applied sciences

MDPI

Article

A Study of Adversarial Attacks and Detection on Deep Learning-Based Plant Disease Identification

Zhirui Luo, Qingqing Li and Jun Zheng *

Department of Computer Science and Engineering, New Mexico Institute of Mining and Technology, Socorro, NM 87801, USA; zhirui.luo@student.nmt.edu (Z.L.); qingqing.li@student.nmt.edu (Q.L.)
* Correspondence: jun.zheng@nmt.edu

Abstract: Transfer learning using pre-trained deep neural networks (DNNs) has been widely used for plant disease identification recently. However, pre-trained DNNs are susceptible to adversarial attacks which generate adversarial samples causing DNN models to make wrong predictions. Successful adversarial attacks on deep learning (DL)-based plant disease identification systems could result in a significant delay of treatments and huge economic losses. This paper is the first attempt to study adversarial attacks and detection on DL-based plant disease identification. Our results show that adversarial attacks with a small number of perturbations can dramatically degrade the performance of DNN models for plant disease identification. We also find that adversarial attacks can be effectively defended by using adversarial sample detection with an appropriate choice of features. Our work will serve as a basis for developing more robust DNN models for plant disease identification and guiding the defense against adversarial attacks.

Keywords: plant disease identification; deep learning; adversarial attacks; white-box attacks; adversarial sample detection

Citation: Luo, Z.; Li, Q.; Zheng, J. A Study of Adversarial Attacks and Detection on Deep Learning-Based Plant Disease Identification. *Appl. Sci.* **2021**, *11*, 1878. https://doi.org/10.3390/app11041878

Academic Editor: Sungju Lee

Received: 17 January 2021
Accepted: 16 February 2021
Published: 20 February 2021

1. Introduction

On a global scale, pathogens and pests are one major reason that reduces the yield and quality of agricultural production. According to the study of [1], estimated yield losses of five major crops (wheat, rice, maize, potato, and soybean) due to pathogens and pests range from 10.1% to 41.1% globally. Identifying plant diseases in an early time can prevent further yield losses of production by informing farmers of appropriate treatment processes regarding the diagnosis. Advanced lab-based methods for plant disease diagnosis such as DNA-based and serological methods, are accurate and authentic [2]. However, these methods are either more time-consuming or more costly than those based on the visual observation of symptoms shown on the organs of species. Traditionally the visual observations are performed by experienced experts or producers which are labor intensive and not reliable [3,4]. In recent years, due to the advancement of information and communication technologies (ICT), acquiring images from farms can be done easily by human observers using mobile devices [5] or automated sensing technologies such as unmanned aerial vehicles (UAVs) [6]. Thus, automatic identification of plant diseases using plant leaf images becomes more and more popular [7].

The automation of plant disease identification methods is achieved through predictive models built with machine learning algorithms. Traditional machine learning algorithms such as k-nearest neighbor (KNN) [8], artificial neural network (ANN) [9], support vector machine (SVM) [10], and random forest [11] have been widely applied for the problem. These algorithms rely heavily on features generated from plant leaf images, which require advanced image processing techniques and extensive involvement of domain experts. Recently, deep learning (DL) has emerged as a promising solution for many computer vision applications including image-based plant disease identification [12]. Instead of relying on advanced image processing techniques and domain experts, DL-based methods

use deep neural networks (DNNs) that are capable of automatically extracting image features from raw data [13]. In addition, it has been shown that DL offers significantly better detection performance than traditional machine learning algorithms [3,5,12,14,15]. The major challenge associated with DL-based methods is the need for large amounts of data and vast computing resources to train DNN models. Fortunately, transfer learning solves this problem by using a pre-trained model from a similar domain instead of starting the model training from scratch [16,17]. Majority of DL-based plant disease identification models were built based on pre-trained DNN models such as VGGNet [18], ResNet [19], Inception [20], and DenseNet [21].

Although DL models have shown superior performance in many applications, they are susceptible to carefully crafted adversarial attacks. Adversaries can easily perturb normal samples to produce adversarial samples which cause DL models to make wrong predictions [22]. Adversarial attacks against DL can be categorized as white-box, gray-box, and black-box attacks of which the difficulties increase in order [23]. The pre-trained model used in transfer learning is usually publicly available to both normal users and adversaries. Based on this vulnerability, an adversary can generate adversarial samples solely with the knowledge of the pre-trained model to launch an effective and efficient white-box attack [24]. Recently web-based plant disease identification systems with DL have been proposed which use leaf images uploading from smartphones [25,26]. Adversaries can intercept uploaded normal images and apply white-box attacks to convert them to adversarial images. In the end, the misdiagnosis of plant diseases by the systems could result in a significant delay of treatments and huge economic losses. In this paper, we conduct a comprehensive study of the effects of popular white-box adversarial attacks on pre-trained DNN models widely used in plant disease identification. We also investigate the effectiveness of different adversarial sample detection methods on defending adversarial attacks. To the best of our knowledge, our work is the first attempt to investigate adversarial attacks and detection on DL-based plant disease identification.

The rest of this paper is organized as follows. In Section 2, the popular pre-trained DNN models for plant disease identification are introduced followed by the description of white-box adversarial attacks and adversarial sample detection methods. The experiments and results of adversarial attacks and detection on DL-based plant disease identification are presented in Section 3. Finally, we conclude the paper in Section 4.

2. Methods

In this section, we first describe the problem of plant disease identification. Popular pre-trained DNN models for plant disease identification are then introduced. After that, white-box adversarial attacks and adversarial sample detection methods adopted in this paper are presented.

2.1. Plant Disease Identification Problem

Plant disease identification is a classification problem that can be either 2-class or multi-class. A 2-class model classifies a plant leaf image as healthy or diseased while a multi-class model does a more fine-grained classification to predict the input image as healthy or a certain type of disease. Figure 1 shows the system architecture adopted in this paper that applies a DL model for plant disease identification. To alleviate the problem of insufficient training data for a DL model, the system transfers knowledge from the source domain of ImageNet [27] to the target domain of plant disease detection. This process first employs the partial network from the source domain to serve as a feature extractor of the new network and then replaces the final output layer of the source domain network with a new dense layer followed by a SoftMax function corresponding to the plant disease dataset. Finally, the new network is fine-tuned by using the plant disease dataset. Fine-tuning can optimize network parameters wholly or partially [28]. The shallow training of our work fine-tunes all network parameters.

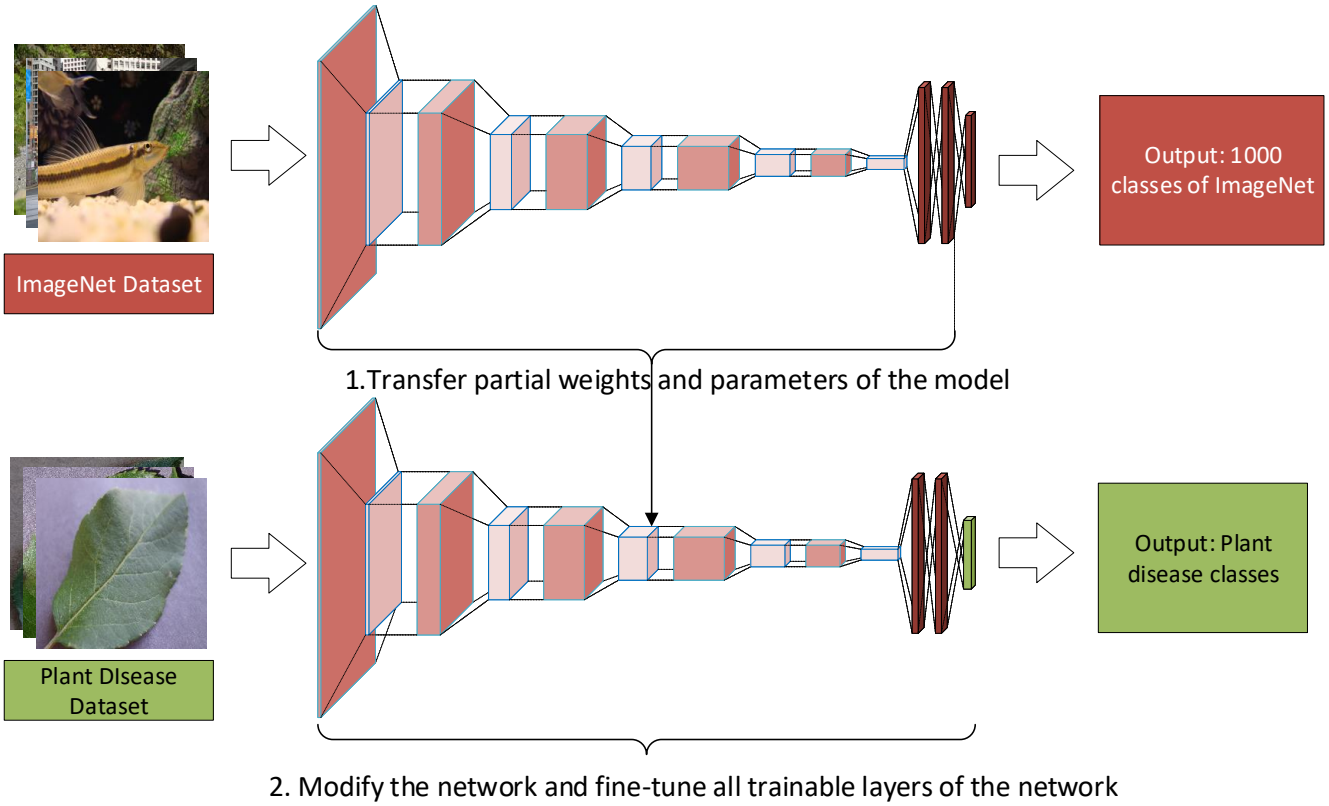


Figure 1. Architecture of a DL-based plant disease identification system.

2.2. Pre-Trained DNN Models for Plant Disease Identification

In this study, we consider four pre-trained DNN models that have been widely applied for plant disease identification: VGGNet [3,5,17,29–32], ResNet [17,29,33], Inception [17,33,34] and DenseNet [4,17,35].

2.2.1. VGGNet

VGGNet is a deep convolutional neural network (CNN) model proposed for the ILSVRC-2014 challenge [18]. The input of the model is a fixed-size 224 × 224 image, which passes through a stack of convolutional layers with 3 × 3 filter. The model also uses 2 × 2 max-pooling layer following some convolutional layers for down-sampling which reduces the input size of later layers. The end of the network consists of two fully connected layers with 4096 neurons each followed by a SoftMax layer. Depending on the number of convolutional layers of the network, there are two VGGNet architectures: VGG-16 and VGG-19. VGG-16 is considered in our study.

2.2.2. ResNet

Deep residual networks (ResNet) were proposed by He et al. in [19], which have shown compelling performance and good convergence behaviors. The ResNet architecture accepts a 224 × 224 image as input. It consists of a stack of residual blocks, which are feed-forward neural networks with shortcuts (or skip connections). Shortcuts are connections skipping over some layers which are used to deal with the problem of vanishing-gradients as the network goes deeper. The ResNet architecture considered in this paper is ResNet-101.

2.2.3. Inception

The idea of Inception was first introduced in [20] as a module for the GoogleNet architecture, which approximates an optimal local sparse structure of a convolutional

network by dense components. An Inception Module is a stack of a max-pooling layer and convolution layers, which is the basic module to construct the Inception network. Inception V3 proposed in [36] is considered in this paper, which is the 3rd version of Inception architecture. Inception V3 inherits the basic idea of "Inception Module" and the batch normalization introduced in Inception V2 [37]. In addition, three new features are added to Inception V3 including convolution factorization, efficient grid size reduction, and auxiliary classifier [36]. The input of Inception V3 is a fixed-size 299 × 299 image.

2.2.4. DenseNet

DenseNet was introduced by Huang et al. in [21], which maximizes the information flow between layers by connecting a layer to other layers in a feed-forward manner. The inputs of a layer in DenseNet are features maps of all preceding layers. The feature maps of a layer will then be used as inputs of all subsequent layers. DenseNet requires significantly fewer parameters than traditional deep CNNs [21]. It also provides other benefits including alleviating the vanishing-gradient problem, strengthening feature propagation, and encouraging feature reuse [21]. In this paper, DenseNet-121 is considered to be the DenseNet architecture, which accepts a fixed-size 224 × 224 image as input.

2.3. Adversarial Attacks

The idea of using adversarial samples to cause the misclassification of a DL model was first explored by Szegedy et al. [22]. There are two kinds of adversarial samples that can be crafted. Given a dataset $(\boldsymbol{x}, y)$ where $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ is a normal sample and $y = f(\boldsymbol{x})$ is the corresponding label of the sample, an untargeted adversarial sample $\boldsymbol{x'}$ is crafted to make $f(\{x \neq y$ yet $\boldsymbol{x}$ and $\boldsymbol{x'}$ are close according to certain metric. Another more powerful but harder attack uses targeted adversarial samples. Given a normal sample $\boldsymbol{x}$ and a target label $y' \neq f(\boldsymbol{x})$, the attacking algorithm searches for an adversarial sample $\boldsymbol{x'}$ such that $f(\boldsymbol{x'}) = y'$ where $\boldsymbol{x}$ and $\boldsymbol{x'}$ are close. To measure the similarity between $\boldsymbol{x}$ and $\boldsymbol{x'}$, L_0, L_2, L_∞ are the three most widely used distance metrics among all L_p-norms for generating adversarial samples [38]. The L_p distance is also written as $||\boldsymbol{x} - \boldsymbol{x'}||_p$, where $|| \cdot ||_p$ of $\boldsymbol{x}$ is defined as

$$||\boldsymbol{x}||_p = (|x_1|^p) + |x_2|^p + \cdots + |x_n|^p)^{1/p} \quad (1)$$

For the three distance metrics, L_0 norm measures the number of points that differ between $\boldsymbol{x}$ and $\boldsymbol{x'}$, L_2 norm is the standard Euclidean distance, and L_∞ norm measures the maximum change to any of the points. Although all three distance metrics approximate to the human perceptual similarity, L_∞, also known as max-norm, is the most commonly used one due to its better consistency to human perception [39].

There are three types of adversarial attacks: white-box, gray-box, and black-box attacks. In this work, we focus on untargeted white-box attacks under the L_∞ norm distance metric. White-box attacks require attackers have the highest-level knowledge of the model among the three types of attacks, which then have a greater impact on the performance of DL-based models than other two types of attacks. Since many DL-based plant disease identification schemes are built based on pre-trained DNN models, adversarial samples generated in the white-box manner against a fine-tuned model based on a pre-trained DNN model can be successfully transferred to the target model [24]. In the following, we describe four popular white-box attacks considered in this study: fast gradient sign method (FGSM) [40], basic iterate method (BIM) [41], projected gradient descent (PGD) [39], and Carilini and Wagner attack (CW) [38].

2.3.1. FGSM

FGSM is a popular untargeted white-box attack introduced by Goodfellow et al. [40], which perturbs one-step along the gradient direction of the adversarial loss $J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ with a max-norm constraint of ϵ:

$$\boldsymbol{x'} = \boldsymbol{x} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)) \tag{2}$$

2.3.2. BIM

BIM was proposed in [41] which extends FGSM as an iterative method. BIM perturbs the input $\boldsymbol{x}$ iteratively with a step size α under the max-norm ϵ.

$$\boldsymbol{x'_0} = \boldsymbol{x}, \quad \boldsymbol{x'_t} = Clip_{\boldsymbol{x},\epsilon}(\boldsymbol{x'_{t-1}} + \alpha \cdot \text{sign}(\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x'_{t-1}}, y))) \tag{3}$$

where $\boldsymbol{x'_t}$ is the adversarial sample generated at t-th step, $J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ is the optimization loss of adversarial attack in which $\boldsymbol{\theta}$ represents the weights of model. The number of perturbation steps, T, is chosen heuristically [41]. The step size is usually set to $\epsilon/T \leq \alpha < \epsilon$. $Clip_{x,\epsilon}(x')$ is defined as following.

$$Clip_{\boldsymbol{x},\epsilon}(\boldsymbol{x'}) = \min\{\text{img\_max}, \boldsymbol{x} + \epsilon, \max\{\text{img\_min}, \boldsymbol{x} - \epsilon, \boldsymbol{x'}\}\} \tag{4}$$

where *img_max* and *img_min* are the maximum and minimum of image range, e.g., 1 and 0 for images in the range [0,1].

2.3.3. PGD

PGD attack was proposed by Madry et al. [39] to find adversarial samples. PGD perturbs a normal sample $\boldsymbol{x}$ for a total of T steps where each step perturbs $\boldsymbol{x}$ in the gradient direction of the adversarial loss with a projection constraint, which is a set of allowed perturbations denoted as $S \subseteq \mathbb{R}^d$.

$$\boldsymbol{x'_t} = \Pi_{\boldsymbol{x}+S}(\boldsymbol{x'_{t-1}} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x'_{t-1}}, y))) \tag{5}$$

where α is the step size, and $\Pi(\cdot)$ is the projection function which projects the intermediate perturbation into the valid data range and the L_∞-ball around the normal sample $\boldsymbol{x}$. PGD is similar to BIM with the differences of the projection step and random start.

2.3.4. CW

CW attack is an optimization-based adversarial attack [38], which achieves a perfect attack success rate against defensive distillation, an efficient approach hardening neural networks against adversarial samples [42]. CW attack creates an adversarial sample $\boldsymbol{x'}$ from a normal sample $\boldsymbol{x} \in [0, 1]$ by minimizing $||\delta||p + c \cdot f(\boldsymbol{x} + \delta)$ such that $\boldsymbol{x} + \delta \in [0, 1]^n$, where δ is the pixel-wise difference between $\boldsymbol{x}$ and $\boldsymbol{x'}$, p denotes the L_p norm distance metric (L_∞ in our study). The box constraints of CW attack uses the idea of *change of variables* instead of projected gradient descent and clipped gradient descent used in PGD and BIM, respectively [38], which introduces and optimizes over a new variable w to smooth the box constraint process as follows:

$$\delta = \frac{1}{2}[\tanh(w) + 1] - x \tag{6}$$

2.4. Detection of Adversarial Samples

Several adversarial defenses have been proposed in recent years for DL models such as adversarial training [22,40], input data compression [43], gradient regularization [44], and defensive distillation [42]. However, those defenses were proved later that do not work partially or wholly [45]. Therefore, recent research has more focused on detection-based defenses [46–48] that detect adversarial samples using features extracted from trained DL

models. In the following, we describe adversarial sample detection methods investigated in our study.

2.4.1. Kernel Density (KD) and Bayesian Uncertainty (BU)

KD estimation was proposed in [46] as a measure to submanifold in the feature space of the last hidden layer. The assumption is that an adversarial sample lies far from the normal data manifold. Given a sample $\boldsymbol{x}$ and a training set $\boldsymbol{X_t}$ of class t, the KD estimation of $\boldsymbol{x}$ based on Gaussian distribution is calculated as:

$$KD(\boldsymbol{x}) = \frac{1}{|\boldsymbol{X_t}|} \sum_{\boldsymbol{x_i} \in \boldsymbol{X_t}} \exp(||\boldsymbol{x_i} - \boldsymbol{x}||^2 / \sigma^2) \tag{7}$$

where σ is the bandwidth of the kernel. σ controls the smoothness of the density estimation which we heuristically set it to 1.2. BU, the second detection method proposed in [46], is based on an approximation from dropout mechanism in DNNs to the deep Gaussian process. The uncertainty is the additional information to the label prediction which gives a confidence interval to the prediction. Features extracted with KD and BU can be used as the input of a machine learning-based detector.

2.4.2. LID

LID models dimensional characteristics of adversarial subspaces based on the distance distribution amid adversarial samples [48]. The argument of [48] is that KD can fail to differentiate adversarial samples from normal samples which are differentiable in high-dimension manifold but not in low-dimension manifold. Given a sample $\boldsymbol{x}$, the maximum likelihood estimator of LID uses its distances to k nearest neighbors:

$$\hat{LID}(\boldsymbol{x}) = -\left(\frac{1}{k} \sum_{i=1}^{k} \log \frac{r_i(\boldsymbol{x})}{r_k(\boldsymbol{x})} \right) \tag{8}$$

where r_i is the distance between $\boldsymbol{x}$ and its i-th nearest neighbor, $r_k(\boldsymbol{x})$ is the distance between $\boldsymbol{x}$ and $\boldsymbol{x}$'s furthest neighbor of k nearest neighbors. A LID-based detector is built based on LID computed from each layer of the DNN under a mini-batch manner for given training samples.

2.4.3. SafetyNet

Lu et al. [47] proposed an adversarial sample detection architecture called SafetyNet which uses RBF-SVM as the adversarial detector with features extracted from the outputs of later activation layers. The hypothesis of the method is that adversarial samples produce different patterns of activation in late stage than those produced by normal samples. There are two different kinds of features used in [47]: raw features extracted directly from activation denoted as *DeepF* and discrete features obtained by quantizing activation as discrete levels denoted as *DiscF*. *DiscF* forces the attacker to solve a hard discrete optimization problem [47]. Both *DeepF* and *DiscF* are considered in our study.

3. Experiments and Results

In this section, the efficacy of adversarial attacks (FGSM, BIM, PGD, and CW) against four popular DNN models (VGG-16, ResNet-101, Inception V3, and DenseNet-121) for plant disease identification is investigated. The effectiveness of adversarial sample detection methods against adversarial attacks is also studied. Without loss of generality, we present the results of apple leaf disease identification for which several DL models have been developed [30,35]. Although we only report the results of apple leaf disease identification, our unreported experiments obtained similar results from other leaf disease datasets.

3.1. Datasets

We use a publicly available apple leaf disease dataset which is a subset of the PlantVillage dataset [49]. Table 1 shows the details of the dataset including classes of apple leaf images and the number of images for each class. The dataset can be directly used to build multi-class disease identification models that a leaf image is labeled as one of the four classes (healthy or one of the three diseases). To build 2-class disease identification models, all images of three disease classes are labeled as "diseased" which are combined with healthy images to form the dataset.

Table 1. Apple leaf disease dataset.

Class	Number of Images
Scab	630
Black Rot	621
Cedar Apple Rust	275
Healthy	1645
Total	3171

3.2. Performance of Fine-Tuned DNN Models without Adversarial Attacks

To evaluate the performance of different DNN models without adversarial attacks, the apple leaf disease dataset (2-class or multi-class) is divided as a training set and a testing set with an 80/20 ratio. For each DNN model, a leaf image is resized as the standard input size required by a pre-trained DNN model. In the fine-tuning phase, each DNN model is fine-tuned from weights pre-trained using ImageNet dataset [27]. The models are trained with a stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.001 and momentum of 0.9. The learning rate decays with a rate of 0.1 every 7 epochs. Our shallow training has 100 epochs with an early stopping of 7-epoch tolerance. The test accuracy of the four fine-tuned DNN models are presented in Table 2. It can be seen that without adversarial attacks all models perform very well on both 2-class and multi-class disease identification.

Table 2. Performance of fine-tuned DNN models without adversarial attacks.

DNN Model	Dataset	Test Accuracy
VGG-16	2-class	100%
VGG-16	multi-class	99.67%
ResNet-101	2-class	99.84%
ResNet-101	multi-class	100%
Inception V3	2-class	100%
Inception V3	multi-class	100%
DenseNet-121	2-class	100%
DenseNet-121	multi-class	100%

3.3. Efficacy of Adversarial Attacks

To investigate the efficacy of adversarial attacks, we perturb the testing set used in Section 3.2. Each of the four adversarial attacks of Section 2.3 is applied to generate adversarial samples from randomly selected 50% of the test samples. The generated adversarial samples are combined with another half of normal samples as the testing set for adversarial attacks. Please note that all four attacks are bounded by a pre-defined maximum perturbation size ϵ with respect to the L_∞ norm. We generate different testing sets by varying ϵ from 0.2/255 to 4/255. Examples of adversarial images generated by the four attacks and their corresponding normal images under $\epsilon = 1/255$ for fine-tuned 2-class and multi-class VGG-16 models are shown in Figures 2 and 3, respectively. It can be seen

that adversarial images generated by the four attacks under small perturbations are hard to distinguish from original images by human eyes.

Figures 4 and 5 show the results of applying adversarial attacks on fine-tuned 2-class and multi-class DNN models, respectively. It can be observed that the results of 2-class and multi-class models are similar. For all models, the accuracy of disease identification drops significantly as ϵ increases which demonstrates the efficacy of the attacks. One can find that three iterative perturbation attacks (BIM, PGD, and CW) are more efficient than the only one-step perturbation attack, FGSM. Another interesting finding is that VGG-16 is the most robust one against adversarial attacks among the four DNN models although its performance is still significantly degraded under attacks.

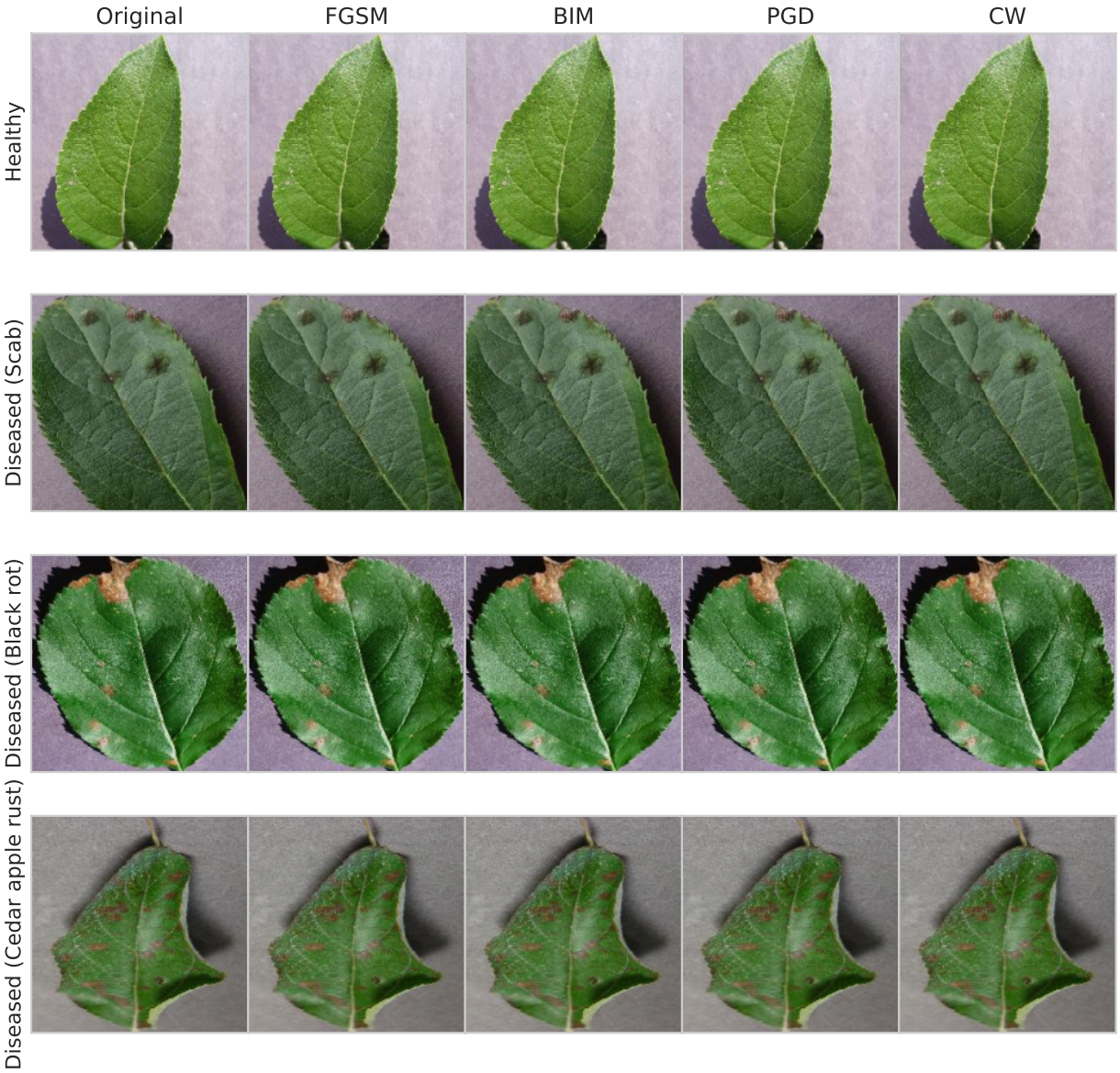


Figure 2. Examples of normal images and adversarial images generated by different attacks (VGG-16, 2-class, $\epsilon = 1/255$).

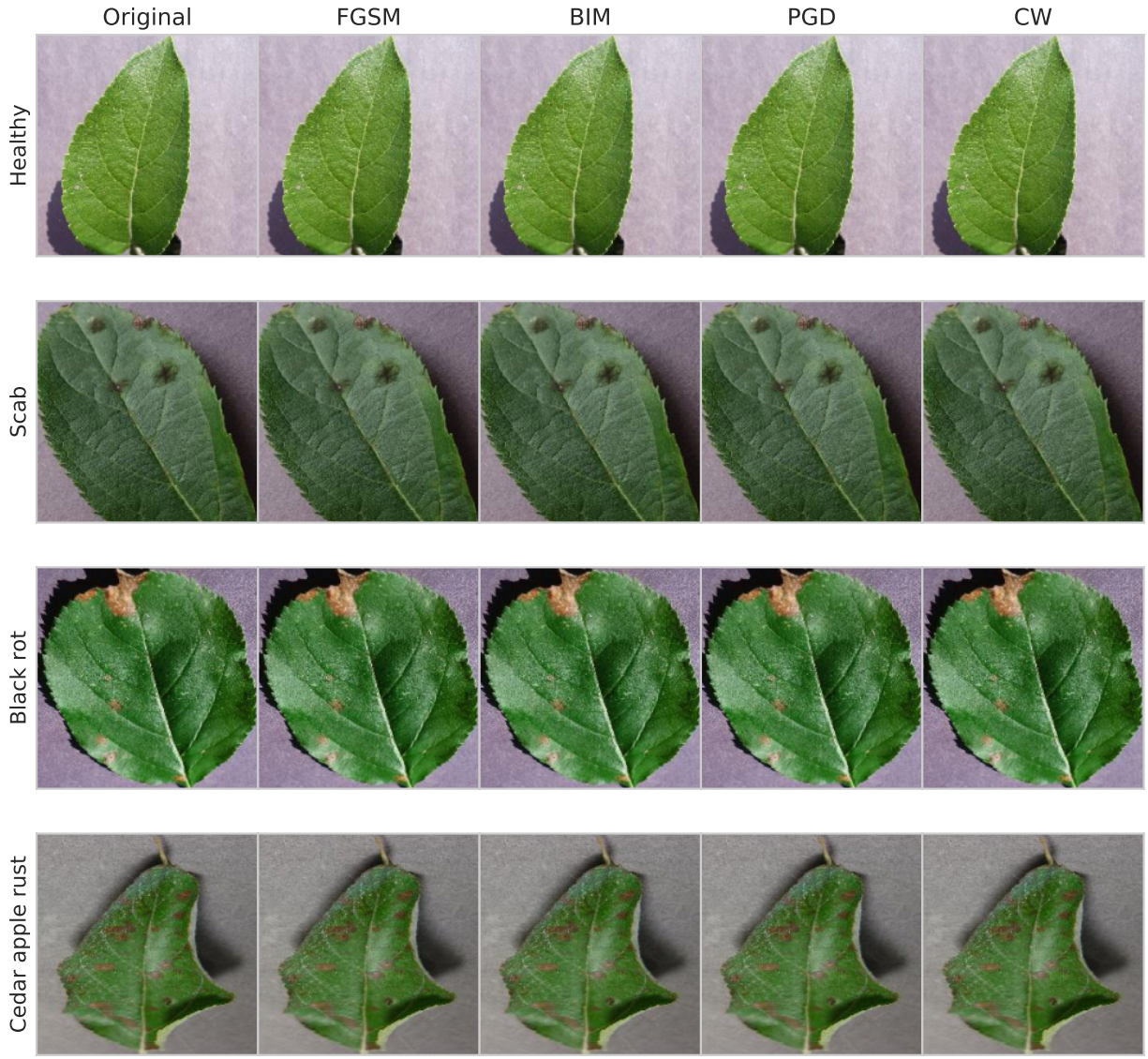
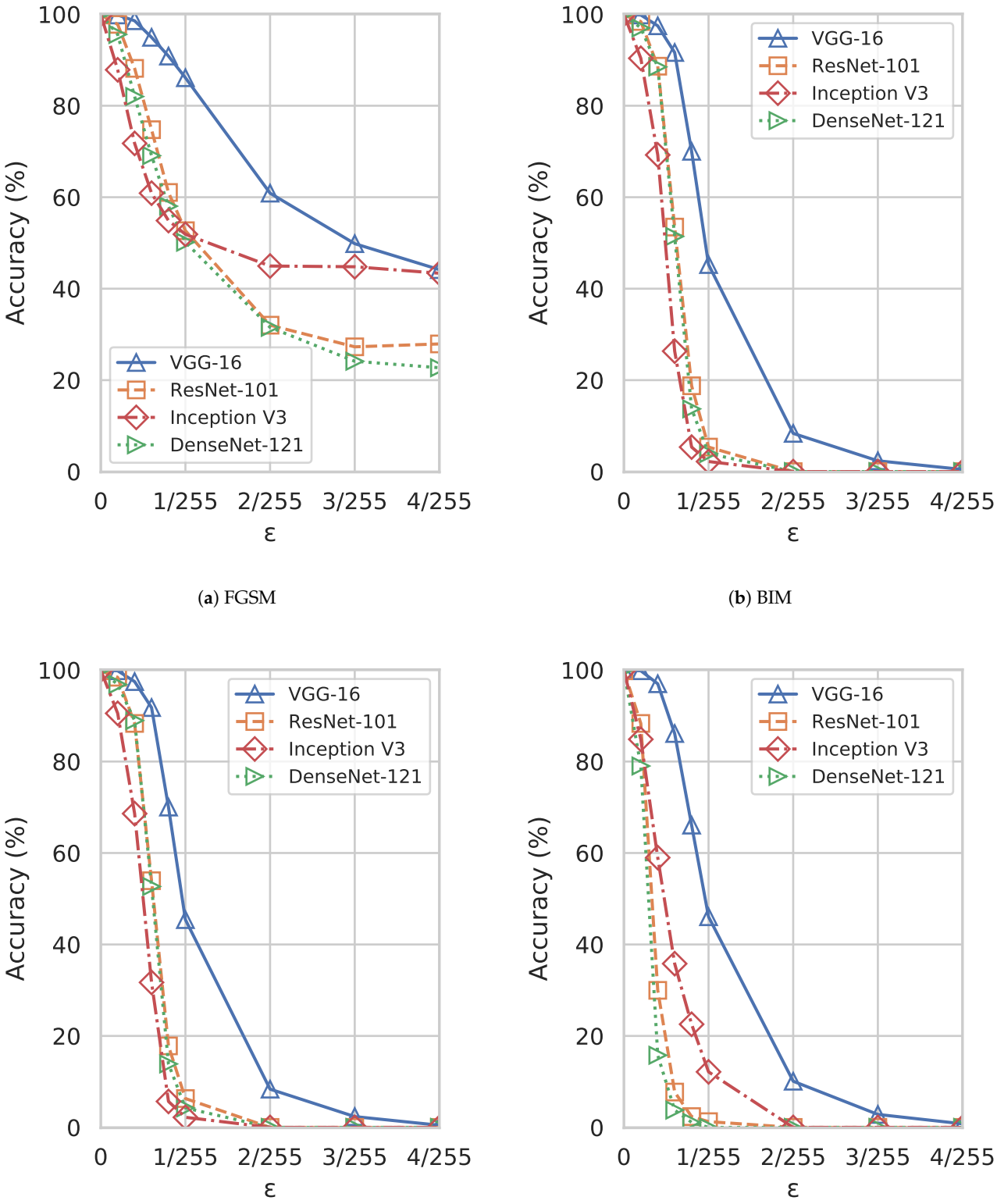


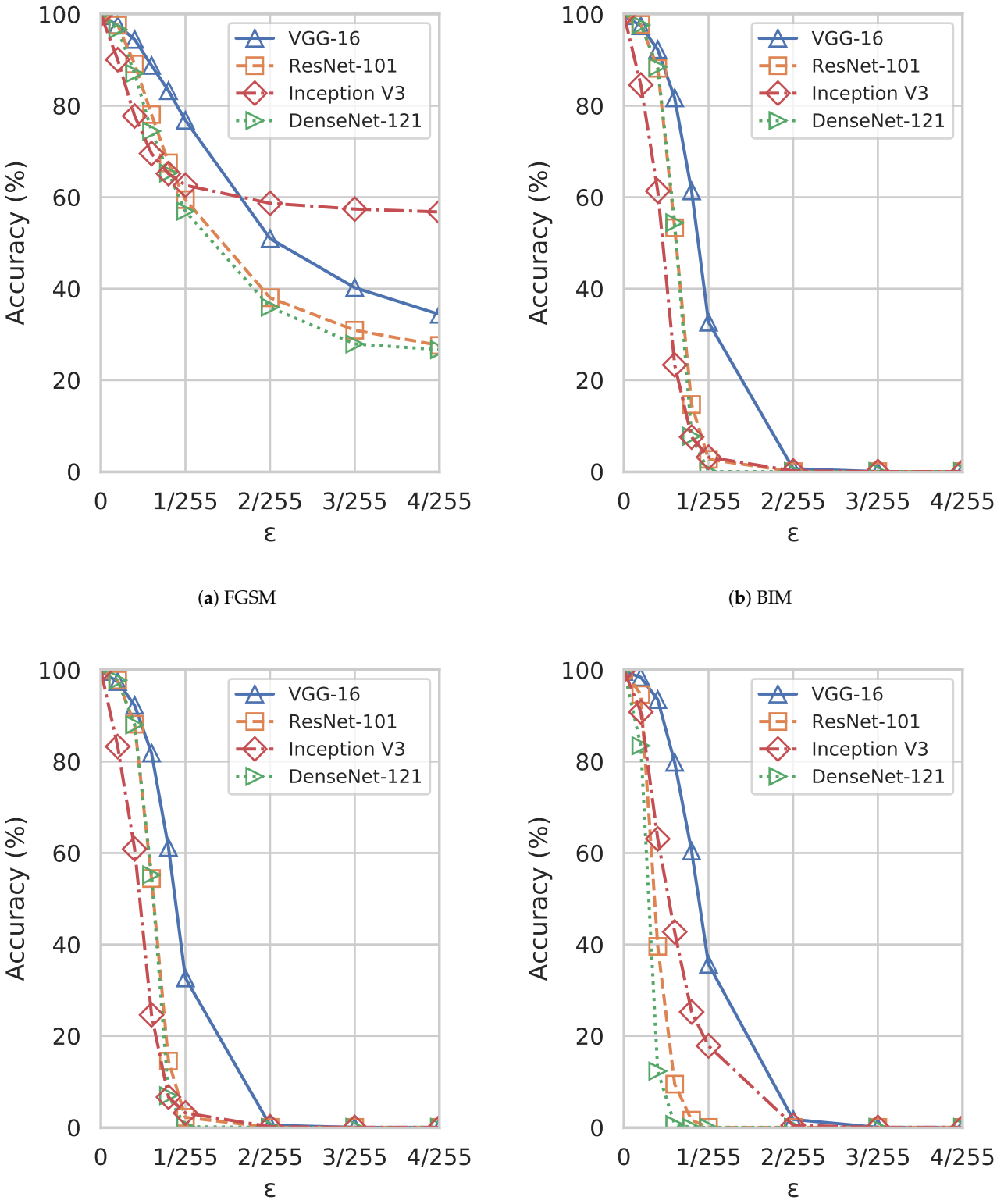
Figure 3. Examples of normal images and adversarial images generated by different attacks (VGG-16, multi-class, $\epsilon = 1/255$).



(**a**) FGSM (**b**) BIM

(**c**) PGD (**d**) CW

Figure 4. Performance comparison of four adversarial attacks on 2-class DNN models under different ϵ.



(**a**) FGSM (**b**) BIM

(**c**) PGD (**d**) CW

Figure 5. Performance comparison of four adversarial attacks on multi-class DNN models under different ϵ.

3.4. Results of Adversarial Sample Detection

The testsets generated for adversarial attacks with $\epsilon = 1/255$ in Section 3.3 are used as the datasets for evaluating the performance of adversarial sample detection methods. Detection features are extracted from the fine-tuned DNN models of Section 3.2. The KD and BU features are estimated from the second-last layer and the SoftMax layer of the network, respectively. The LID features are calculated from the outputs of all layers of the network with a min-batch size of 100. *DeepF* and *DiscF* features are obtained from the second-last layer of the network. According to [46,48], logistic regression classifier is used for KD + BU and LID features. RBF-SVM classifier is used for *DeepF* and *DiscF* features based on the SafetyNet architecture [47]. We apply a 5-fold cross-validation to evaluate the performance of different adversarial sample detection methods.

Tables 3 and 4 show the results of adversarial sample detection for fine-tuned 2-class and multi-class DNN models, respectively. The results show that KD + BU features achieve significantly better performance than other features with a nearly perfect detection rate. This demonstrates that adversarial attacks on DL-based plant disease identification models can be effectively defended by using adversarial sample detection with an appropriate choice of features. Surprisingly, LID features are the worst performed ones which were shown superior performance over KD + BU features on three benchmark image datasets: MNIST, CIFAR-10, and SVHN in [48]. This implies that the intrinsic characteristics of leaf images used for plant disease identification are different from those of benchmark images.

Table 3. Adversarial sample detection results for 2-class DNN models using different features.

DNN Model	Features	FGSM	BIM	PGD	CW
VGG-16	KD + BU	**0.998**	**1**	**1**	**0.998**
	LID	0.738	0.666	0.633	0.637
	DeepF	0.882	0.899	0.902	0.732
	DiscF	0.918	0.921	0.92	0.737
ResNet-101	KD + BU	**1**	**1**	**1**	**1**
	LID	0.736	0.698	0.607	0.807
	DeepF	0.94	0.948	0.957	0.945
	DiscF	0.946	0.968	0.956	0.91
Inception-V3	KD + BU	**1**	**1**	**0.998**	**0.998**
	LID	0.754	0.74	0.705	0.587
	DeepF	0.942	0.907	0.912	0.839
	DiscF	0.954	0.905	0.885	0.838
DenseNet-121	KD + BU	**1**	**1**	**1**	**1**
	LID	0.738	0.606	0.571	0.666
	DeepF	0.962	0.942	0.948	0.94
	DiscF	0.967	0.984	0.967	0.916

Table 4. Adversarial sample detection results for multi-class DNN models using different features.

DNN Model	Features	FGSM	BIM	PGD	CW
VGG-16	KD + BU	**0.987**	**1**	**1**	**0.987**
	LID	0.728	0.674	0.641	0.612
	DeepF	0.867	0.785	0.782	0.696
	DiscF	0.91	0.864	0.889	0.756
ResNet-101	KD + BU	**1**	**1**	**1**	**1**
	LID	0.7	0.626	0.639	0.793
	DeepF	0.844	0.801	0.756	0.874
	DiscF	0.926	0.896	0.912	0.913
Inception V3	KD + BU	**0.998**	**1**	**1**	**0.997**
	LID	0.74	0.719	0.663	0.675
	DeepF	0.872	0.73	0.733	0.741
	DiscF	0.948	0.864	0.875	0.825
DenseNet-121	KD + BU	**1**	**1**	**1**	**1**
	LID	0.691	0.629	0.604	0.675
	DeepF	0.898	0.855	0.871	0.784
	DiscF	0.957	0.946	0.951	0.894

Finally, we investigate the transferability of detection models built with KD + BU features. The detection model for a DNN model is trained with a testset generated in Section 3.3 that consists of normal samples and adversarial samples generated by one of the four attacks. The model is then applied for detecting adversarial samples generated by other three attacks. Tables 5 and 6 show the detection performance in terms of accuracy for 2-class and multi-class DNN models, respectively. It can be seen that detection models for 2-class and multi-class DNN models have comparable transferability. Detection models trained with adversarial samples generated by CW attack have the best transferability which can perfectly detect adversarial samples generated by other three attacks except one case (Inception V3, 2-class).

Table 5. Results of detection model transferability for 2-class DNN models.

Model	Source	FGSM	BIM	PGD	CW
VGG-16	FGSM	–	1	1	0.768
	BIM	0.846	–	0.861	0.622
	PGD	0.988	0.994	–	0.746
	CW	1	1	1	–
ResNet-101	FGSM	–	1	1	0.994
	BIM	1	–	1	0.994
	PGD	1	1	–	0.995
	CW	1	1	1	–
Inception V3	FGSM	–	1	1	0.840
	BIM	0.997	–	1	0.819
	PGD	0.997	1	–	0.820
	CW	0.983	1	1	–
DenseNet-121	FGSM	–	1	1	0.957
	BIM	1	–	1	0.961
	PGD	1	1	–	0.957
	CW	1	1	1	–

Table 6. Results of detection model transferability for multi-class DNN models.

Model	Source	FGSM	BIM	PGD	CW
VGG-16	FGSM	–	1	1	0.802
	BIM	1	–	1	0.805
	PGD	0.998	1	–	0.802
	CW	1	1	1	–
ResNet-101	FGSM	–	1	1	0.976
	BIM	1	–	1	0.986
	PGD	1	1	–	0.978
	CW	1	1	1	–
Inception V3	FGSM	–	1	1	0.826
	BIM	1	–	1	0.824
	PGD	0.999	0.999	–	0.800
	CW	1	1	1	–
DenseNet-121	FGSM	–	1	1	0.953
	BIM	1	–	1	0.957
	PGD	1	1	–	0.957
	CW	1	1	1	–

4. Conclusions

Pre-trained DNN models have been widely used in machine learning and computer vision applications including plant disease identification. In this paper, the vulnerabilities of DL-based plant disease identification models under four popular white-box adversarial attacks are investigated. Our results show that all attacks can significantly affect the performance of DNN models for plant disease identification. A small number of perturbations introduced by the attacks on acquired leaf images can lead to a significant degradation of disease identification performance. It is found that VGG-16 is more robust against attacks than other DNN models. We then study the effectiveness of adversarial sample detection methods based on features extracted from fine-tuned DNN models. The results show that attacks can be effectively detected with an appropriate choice of features such as KD + BU. The findings of this paper will serve as a basis for developing more robust DNN models for plant disease identification and guiding the defense against adversarial attacks.

Author Contributions: Conceptualization, Z.L. and J.Z.; methodology, Z.L., Q.L. and J.Z.; software, Z.L. and Q.L.; data curation, Z.L. and Q.L.; writing–original draft preparation, Z.L., Q.L. and J.Z.; writing-review and editing, J.Z.; supervision, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Science Foundation EPSCoR Cooperative Agreement OIA-1757207 and the Institute for Complex Additive Systems Analysis (ICASA) of New Mexico Tech.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Savary, S.; Willocquet, L.; Pethybridge, S.J.; Esker, P.; McRoberts, N.; Nelson, A. The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* **2019**, *3*, 430–439. [CrossRef] [PubMed]
2. Martinelli, F.; Scalenghe, R.; Davino, S.; Panno, S.; Scuderi, G.; Ruisi, P.; Villa, P.; Stroppiana, D.; Boschetti, M.; Goulart, L.R.; et al. Advanced methods of plant disease detection. A review. *Agron. Sustain. Dev.* **2015**, *35*, 1–25. [CrossRef]

3. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [CrossRef]
4. Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Hassanien, A.E.; Pandey, H.M. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. *Comput. Electron. Agric.* **2020**, *175*, 105456. [CrossRef]
5. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
6. Wolfert, S.; Ge, L.; Verdouw, C.; Bogaardt, M.J. Big Data in Smart Farming—A review. *Agric. Syst.* **2017**, *153*, 69–80. [CrossRef]
7. Kaur, S.; Pandey, S.; Goel, S. Plants Disease Identification and Classification Through Leaf Images: A Survey. *Arch. Comput. Methods Eng.* **2019**, *26*, 507–530. [CrossRef]
8. Hossain, E.; Hossain, M.F.; Rahaman, M.A. A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 1–6.
9. Golhani, K.; Balasundram, S.K.; Vadamalai, G.; Pradhan, B. A review of neural networks in plant disease detection using hyperspectral data. *Inf. Process. Agric.* **2018**, *5*, 354–371. [CrossRef]
10. Padol, P.B.; Yadav, A.A. SVM classifier based grape leaf disease detection. In Proceedings of the 2016 Conference on Advances in Signal Processing (CASP), Pune, India, 9–11 June 2016; pp. 175–179.
11. Sandika, B.; Avil, S.; Sanat, S.; Srinivasu, P. Random forest based classification of diseases in grapes from images captured in uncontrolled environments. In Proceedings of the 2016 IEEE 13th International Conference on Signal Processing (ICSP), Chengdu, China, 6–10 November 2016; pp. 1775–1780.
12. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
14. Le, N.Q.K. Fertility-GRU: Identifying Fertility-Related Proteins by Incorporating Deep-Gated Recurrent Units and Original Position-Specific Scoring Matrix Profiles. *J. Proteome Res.* **2019**, *18*, 3503–3511. [CrossRef]
15. Le, N.Q.K.; Do, D.T.; Hung, T.N.K.; Lam, L.H.T.; Huynh, T.T.; Nguyen, N.T.K. A Computational Framework Based on Ensemble Deep Neural Networks for Essential Genes Identification. *Int. J. Mol. Sci.* **2020**, *21*, 9070. [CrossRef]
16. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. *A Survey on Deep Transfer Learning BT—Artificial Neural Networks and Machine Learning—ICANN 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 270–279.
17. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
18. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
21. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 2261–2269.
22. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014.
23. Ren, K.; Zheng, T.; Qin, Z.; Liu, X. Adversarial Attacks and Defenses in Deep Learning. *Engineering* **2020**, *6*, 346–360. [CrossRef]
24. Rezaei, S.; Liu, X. A Target-Agnostic Attack on Deep Models: Exploiting Security Vulnerabilities of Transfer Learning. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, Addis Ababa, Ethiopia, 26–30 April 2020.
25. Cruz, A.C.; Luvisi, A.; De Bellis, L.; Ampatzidis, Y. X-FIDO: An Effective Application for Detecting Olive Quick Decline Syndrome with Deep Learning and Data Fusion. *Front. Plant Sci.* **2017**, *8*, 1741. [CrossRef] [PubMed]
26. Ngugi, L.C.; Abelwahab, M.; Abo-Zahhad, M. Tomato leaf segmentation algorithms for mobile phone applications using deep learning. *Comput. Electron. Agric.* **2020**, *178*, 105788. [CrossRef]
27. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
28. Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; Feris, R. SpotTune: Transfer Learning Through Adaptive Fine-Tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
29. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
30. Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. *IEEE Access* **2019**, *7*, 59069–59080. [CrossRef]

31. Darwish, A.; Ezzat, D.; Hassanien, A.E. An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis. *Swarm Evol. Comput.* **2020**, *52*, 100616. [CrossRef]
32. Hernández, S.; López, J.L. Uncertainty quantification for plant disease detection using Bayesian deep learning. *Appl. Soft Comput.* **2020**, *96*, 106597. [CrossRef]
33. Maeda-Gutiérrez, V.; Galván-Tejada, C.E.; Zanella-Calzada, L.A.; Celaya-Padilla, J.M.; Galván-Tejada, J.I.; Gamboa-Rosales, H.; Luna-García, H.; Magallanes-Quintanar, R.; Guerrero Méndez, C.A.; Olvera-Olvera, C.A. Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases. *Appl. Sci.* **2020**, *10*, 1245. [CrossRef]
34. Ramcharan, A.; Baranowski, K.; McCloskey, P.; Ahmed, B.; Legg, J.; Hughes, D.P. Deep Learning for Image-Based Cassava Disease Detection. *Front. Plant Sci.* **2017**, *8*, 1852. [CrossRef]
35. Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* **2020**, *168*, 105146. [CrossRef]
36. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision . In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
37. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning (PMLR 37), Lille, France, 6–11 July 2015: pp. 448–456.
38. Carlini, N.; Wagner, D. 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.
39. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
40. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
41. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Examples in the Physical World. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
42. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, 22–26 May 2016; pp. 582–597.
43. Dziugaite, G.K.; Ghahramani, Z.; Roy, D.M. A Study of the Effect of JPG Compression on Adversarial Images. In Proceedings of the International Society for Bayesian Analysis (ISBA 2016) World Meeting, Sardinia, Italy, 13–17 June 2016.
44. Ross, A.S.; Doshi-Velez, F. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 1660–1669.
45. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Vienna, Austria, 25–31 July 2018; pp. 436–448.
46. Feinman, R.; Curtin, R.R.; Shintre, S.; Gardner, A.B. Detecting Adversarial Samples from Artifacts. *arXiv* **2017**, arXiv:1703.00410.
47. Lu, J.; Issaranon, T.; Forsyth, D. SafetyNet: Detecting and Rejecting Adversarial Examples Robustly. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 446–454.
48. Ma, X.; Li, B.; Wang, Y.; Erfani, S.M.; Wijewickrema, S.; Schoenebeck, G.; Song, D.; Houle, M.E.; Bailey, J. Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
49. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef] [PubMed]