of the model by making the network model lightweight while enabling the model to locate and identify the target region more accurately. Since plant disease features are distributed in different positions on the front of leaves, the classification network needs to accurately pay attention to the spatial location of disease features. Therefore, coordinate attention can significantly improve the recognition accuracy of plant diseases

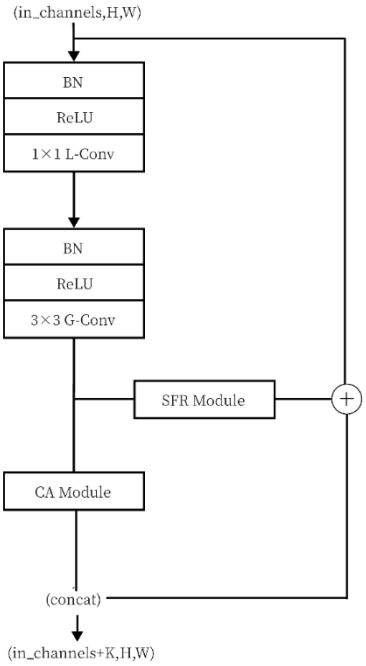

Fig. 3. CL-CondenseNetV2 structure.

IV. Preparation for the Experiment

A. Experimental Environment

The experiment's environment setting is as follows: The graphics card is an NVIDIA GeForce RTX 3060, and the Windows 64-bit system CPU is an 8-core, 16-thread AMD Rayon R7-5800H processor. The memory is DDR4 16G, and the hard drive storage is 512G SSD. The software environment consists of Anaconda 4.10.3 and CUDA 11.6. The model is built and trained using the Python programming language and PyTorch framework.

B. Parameter Setting

In the model training parameter settings, the training batch size is set to 16, the test batch size to 8, and the number of iterations to 50 rounds (Epochs). The employed optimizer is SGD (Stochastic Gradient Descent) [25]. The rate of learning is set at 0.001. Adopting the learning rate exponential decay approach. Gamma has been adjusted to 0.9. Loss is represented by the SoftMax cross-entropy loss function. The definition of the function is:

$$L=-\sum_{k=1}^{n}\sum_{i=1}^{C}t_{ki}\lg y_{ki} \tag{10}$$

Where, n is the pixel of the picture; t_{ki} is the probability that pixel k belongs to the category i; y_{ki} is the probability of predicting the pixel k as the category i for the classification network.

C. Data Set

The data set utilized in this experiment is PlantVillage, which consists of tens of thousands of photos of healthy and diseased plants annotated by plant pathologists and is available for free download at www.PlantVillage.org. All photographs in the PlantVillage database were captured at experimental research stations affiliated with American institutions (Pennsylvania, Florida, Cornell, etc.). The data set consists of 54303 health and disease images split into 38 categories, and it is still growing [26]. The image and caption of several plants pest data sets are depicted in Fig. 4.

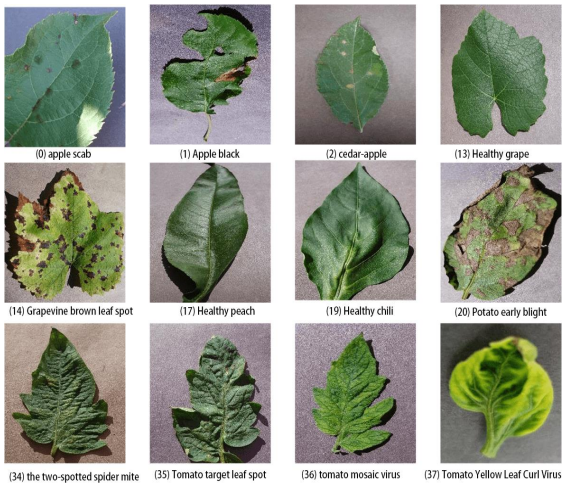

Fig. 4. Examples of images and labels of some crop pest data sets.

D. Data Preprocessing

By using a data enhancement approach, this work also increases the dataset. First, the data image is separated into a training set and a test set with an 8:2 ratio, and then the training set's data image is expanded to 81454 images to serve as the expanded training set. The resolution of the data image is finally rebuilt to 256×256 resolution, and the image is then standardized using the mean and standard deviation. Not only can data preparation imitate the actual agricultural setting and increase the diversity of training samples, but it can also improve the model's robustness and prevent overfitting.

E. Transfer Learning

In the realm of artificial intelligence, there exists a method known as "transfer learning". First, acquire knowledge in the source domain, and then apply it to the target domain, so that the target domain can achieve superior learning outcomes [27]. We can use this technique to reduce the number of training samples required by the model, eliminate the time-consuming and inefficient "ab initio" training process, accelerate network model training, and improve their overall learning efficiency. It has found widespread application in the field of image classification.

The improved model employs the model fine-tuned transfer learning method, employs the CondenseNetV2 pre-training model trained on the ImageNet large open dataset, and combines the transfer learning fine-tuning method to apply its parameters to the CL-CondenseNetV2 model, and uses it to identify plants diseases and pests.

F. Performance Metrics

For each of the experiments examined in this study, the evaluation metrics Accuracy, Precision, Recall, and Specificity are used to assess how well the network performed in identifying the test pictures. The evaluation metrics are calculated using the following equation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

Where TP is the true case, FP is the false positive case, FN is the false negative case, and TN is the true negative case. Accuracy is the percentage of samples that the model properly recognizes and categorizes to the total samples, which is often used to evaluate the overall accuracy of the model. Precision is the ratio of true cases to the number of positive cases classified by the model; Recall is the ratio of true cases to all positive cases; Specificity is the ratio of true negative cases to all negative cases.

V. Preparation for the Experiment of Classification

A. Experiment of Classification

1) Comparison of experimental effects between CL-CondenseNetV2 and CondenseNetV2: The original CondenseNetV2 and CL-CondenseNetV2 are trained on the extended dataset to evaluate how well the improved approach of this model works. Fig. 5 and 6 depict a comparison of the accuracy curve and loss value curve of the original network and the modified network, while Table II depicts a comparison of the accuracy and loss value results.

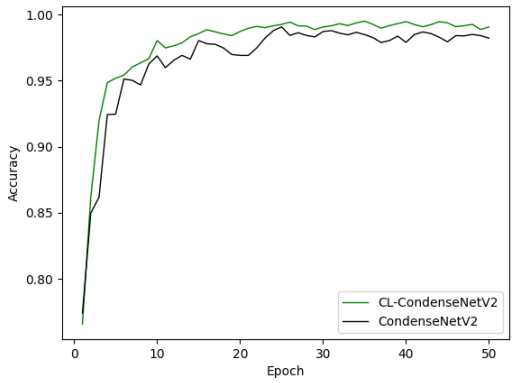

Fig. 5. Comparison of accuracy curves between CL- CondenseNetV2 and CondenseNetV2.

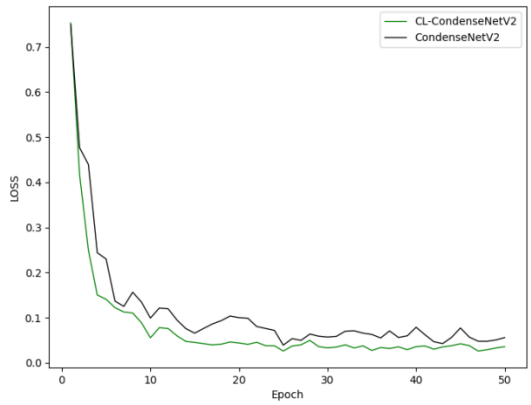

Fig. 6. Comparison of loss curve between CondenseNetV2 and CL-CondenseNetV2.

TABLE II. ACCURACY AND LOSS OF CL- CONDENSENETV2 AND CONDENSENETV2

Algorithms	Acc (%)	Pre (%)	Recall (%)	Spe (%)	Params (M)	Loss
CL-CondenseNetV2	99.45	98.85	98.17	99.89	6.1	0.0259
CondenseNetV2	99.07	96.69	97.03	99.22	6.1	0.0389

As can be shown in Fig. 5 and 6, the CL-CondenseNetV2 suggested in this study is superior than the CondenseNetV2 network model when it comes to the classification and recognition of pictures, as well as the recognition of crop diseases and insect pests. The accuracy and speed of CL-CondenseNetV2's accuracy convergence are faster than those of the original network, and the loss value curve is more consistently steady. According to Table I, the recognition rate of the original network is 99.07 percent, whereas the upgraded network model enhances the recognition rate of crop diseases and pests by 0.38 percent, demonstrating the viability of the improved model CL-CondenseNetV2.

2) Comparison of experimental effects between CL-CondenseNetV2 and other models: To further validate the benefits of the CL-CondenseNetV2 network model, employ three traditional networks, Vgg16, ResNet18, and ResNet50, to train on the improved data set and conduct comparative experiments with CL-CondenseNetV2 in the same experimental environment, using the same training parameters and training timeframes. Fig. 7 and 8 depict a comparison between the accuracy curve and the loss value curve.

Fig. 7 demonstrates that the CL-CondenseNetV2 network model maintains certain advantages. CL-CondenseNetV2's accuracy curve converges more rapidly during training, is more stable, and can essentially maintain its accuracy advantage over Vgg16, ResNet18, and ResNet50.

Fig. 8 shows that the CondenseNetV2 loss value curve has fallen greatly and rapidly, and that the loss value curve is more steady than those of Vgg16, ResNet18, and ResNet50, indicating that the network is more robust. This demonstrates that the CL-CondenseNetV2 network model is preferable. Table III displays a comparison of the experimental outcomes of each model.

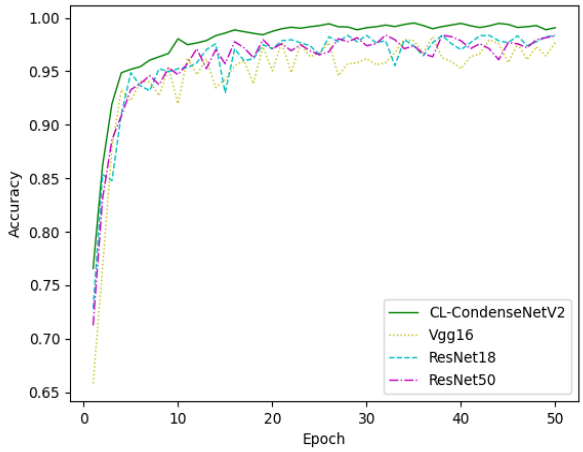

Fig. 7. Comparison of accuracy curves of each model.

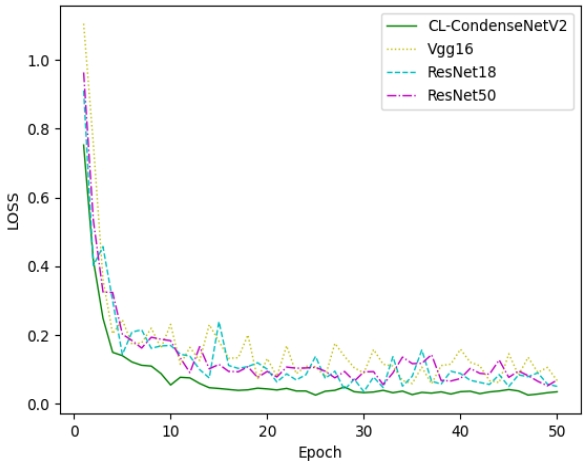

Fig. 8. Comparison of loss curve of each model.

TABLE III. ACCURACY AND LOSS OF EACH MODEL

Algorithms	Acc (%)	Pre (%)	Recall (%)	Spe (%)	Params (M)	Loss
CL-CondenseNetV2	99.45	98.85	98.17	99.89	6.1	0.0259
ResNet18	98.35	96.48	93.57	99.38	11.2	0.0573
ResNet50	98.37	95.37	95.52	99.85	23.5	0.0527
Vgg16	98.21	96.18	96.65	99.34	134.4	0.0596

Table III shows that under the same experimental conditions, CL- CondenseNetV2 achieved the highest accuracy of 99.45, the lowest loss value of 0.0259, and the least parameters of 6.1M, which were superior to Vgg16, ResNet18, and ResNet50. The accuracy of CL-CondenseNetV2 is 1.08 percentage points higher than the highest of the remaining deep learning models, ResNet50，while the number of parameters is 17.4M less than that of ResNet50. When compared to Vgg16, whose parameters is as high as 134.4M, the reductions are even more significant. CL-CondenseNetV2 outperforms other methods when taking into consideration both the network model's parameters and the recognition accuracy, indicating that the enhanced model may be put to better use in the detection of plant diseases and pests.

B. Experiment of Adversarial Example

1) Preparation for the experiment: To fairly validate the performance of the WT-MI-FGSM proposed in this paper, the experimental preparation for the adversarial attack experiments is approximately the same as when a model such as CL-CondenseNetV2 is trained. The experiments use the PlantVillage dataset. This experiment utilized smaller perturbations to make them more difficult to identify rather than setting the hyperparameters as the norm in the momentum method. Using small perturbations can increase the attack algorithm's success rate when compared to other adversarial attack methods. The maximum perturbation ε = 0.3; the number of iterations T = 10; the step size α = 0.03; and the fading factor μ = 1.0. Controlled studies employing the white-box attack techniques of I-FGSM, MI-FGSM, and FGSM, respectively, were also carried out to further confirm the efficacy of the performance of WT-MI-FGSM.

2) Experimental results: This experiment contrasts four white-box adversarial attack algorithms—WT-MI-FGSM, I-FGSM, MI-FGSM, and FGSM—attack Vgg16, ResNet50, ResNet18, CondenseNetV2, and CL- CondenseNetV2, and generates both adversarial and original cases, as illustrated in Fig. 9. According to the experimental findings, all five models are susceptible to adversarial assaults, and all four of these attacks have a high success rate. The attacked models' recognition accuracy was lowered by roughly 91%. It is important to note that the differences between the original instances and the adversarial examples produced by the WT-MI-FGSM given in this research are negligible and challenging for the human eye to detect. The adversarial examples and original examples generated by the experiment are shown in Fig. 9.

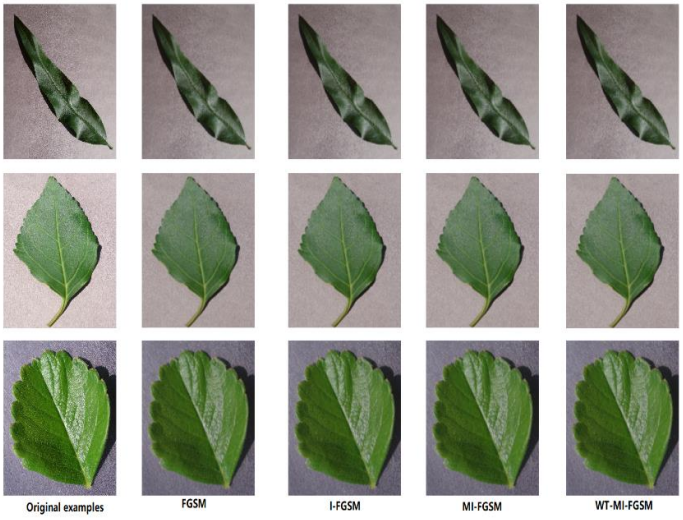

Fig. 9. Adversarial examples generated by various algorithms.

The recognition rate of each model reduces dramatically when attacked by the adversarial attack algorithm, as seen in Table IV. Vgg16 has the fastest drop when attacked by the WT-MI-FGSM algorithm proposed in this research, with a recognition rate of 0.6%, followed by ResNet18 and ResNet50, with a recognition rate of 2.8% and 3.1%. And closely followed by CondenseNetV2, with a recognition rate of 5.3%. CL-CondenseNetV2 still has the greatest recognition rate after receiving the attack, with 10.2%. In addition, Table IV compares the performance of WT-MI-FGSM with other traditional adversarial attack algorithms. The success percentages of FGSM, I-FGSM, and MI-FGSM after CL-CondenseNetV2 received assaults from each algorithm are 85.9%, 86.5%, and 87.3%, respectively. They are all less than 89.8% of WT-MI-FGSM. WT-MI-FGSM outperforms MI-FGSM, which has the highest attack success rate among classic attack algorithms, by 1.9%. It is observed that the proposed adversarial attack algorithm WT-MI-FGSM has the best performance in this paper.

TABLE IV. ACCURACY AND LOSS OF EACH MODEL

Algorithms	ResNet 18	ResNet 50	Vgg16	CondenseNet V2	CL-CondenseNet V2
FGSM	88.5%	88.2%	92.1 %	86.1%	85.9%
I-FGSM	89.8%	89.3%	92.4 %	87.4%	86.5%
MI-FGSM	91.2%	90.1%	93.2 %	89.1%	87.3%
WT-MI-FGSM	97.2%	96.9%	99.4 %	94.7%	89.8%

Adversarial examples can increase model accuracy and robustness. The adversarial examples generated by WT-MI-FGSM are added to the training set, and the individual models are adversarial trained. Table V shows the accuracy of each model after training. CL-CondenseNetV2 has an accuracy of 99.71%, which is 0.26% greater than without adversarial training. Other models' accuracy has also increased. The experimental results show that adversarial examples generated by WT-MI-FGSM can improve model performance.

TABLE V. THE EXPERIMENT OF ADVERSARIAL TRAINING

Models	mAP (%)	mAP(%) (Slow)	mAP(%) (Medium)	mAP(%)(Fast)
CL-CondenseNetV2	99.71	99.07	99.97	99.57
CondenseNetV2	99.38	97.29	99.88	97.93
ResNet18	98.69	93.76	99.67	96.79
ResNet50	98.73	95.75	99.85	95.78
Vgg16	98.63	94.56	99.57	96.85

C. Discussion

To further verify that the CL- CondenseNetV2 model which is added to adversarial examples has a higher recognition rate of plant diseases and pests, it is compared with the DAG-ResNet model in literature [10] and the CondConvSENet detection model in literature [12]. The experimental results are shown in the Table VI.

TABLE VI. EXPERIMENTAL RESULTS

Models	Acc (%)
CL-CondenseNetV2 (Add adversarial examples)	99.71
CL-CondenseNetV2	99.45
DAG-ResNet	98.80
CondConvSENet	97.60

As can be seen from the table, the recognition rate of the CL- CondenseNetV2 model is higher than that of other models, while the classification recognition rate of the CL-CondenseNetV2 model after adding counter samples is far higher than that of other models. The feasibility and necessity of adding adversarial samples in model training are illustrated.

VI. CONCLUSIONS

In this paper, the proposed CL-CondenseNetV2 based on CA attention and CondenseNetV2 effectively improves the network's attention to feature space and enhances the accuracy of identifying agricultural diseases and pests. CL-CondenseNetV2 obtains 99.45% recognition accuracy in comparative studies with many models, outperforming classic CondenseNetV2, ResNet18, ResNet50, and Vgg16. This paper proposes a new adversarial attack algorithm WT-MI-FGSM based on MI-FGSM with the introduction of wavelet transform and histogram equalization. The comparison experiments use different adversarial attack algorithms against various models. The experimental results reveal that WT-MI-FGSM has a greater attack success rate than FGSM, I-FGSM, and MI-FGSM when compared to conventional adversarial attack methods, and the perturbations are too small to be recognized by human eyes. Furthermore, the adversarial samples generated by WT-MI-FGSM are added to the training set. After adversarial training, the recognition rate of CL-CondenseNetV2 may reach 99.71%, which is 0.26% higher than the accuracy rate without adversarial training, effectively increasing the model recognition's accuracy and robustness. Adversarial training is an efficient method for increasing the model's robustness. However, it has drawbacks such as sluggish training speed and overfitting when trained on tiny data sets. As a result, enhancing the performance of adversarial training will be the main focus of future study.

REFERENCES

[1] Di, T. Analysis on integrated control of plant diseases and pests in landscaping. World Tropical Agriculture Information, 2022 ,05: 63-64.

[2] Szegedy, C. et al. (2014) Going deeper with convolutions, arXiv.org. Available at: https://arxiv.org/abs/1409.4842.

[3] SIMONYAN,Karen;ZISSERMAN,Andrew.Very deep convolutional networks for large-scale image recognition.arXiv preprint arXiv:1409.1556, 2014.

[4] HE,Kaiming,et al. Deep residual learning for image recognition. In:Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.

[5] Lv M, Zhou G, He M, et al. Maize leaf disease identification based on feature enhancement and dms-robust alexnet[J]. IEEE Access, 2020, 8: 57952-57966.

[6] Pandian J A, Geetharamani G, Annette B. Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques[C]. 2019 IEEE 9th International Conference on Advanced Computing (IACC). IEEE, 2019: 199-204.

[7] Durmus, H.; Gunes, E.O.; Kirci, M. A hybrid approach for noise reduction-based optimal classifier using genetic algorithm: A case study in plant disease prediction. In Proceedings of the 2017 6th International Conference on Agro-Geoinformatics, Fairfax V A, USA, 7–10 August 2017; pp. 1–5.

[8] Iandola, F.N.; Moskewicz, M.W.; Ashraf, K.; Han, S.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. arXiv 2016, arXiv:1602.07360.

[9] Guadarrama, L.; Paredes, C.; Mercado, O. Plant Disease Diagnosis in the Visible Spectrum. Appl. Sci. 2022, 12, 1023–1049.

[10] Kaur, M.; Bhatia, R. Development of an improved tomato leaf disease detection and classification method. In Proceedings of the 2019 IEEE Conference on Information and Communication Technology (CICT), Jeju, Korea, 6–18 October 2019; pp. 1–5.

[11] Fuentes A, Yoon S, Kim S C, et al. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition[J]. Sensors, 2017, 17(9): 2022.

[12] Tang, W.; Huang, Z. Lightweight model of tomato leaf diseases identification based on knowledge distillation. Jiangsu J. Agric.Sci. 2021, 37, 9.

[13] Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. arXiv 2019, arXiv:1412.6572.

[14] Moosavi-Dezfooli, S.; Fawzi, A.; Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks.In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las V egas, NV , USA, 27–30 June 2016; pp. 2574–2582

[15] Carlini, N.; Wagner, D.A. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.

[16] Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y .; Abbeel, P . Adversarial attacks on neural network policies. arXiv 2017, arXiv:1702.02284.

[17] Kurakin, A., Goodfellow, I. J., & Bengio, S. (2018). Adversarial examples in the physical world. In Artificial intelligence safety and security (pp. 99-112). Chapman and Hall/CRC.

[18] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., & Li, J. (2018). Boosting adversarial attacks with momentum. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9185-9193).

[19] HUANG, Gao, et al. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition.2017. p. 4700-4708.

[20] Huang, G., Liu, S., Van der Maaten, L., & Weinberger, K. Q. (2018). Condensenet: An efficient densenet using learned group convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2752-2761).

[21] Yang, L., Jiang, H., Cai, R., Wang, Y., Song, S., Huang, G., & Tian, Q. (2021). Condensenet v2: Sparse feature reactivation for deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3569-3578).

[22] Hu, J. et al. (2019) Squeeze-and-excitation networks, arXiv.org. Available at: https://arxiv.org/abs/1709.01507.

[23] Woo, S. et al. (2018) CBAM: Convolutional Block Attention Module, arXiv.org. Available at: https://arxiv.org/abs/1807.06521.

[24] Hou, Q., Zhou, D. and Feng, J. (2021) Coordinate attention for efficient mobile network design, arXiv.org. Available at: https://arxiv.org/abs/2103.02907.

[25] Robbins, H.E. A Stochastic Approximation Method. Ann. Math. Stat. 2007,22, 400–407. [SGD]

[26] Hughes D, Salathé M. An open access repository of images on plant health to enable the development of mobile disease diagnostics[J]. arXiv preprint arXiv: 1511.08060, 2015.[17]

[27] Bousmalis, K. et al. (2016) Domain separation networks, arXiv.org. Available at: https://arxiv.org/abs/1608.06019.