agriculture

Article

Deep Network with Score Level Fusion and Inference-Based Transfer Learning to Recognize Leaf Blight and Fruit Rot Diseases of Eggplant

Md. Reduanul Haque [1] and Ferdous Sohel [1,2,*]

[1] Information Technology, Murdoch University, Murdoch, WA 6150, Australia
[2] Centre for Crop and Food Innovation, Food Futures Institute, Murdoch University, Murdoch, WA 6150, Australia
* Correspondence: f.sohel@murdoch.edu.au

Abstract: Eggplant is a popular vegetable crop. Eggplant yields can be affected by various diseases. Automatic detection and recognition of diseases is an important step toward improving crop yields. In this paper, we used a two-stream deep fusion architecture, employing CNN-SVM and CNN-Softmax pipelines, along with an inference model to infer the disease classes. A dataset of 2284 images was sourced from primary (using a consumer RGB camera) and secondary sources (the internet). The dataset contained images of nine eggplant diseases. Experimental results show that the proposed method achieved better accuracy and lower false-positive results compared to other deep learning methods (such as VGG16, Inception V3, VGG 19, MobileNet, NasNetMobile, and ResNet50).

Keywords: eggplant disease; deep learning; transfer learning; recognition; leaf blight; fruit rot disease

check for updates

Citation: Haque, M.R.; Sohel, F. Deep Network with Score Level Fusion and Inference-Based Transfer Learning to Recognize Leaf Blight and Fruit Rot Diseases of Eggplant. *Agriculture* **2022**, *12*, 1160. https://doi.org/10.3390/agriculture12081160

Academic Editors: Nen-Fu Huang and Ho-Hsien Chen

Received: 29 June 2022
Accepted: 3 August 2022
Published: 4 August 2022

1. Introduction

Eggplant (*Solanum melongena*) is a popular vegetable crop. Plant health is important for the yield of nutrient-rich eggplants. However, eggplant plants and crops can be attacked by various diseases and pests, such as *Cercospora melangenae*, bacterial wilt, aphids, anthracnose fruit rot, *Alternaria* rot, collar rot, damping off, phytophthora blight, *Phomopsis* blight and fruit rot, leaf spot, and mosaic [1]. Despite continuous efforts to combat diseases in different ways, e.g., disease-tolerant varieties and targeted pest and disease control using selective chemicals, many diseases cause significant crop losses.

An automated eggplant disease diagnostic system could provide information for the prevention and control of eggplant diseases. Several deep learning and imaging-based techniques have been proposed in the literature. In recent years, deep convolutional neural networks (CNNs) have been applied extensively in image detection and classification tasks [2–5]. Deep learning-based high performing object detection and classification models include R-CNN [6], fast R-CNN [7], YOLO family [8], faster R-CNN [9], SSD (single-shot multi-box detector) [10], and R-FCN (region-based fully convolutional network) [11]. Due to its significant impact and outstanding classification performance, the use of deep learning in the agricultural field, especially in agricultural image processing areas increased tremendously over the years, e.g., weed detection [12], frost detection [13], pest detection [14], agriculture robot [15], and crop disease [16–19]. However, there is a problem concerning the availability of large datasets with reliable ground truth, which is needed to build a good predictive model with high predictive performance [20,21]. It is also the same for the plant disease dataset. Table 1 shows an overview of different approaches used in eggplant disease recognition.

Table 1. Summary of existing works on eggplant disease recognition.

Reference	Models	#Images	#Classes	Accuracy	Main Weakness
Aravind et al. [22]	CNN, VGG16, AlexNet	643	5	93.3%	It performs worse in illumination and color variations.
Maggay et al. [23]	CNN, MobileNetV2	2465	5	>80%	The quality of the captured images is of low resolution and the accuracy is relatively low.
Xie et al. [24]	KNN and Adaboost	157	2	88.46%	HLS images performed more poorly than other types of images.
Ma at al. [25]	MSC and SG pretreatment method, principal component analysis	40	2	90%	It works well only in normal conditions.
Sabrol et al. [26]	GLCM matrix, ANFIS-based classification model	520	4	98.0%	The dataset is too small and classifies only four categories of diseases.
Wu et al. [27]	Backpropagation neural networks, principal component analysis	220	2	85%, 70%	It works with only two categories and low accuracy. Moreover, the method requires calibration sample sets which have spectra and corresponding quality attributes. However, it is impractical to obtain the reference values of attributes for every pixel using reference analysis.
Krishna- swamy et al. [28]	VGG16, MSVM	1747	5	99.4%	It achieves high accuracy on RGB images but low performance in other color spaces (HSV, YCbCr, and grayscale).

Aravind et al. (2019) worked on the classification of *Solanum Melongena* using transfer learning (VGG16 and AlexNet) and achieved an accuracy of 96.7%. However, their accuracy dropped sharply with illumination and color variations [22]. Maggay et al. (2020) proposed a mobile-based eggplant disease recognition system using image processing techniques. However, their accuracy was around 80% to 85% [23], which is rather low. Xie [24] worked on early blight disease of eggplant leaves with spectral and texture features using KNN (K-nearest neighbor) and AdaBoost methods. This work demonstrated that spectrum and texture features are effective for early blight disease detection on eggplant leaves and achieved an accuracy over 88.46%. However, they reported that HLS images performed a little worse, while other types of images achieved better results.

Wei et al. [25] used infrared spectroscopy to identify young seedlings in eggplant infected by root knot nematodes using multiple scattering correction (MSC) and Savitzky–Golay (SG) smoothing pretreatment method. Their method gave 90% accuracy but worked well only in normal conditions. Sabrol et al. proposed a GLCM matrix to compute the features and then an ANFIS algorithm to classify the diseases of eggplant, providing 98.0% accuracy, but their image dataset was too small and classified only four categories of diseases [26]. Wu et al. proposed visible and near-infrared reflectance spectroscopy for early detection of the disease on eggplant leaves before the symptoms appeared. They used principal component analysis along with backpropagation neural networks and achieved an accuracy of 85% [27]. Krishnaswamy et al. applied pretrained VGG16 and multiclass SVM to predict different eggplant diseases. They used different color spaces (RGB, HSV, YCbCr, and grayscale) and gained the highest accuracy (99.4%) using RGB images [28]. However, there has been limited work on eggplant disease recognition from images. Hence, the present paper tries to fill this gap by developing a deep CNN with a transfer learning-based automated system. In addition, there is no publicly available benchmark dataset for eggplant disease classification.

To fill this gap, an eggplant disease dataset was prepared in this work with nine (existing works cover a maximum of five classes) disease classes. A transfer learning with pretrained CNN models (with fine-tuning of the structure and hyperparameter optimization) was applied to recognize the diseases. In addition, we employed a fusion of CNN-SVM and CNN-Softmax strategies with the inference of the best three deep CNN models. In this study, nine diseases (namely, aphids, bacterial wilt, *Cercospora melongenae*, collar rot, eggplant–Colorado potato beetle, little leaf disease, spider mites, *Phomopsis* blight, and tobacco mosaic virus) of eggplant were considered, and we verified the effectiveness of the method. The key contributions of this paper are summarized as follows:

- The (L, n) transfer feature learning method is applied before retraining with six pre-trained transfer learning networks with diverse settings.
- A deep CNN based on adaptive learning rate is implemented using the trend of the loss function.
- Score level fusion using the CNN-SVM and CNN-Softmax methods, as well as an inference model, is developed from the best three pretrained CNN models.
- A dataset containing the nine most frequent eggplant diseases in Bangladesh (with mostly tropical climate) is developed.

2. Materials and Methods

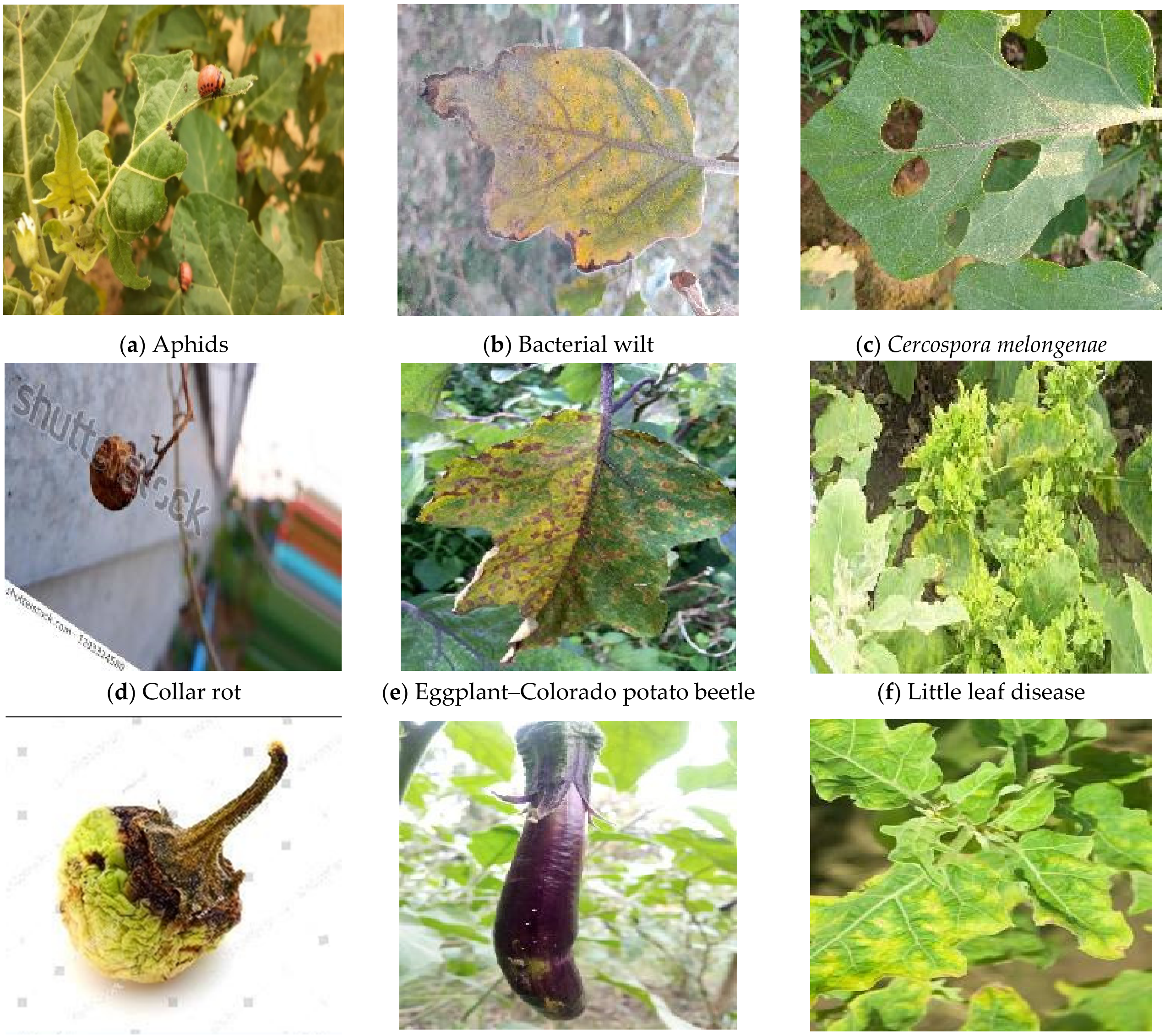
2.1. Dataset Description

The dataset consisted of 2284 images of nine disease classes. Out of these, 1842 images were collected from different areas in Bangladesh. The images were taken at different locations and situations using a Nikon D7200 DSLR camera with an 18–140 mm lens. The images were collected from November 2020 to May 2021. Each image contained three color channels: red, green, and blue. In addition, 442 images were downloaded from the internet. Since these images were collected from a variety of sources and certain types of diseases were more frequent than others, the number of samples per category was highly unbalanced. Moreover, only infectious images were collected for this study. This dataset consisted of images of aphids (class 0), bacterial wilt (class 1), *Cercospora melongenae* (class 2), collar rot (class 3), eggplant–Colorado potato beetle (class 4), little leaf disease (class 5), spider mites (class 6), *Phomopsis* blight (class 7), and tobacco mosaic virus (class 8). All the images were automatically cropped and preprocessed. During pre-processing, image resizing and normalizing operations were conducted. The images were resized to 224 × 224 during training and testing. The breakdown of the number of images used in this study is given in Table 2.

Table 2. Statistics of the eggplant disease dataset.

Class Level	Class Name	#Images
0	Aphids	103
1	Bacterial wilt	104
2	*Cercospora melongenae*	320
3	Collar rot	300
4	Eggplant–Colorado potato beetle	226
5	Little leaf disease	285
6	Spider mites	232
7	*Phomopsis* blight	266
8	Tobacco mosaic virus	448
	Total	2284

Among the images, 1924 were (84%) used for training and 360 (16%) were used for testing. The training dataset was further augmented to enrich the training dataset, so that it was more balanced and gave a better performance. We applied random rotations, shifts, flips, scales, and cropping while augmenting data. This helped reduce biases and mitigated overfitting problems while training the network. After augmentation, the total number of training images became 9620. They were divided into training and validation sets with a ratio of 8:2. Figure 1 shows sample eggplant images from the database.



(**a**) Aphids (**b**) Bacterial wilt (**c**) *Cercospora melongenae*

(**d**) Collar rot (**e**) Eggplant–Colorado potato beetle (**f**) Little leaf disease

(**g**) Spider mites (**h**) *Phomopsis* blight (**i**) Tobacco mosaic virus

Figure 1. Sample images from our eggplant disease dataset.

2.2. Method

In this section, we propose the two stream deep fusion architecture for the classification of eggplant diseases. The first stream is called feature extraction by deep CNN with transfer learning (Inception V3, VGG16, VGG19, ResNet50, MobileNet, and NasNetMobile), which can extract features from images using the preprocessed RGB images as input to the network. The second stream is called fusion along with inference and recognition, which uses CNN-SVM and CNN-Softmax strategies to fuse and predict the classes, together with the inference method through the greedy selection approach for leaf blight and fruit rot disease recognition. The overall framework of our proposed method is shown in Figure 2. As shown in Figure 2, our proposed architecture includes four parts, as described in the remainder of this section.

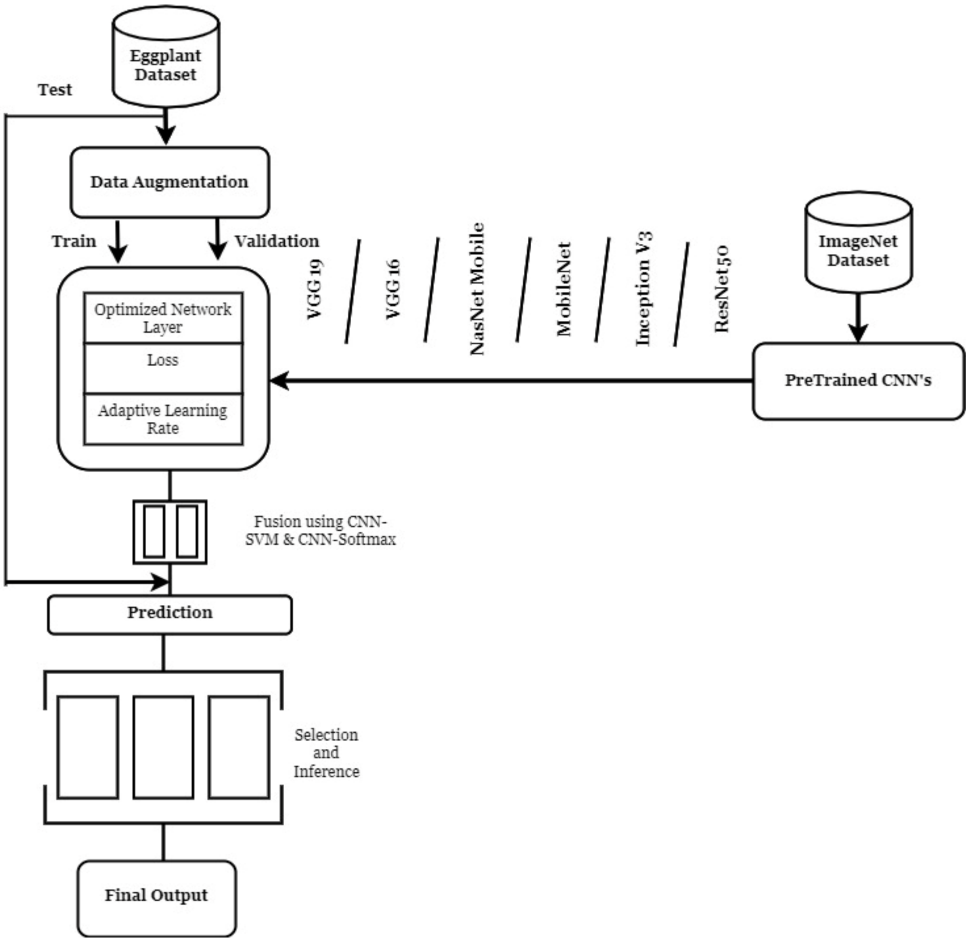


Figure 2. Schematic diagram of the proposed system.

2.2.1. (*L*, *n*) Transfer Feature Learning (TFL)

In this work, Inception v3, MobileNet, ResNet50, NasNetMobile, VGG16, and VGG19 pretrained models were evaluated. Usually, the last fully connected layer is modified in transfer learning methods and then used to retrain the whole network or retrain the modified layer. Here, the (*L*, *n*) transfer feature learning algorithm was applied, where (i) the number of layers *L* to be removed (NLR) and the value of *L* were optimized, and (ii) *n* fully connected layers, dropout layers, and regularizers were added to improve the performance. The pseudo-code of the TFL is presented in Algorithm 1.

Algorithm 1. (*L*, *n*) transfer feature learning

```
Step I : Read a pretrained network, $N_0$
Step II : Remove the last L − learnable layers from $N_0$ and get $N_{n1}, N_{n2} = F_n(N_0, L)$,
Step III : Add n new fully connected layers, $N_{n2} = F_l(N_{n1}, \mathrm{n})$,
Step IV : Freeze early layers, $lr \rightarrow [N_{n2}\ (1{:}\ L_0 - L)] \leftarrow 0$,
Step V : Retrain the last n layers, $lnr \rightarrow N_{n2}\ (L_n - n : L_n)] \leftarrow n$
Step VI : Retrain the whole network and extract the features, $f_N = F\ (N_{n2}, L_n - 1)$.
```

2.2.2. Adaptive Learning Rate

The learning rate is one of the most important hyperparameters for tuning neural networks; it plays a vital role in the convergence to minima of a model. However, choosing a learning rate that is too small results in a long time needed to converge the neural network or it becoming stuck in an undesirable local minimum. Moreover, too high a learning rate leads to a risk of overfitting, which occurs when we take a step that is too large in the direction of the minimized loss function. Hence, choosing a proper learning rate is crucial, which is a difficult task. Different algorithms are proposed for finding a suitable learning rate but none of these are suitable for a generic task. Thus, an adaptive learning rate strategy was applied here according to the values of the training and validation loss to overcome the above constraints, thus improving the accuracy. The steps of the adaptive learning rate strategy were as follows:

1. If the training loss and validation loss are almost fixed (indicating a condition of overfitting), then the learning rate is decreased (until a threshold value); when it reaches the value, the learning rate is set to the initial value.
2. In other situations, no change in learning rate is applied.

The pseudo-code of the proposed algorithm (Algorithm 2) is presented below.

Algorithm 2. Adam with optimized learning rate

```
Require: decay rate, $\alpha$, average, $\underline{av}$
Require: training loss difference, lt1, lt2, lt3, . . . , ltN, and validation loss difference, lv1, lv2, lv3, . . . , lvN, no. of epochs, T
Require: decay-factor, $\alpha$, threshold, $\eta$
Require: initial value for parameters
    $Lr = 0.001$
Step I:     FOR i = 1 to T do
Step II:        Compute loss difference and store in a list, Lt[] and Lv[]
Step III:       Accumulate loss difference, L over 10 consecutive epochs and calculate average,
$\underline{av_v} = \frac{1}{10}\sum_{i=1}^{10} L_v[i]$ and $\underline{av_t} = \frac{1}{10}\sum_{i=1}^{10} L_t[i]$
Step IV:            IF $\underline{av_v} < \eta$, $\underline{av_t} < \eta$, and $L_r > 0.00001$ do
Step V:     $\Delta l = Lr^{*}\alpha$
Step VI:          END IF
Step VII:         Update learning rate, $Lr_{t+1} = Lr_t + \Delta l$
Step VIII: END FOR
```

2.2.3. Score Level Fusion by CNN-Softmax and CNN-SVM

The softmax classifier classifies multiple classes on the basis of the softmax function, defined as follows:

$$v_j = \frac{e^{u_j}}{\sum_{k=1}^{K} e^{u_k}}, \tag{1}$$

where $j = 1, \ldots, K$. The softmax function takes as input a vector u_j of K real numbers and normalizes (0 and 1) it into a probability distribution into another vector of real values v_j. It increases the separation among classes while mapping high-dimensional data samples to a lower-dimensional domain.

The support vector machine (SVM) is a linear model for classification and regression problems. Given a set of k training samples denoted by $(X_k,\ Y_k)$, where $k = 1, 2, \ldots, K$, and $X_k = X_{1k},\ X_{2k},\ \ldots, X_{ik}$ corresponds to the attribute set for the k-th sample, the decision boundary of the linear machine is defined as follows:

$$W^T X + b = 0, \tag{2}$$

where W is the weight vector, and b is a bias term.

However, it can be extended to a multiclass classification called the one-vs,-rest strategy. For n class problems, n linear SVMs are trained independently, where the data from the other classes form the negative values. The output and predicated class are calculated as follows:

$$Output_n\ (X) = W^T X. \tag{3}$$

$$arg_n maxOutput_i(X). \tag{4}$$

In score level fusion, the feature values extracted from each method are fused together using CNN-Softmax and CNN-SVM to predict the classes. Recall that CNN is mainly used to calculate the probability score for each class while classifying test images. The classifier constructs a regression function $S(Z_i\ )$ that allows one to handle $0, \ldots,\ k$ classes, as shown in Equation (1), which calculates the score for each category or class, and their sum is always 1.

$$Probability(Z_i) = (S(Z_i))_j = \frac{e^{Z_{ij}}}{\sum_{p=0}^{k}\ e^{Z_{ip}}}, \tag{5}$$

where Z_i is the logit vector. To recognize the category of a given test image, weight is defined on the basis of the obtained predicted score. To calculate the weight for each test image, a threshold value is set to 0.5 by a heuristic approach that indicates if the score passes the threshold, in which case the weight is 1; otherwise, it is 0.

A good score fusion technique is expected to maximize the genuine scores and minimize the false-positive scores in order to have a higher genuine acceptance rate (GAR) and lower false acceptance rate (FAR). Toward this aim, the match scores generated by two different matches (CNN-Softmax and CNN-SVM) were combined by the sum rule to obtain a new match score which was then used to make the final decision. In our case, we utilized the weighted match score sum rule to obtain the fused score. The match score fusion scheme combined the matching scores from two different classifiers. If the fused score, *FS*, was greater than or equal to a specified threshold (τ), the image was categorized as a correct class; otherwise, it was categorized as an impostor. After that, for the robustness of the accuracy, the three best pretrained models were selected and inferenced.

2.2.4. Inference of the Pretrained Models by Selection

The idea was to select the best three pretrained models using a greedy selection strategy, and then the inference method was applied to determine the final decision according to the majority of experts.

- Selection Algorithm

Dataset *D* is split into a training set *D1*, a validation set *D2*, and a test set *D3*, resulting in ∣*D1*∣+ ∣*D2*∣+ ∣*D3*∣ = ∣*D*∣. The greedy selection algorithm uses *D2* to create a performance rank list, choose the best three pretrained models from that list, and ensemble them for better accuracy.

- Inferencing Method

There are two possible techniques to combine CNNs. The first one is to extract features using multiple CNNs and combine them for recognition tasks with complicated algorithms. The second one is to integrate model predictions using a mathematical technique such as inferencing that is simple, fast, and reliable. A convenient ensemble system can be generated by integrating the predictions of different models. In other words, an ensemble method accepts several expert opinions to determine the final decision. As the ensemble systems generally use various voting techniques, some of which might be misleading, probabilistic approaches are normally preferred.

Here, the three best fused models (from six models) were selected for computing the output probability, and if two of the models could recognize an object of a certain category, then it was counted as recognized; otherwise, it was not. Compared to naïve averaging, majority voting was less sensitive to the output from a single network.

2.3. Evaluation Metrics

The performance of the method was evaluated by several statistical parameters in the study such as accuracy, F1-Score, specificity, sensitivity, training and testing loss, and testing time, which are popular metrics for evaluating recognition methods. The recognition accuracy is a key indicator (a higher accuracy indicates better performance by a recognizer).

$$Accuracy = (TP + TN)/(TP + TN + FP + FN), \quad (6)$$

where TP, FP, TN, and FN are the numbers of true positives, false positives, true negatives, and false negatives in the detection results, respectively.

The sensitivity measures the prediction ability of models and is mainly used to select the instance of a certain class from a dataset, which is calculated as follows:

$$Sensitivity = TP/(TP + FN). \quad (7)$$

Specificity is the ability of a model to select an individual as 'disease'-free, which is calculated as follows:

$$Specificity = TN/(TN + FP). \quad (8)$$

Moreover, to investigate the performance of the feature extraction by applying transfer feature learning (L, n) and an adaptive learning rate, the *t*-distributed stochastic neighbor embedding (t-SNE) method was employed, which has proven to be an effective qualitative indicator [29]. In our work, two-dimensional space was selected for mapping to visualize the features.

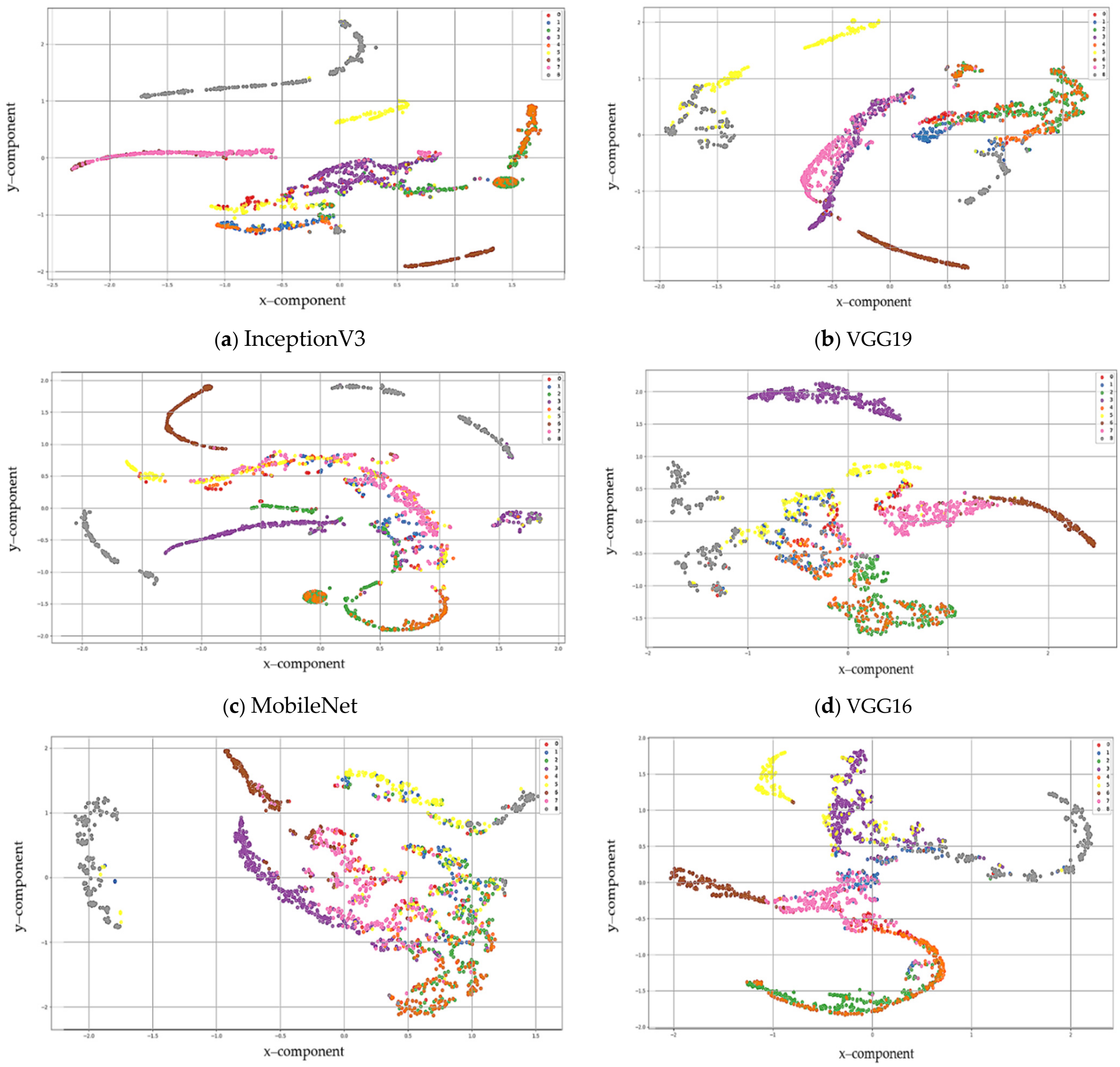
2.4. Experimental Setup

When training the CNNs (Inception V3, VGG16, VGG19, ResNet50, MobileNet, and NasNetMobile), we used the Adam optimizer with an adaptive learning rate, a mini-batch size of 32, and an initial learning rate of 0.001. The learning rate was changed by the adaptive learning rate rule given in Section 2.2.2. Our implementation was derived from the Keras library that uses TensorFlow in the background. The hardware used for carrying out the experiment had the following configuration: operating system, Windows 10 64 bit OS; graphics card, 4 GB NVIDIA GTX 1050Ti; random access memory (RAM), 16 GB.

3. Results

To investigate the performance of the six feature extraction methods after applying transfer feature learning (L, n) and an adaptive learning rate, the t-SNE method was used to visualize the feature maps. Figure 3 presents the maps of the features. The maps in Figure 3a–c indicate that classes were almost separable in the two-dimensional space as compared to other methods. The results suggest that the features extracted using Inception V3, VGG19, and MobileNet were more discriminative than other models. Hence, these three models were selected for fusion and inference to generate the final output.

To explore the effect of score level fusion by CNN-Softmax and CNN-SVM from the features extracted by deep CNNs, we conducted a comparative experiment and quantitative analysis in terms of accuracy. The individual accuracy (top three models: Inception V3, MobileNet, and VGG19) achieved after applying score level fusion is shown in Tables 3–5. We obtained 96.11%, 93.74%, and 92.11% accuracy for Inception V3, MobileNet, and VGG19, respectively. In addition, the confusion matrices of Tables 3–5 (generated after the fusion of CNN-Softmax and CNN-SVM) represent the interclass variability in disease recognition accuracy of the three selected CNNs. In all cases, the three least classified diseases were *Cercospora melongenae*, eggplant–Colorado potato beetle, and little leaf disease. The accuracy of these three classes was approximately 5–10% lower than the mean accuracy. Here, the recognition accuracy of *Cercospora melongenae* and eggplant–colorado potato beetle was lower due to their similar structure and features. Furthermore, the size and the shape of the leaves, variation in illumination, and poor resolution may have been the reason for the lower accuracy when classifying little leaf disease.



(**a**) InceptionV3

(**b**) VGG19

(**c**) MobileNet

(**d**) VGG16

(**e**) ResNet50

(**f**) NasNetMobile

Figure 3. t-SNE feature maps of different models.

Table 3. Confusion matrix for MobileNet (CNN-Softmax and CNN-SVM).

		Predicted Class								
		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Actual Class	Class 0	40	0	0	0	0	0	0	0	0
	Class 1	0	40	0	0	2	0	0	0	0
	Class 2	0	0	31	0	0	0	0	0	0
	Class 3	1	3	0	40	0	0	0	5	1
	Class 4	0	2	10	0	33	0	0	1	0
	Class 5	0	0	0	0	0	30	0	0	0
	Class 6	0	0	2	0	0	0	38	0	0
	Class 7	0	0	0	0	0	0		40	0
	Class 8	0	0	0	0	0	0	0	0	40
Average Accuracy: 92.22%										

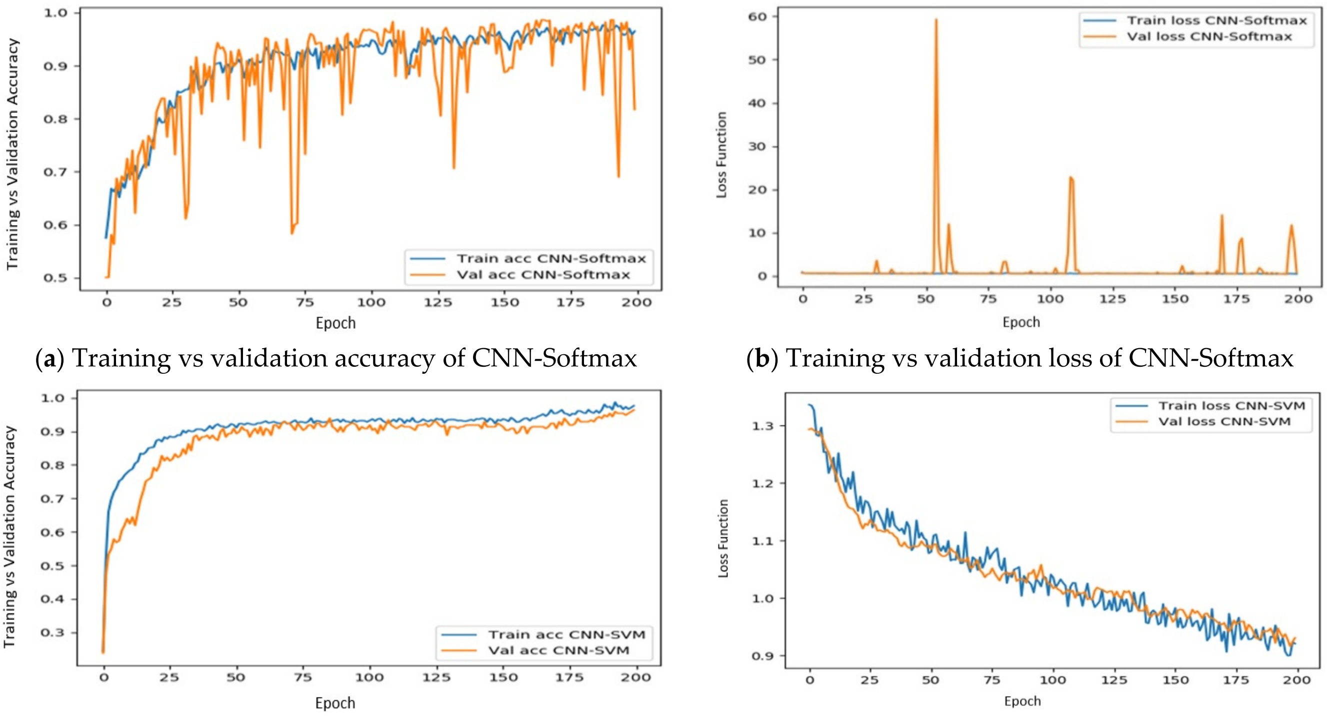
Table 4. Confusion matrix for Inception V3 (CNN-Softmax and CNN-SVM).

		Predicted Class								
		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Actual Class	Class 0	40	0	0	0	0	0	0	0	0
	Class 1	0	40	0	0	2	0	0	0	0
	Class 2	1	0	37	0	0	0	0	0	0
	Class 3	0	0	1	39	0	0	0	5	1
	Class 4	0	2	12	0	36	0	0	1	0
	Class 5	0	9	0	0	0	35	0	0	0
	Class 6	0	0	2	0	0	0	40	0	0
	Class 7	0	0	0	0	0	0		40	0
	Class 8	0	0	1	0	0	0	0	0	39
Average Accuracy: 96.11%										

Table 5. Confusion matrix for VGG19 (CNN-Softmax and CNN-SVM).

		Predicted Class								
		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Actual Class	Class 0	38	0	0	0	0	0	0	0	0
	Class 1	0	40	0	0	2	0	0	0	0
	Class 2	0	0	32	0	0	0	0	0	0
	Class 3	1	3	0	39	0	0	0	5	1
	Class 4	0	2	10	0	34	0	0	1	0
	Class 5	0	0	0	0	0	40	0	0	0
	Class 6	0	0	2	0	0	0	38	0	0
	Class 7	0	0	0	0	0	0		39	0
	Class 8	0	0	0	0	0	0	0	0	40
Average Accuracy: 93.88%										

To examine whether the model was overfitting or not, we plotted accuracy and loss function (training and validation) graphs. Figures 4–6 demonstrate the training accuracy and loss function during the training process for the three standard well-known transfer learning models using CNN-Softmax and CNN-SVM. It is shown that, initially, the accuracy was low; however, the accuracy gradually increased and finally reached 99% in the training samples for most cases. Moreover, the loss function also gradually decreased. Furthermore, it can be seen from the graphs that, during the training process, the training and validation accuracy was almost similar and increased gradually with the number of epochs. The same decreasing trend was followed by the loss function. In these graphs, it can be observed that CNN-Softmax performed better than CNN-SVM. In addition, Inception V3 performed better than the other CNNs.



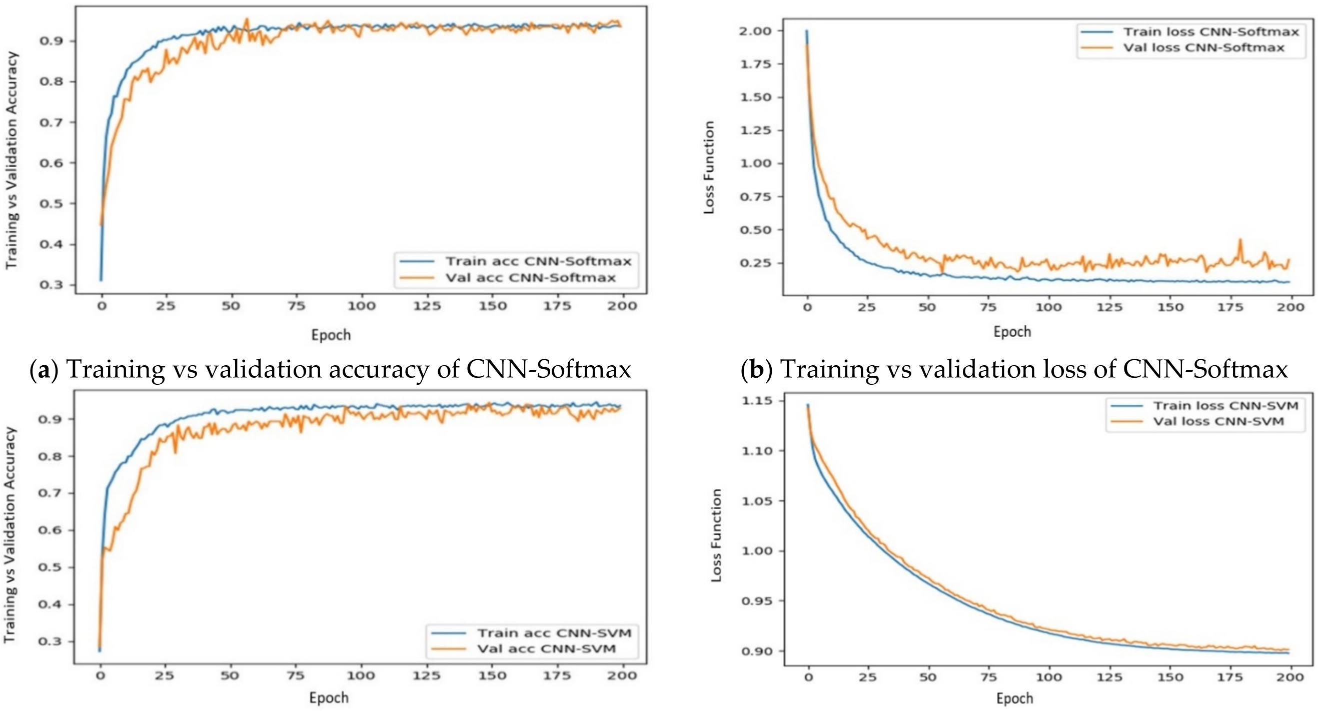
(**a**) Training vs validation accuracy of CNN-Softmax

(**b**) Training vs validation loss of CNN-Softmax

(**c**) Training vs validation accuracy of CNN-SVM

(**d**) Training vs validation loss of CNN-SVM

Figure 4. The training accuracy (**a**,**c**) and loss (**b**,**d**) after applying Inception V3 (CNN-Softmax and CNN-SVM).



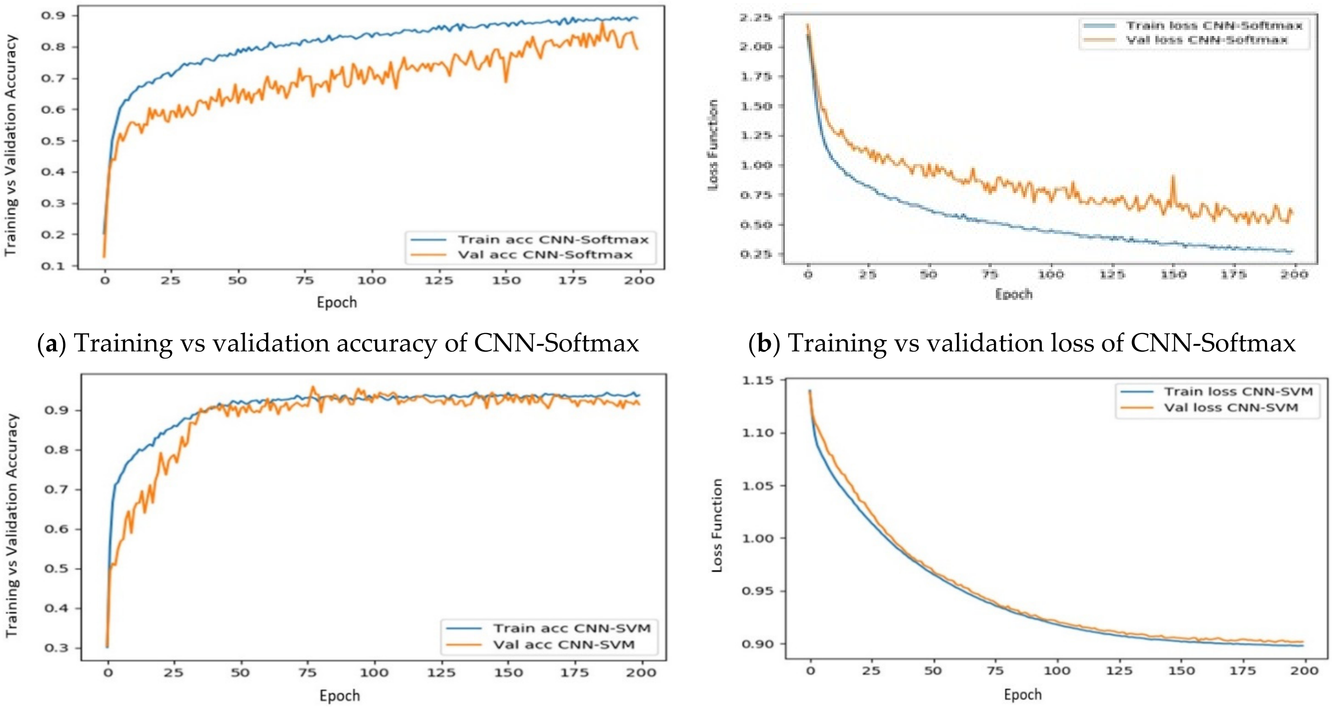
(**a**) Training vs validation accuracy of CNN-Softmax

(**b**) Training vs validation loss of CNN-Softmax

(**c**) Training vs validation accuracy of CNN-SVM

(**d**) Training vs validation loss of CNN-SVM

Figure 5. The training accuracy (**a**,**c**) and loss (**b**,**d**) after applying MobileNet (CNN-Softmax and CNN-SVM).



(**a**) Training vs validation accuracy of CNN-Softmax (**b**) Training vs validation loss of CNN-Softmax

(**c**) Training vs validation accuracy of CNN-SVM (**d**) Training vs validation loss of CNN-SVM

Figure 6. The training accuracy (**a**,**c**) and loss (**b**,**d**) after applying VGG19 (CNN-Softmax and CNN-SVM).

Moreover, the result was further improved when the fusion method was applied. Table 6 describes the evaluation result of the proposed framework and several other state-of-the-art transfer learning models using CNN-Softmax and CNN-SVM. It appears that individual accuracy was slightly improved when the fusion method was applied, and the result was additionally improved after applying the inference method. In addition, we present the F1-score, specificity, and sensitivity values of nine diseases after applying the fusion model using Inception V3, which performed best in this work and is shown in Table 7.

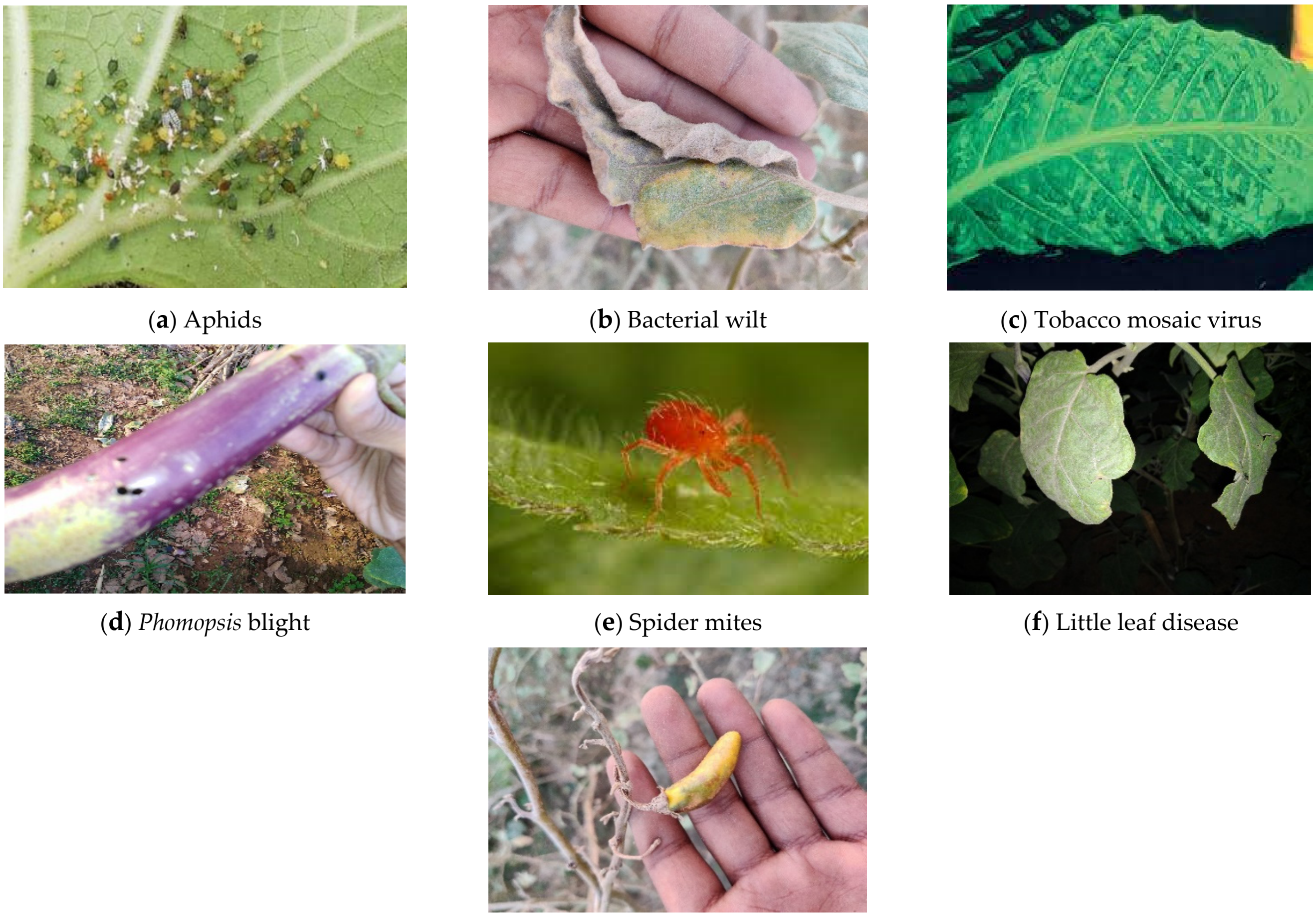
Table 6. Recognition accuracy using different deep networks.

Method	CNN-Softmax	CNN-SVM	Accuracy	
			Fusion of CNN-Softmax and CNN-SVM	Inference Method
Inception V3	**96.83%**	**92.50%**	**96.11%**	**98.9%**
VGG19	94.55%	91.34%	93.74%	
MobileNet	93.05%	86.94%	92.17%	
NasNetMobile	90.27%	86.11%	89.79%	
VGG16	90.55%	83.33%	88.77%	
ResNet50	79.16%	72.22%	78.83%	

As shown in Figure 7, we can see that some images of infected eggplants were not correctly classified by some methods due to similarities in shapes or blurry situations. Hence, when the inference rule was applied, the misclassification could be avoided easily, which improved the accuracy. Furthermore, without much surprise, methods were confused between categories with similar visual characteristics. Thus, they were more likely to be classified by only two or three CNNs and misclassified by others. The inference rule was perfectly applicable in these scenarios. As shown in Figure 7f,g, the deep learning techniques failed to correctly detect the disease due to low resolution and high noise in the data. This was mainly due to the limitations of deep networks against adversarial variations. While preprocessing of the data can address these problems to some extent, a full investigation is needed to address such problems, which will be performed in a future study.

Table 7. F1-score, specificity, and sensitivity values of nine diseases after applying the fusion model using Inception V3 (CNN-Softmax and CNN-SVM).

Class Name	F1-Score	Specificity	Sensitivity
Aphids	**98.7**	99.7	**97.6**
Bacterial wilt	86	99.3	78
Cercospora melongenae	81	99.6	69.8
Collar rot	91.7	**100**	84.7
Eggplant–Colorado potato beetle	80.8	95.5	94.7
Little leaf disease	88.6	**100**	79
Spider mites	97.5	**100**	95
Phomopsis blight	93	**100**	86.9
Tobacco mosaic virus	97.5	99.7	97.5



(**a**) Aphids (**b**) Bacterial wilt (**c**) Tobacco mosaic virus

(**d**) *Phomopsis* blight (**e**) Spider mites (**f**) Little leaf disease

(**g**) Eggplant–Colorado potato beetle

Figure 7. Examples of images of six leaf blight and fruit rot diseases of eggplant. Eggplant images in (**a**,**c**) were recognized by all deep CNNs correctly; (**e**) was recognized by InceptionV3 and MobileNet but VGG16 and ResNet50 both failed to recognize it; (**d**) was only recognized by Inception V3 and MobileNet; (**b**) was recognized after the inference method was applied. However, all three techniques failed to recognize (**f**,**g**) as they were too noisy and of low resolution.

Moreover, to check the robustness of our system, we examined the comparative performance of VGG16, InceptionV3, VGG19, MobileNet, ResNet50, and NasNetMobile methods for rotated (both clockwise and anticlockwise) images. Here, Inception V3 (9°) and MobileNet (8.5°) performed better than other transfer learning methods. The computation time of our investigated deep CNNs is shown in Table 8. We can see that the computation time of the fusion method was slightly more ample than others; however, the other algorithms were deemed acceptable when considering accuracy.

Table 8. Computation time of deep CNN models.

Model Name	Computation Time (s)
MobileNet	1.01
Inception V3	1.90
VGG19	4.24
VGG16	4.25
ResNet50	0.99
NasNetMobile	3.03
Fusion Method	4.41

4. Discussion

The identification and the recognition of plant diseases using deep learning techniques have recently made tremendous progress. They have been widely used due to their high speed and better accuracy. However, few studies were conducted on eggplant disease recognition, and these works mostly focused on the detection or recognition of a small number of disease categories [24,25,27]. To fill the gap, this paper proposed deep learning-based identification of leaf blight and fruit rot diseases of eggplant according to nine distinct categories through two steps of improvement. In the feature extraction step, transfer feature learning with an adaptive learning rate was applied to improve the accuracy, as well as to reduce overfitting. In the second step, score level fusion and inference strategies were implemented to minimize the false-positive rate and increase the genuine acceptance rate.

To check the robustness of our system, we tested in complex environments such as different rotational situations. The model showed substantial improvement over other state-of-the-art deep learning methods.

The application of the proposed system has many benefits. Firstly, rural farmers can give the right medicine for the specific disease rather than using common medicine. Secondly, the production rate can be increased, thus economically benefitting farmers. Moreover, human effort and time can be reduced. Although the method achieved high success rates in the recognition of eggplant diseases, it has some limitations. The system is relatively slow. Novel network architectures need to be investigated for time-efficient and low-resource systems (e.g., for handheld consumer devices).

5. Conclusions

In this work, we investigated popular transfer learning techniques where an inference model with score level fusion using CNN-Softmax and CNN-SVM was applied for eggplant disease recognition. In addition, we employed an optimized learning method based on the trend of the loss function to retrain the deep CNNs. Furthermore, an eggplant disease dataset was established for this particular disease recognition research. This dataset can be merged with other eggplant disease images to build a content-rich dataset, which will be useful in eggplant or even crop disease recognition research. Lastly, we demonstrated that an inference model with an optimized learning rate and early fusion was more effective. The method achieved almost 99% accuracy on the test set. Moreover, our experiments validated that our method achieved better results than other individual methods in terms of varying conditions.

Author Contributions: Conceptualization, M.R.H. and F.S.; methodology, M.R.H. and F.S.; software, M.R.H.; formal analysis, M.R.H.; investigation, M.R.H. and F.S.; resources, M.R.H. and F.S.; data curation, M.R.H.; writing—original draft preparation, M.R.H.; writing—review and editing, F.S.; visualization, M.R.H.; supervision, F.S.; project administration, F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset developed for this work is available for academic purpose at: https://github.com/reduan/Eggplant-Dataset (accessed on 20 June 2022), which will be made public upon acceptance of the paper.

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

References

1. Rotino, G.L.; Sala, T.; Toppino, L. Eggplant. In *Alien Gene Transfer in Crop Plants*; Springer: New York, NY, USA, 2014; Volume 2, pp. 381–409.
2. Rashid, J.; Khan, I.; Ali, G.; Almotiri, S.H.; AlGhamdi, M.A.; Masood, K. Multi-Level Deep Learning Model for Potato Leaf Disease Recognition. *Electronics* **2021**, *10*, 2064. [CrossRef]
3. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
4. Hafiz, R.; Haque, M.R.; Rakshit, A.; Uddin, M.S. Image-based soft drink type classification and dietary assessment system using deep convolutional neural network with transfer learning. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *34*, 1775–1784. [CrossRef]
5. Hacine-Gharbi, A.; Ravier, P. On the optimal number estimation of selected features using joint histogram based mutual information for speech emotion recognition. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 1074–1083. [CrossRef]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
7. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
11. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
12. Hasan, A.M.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G. Weed recognition using deep learning techniques on class-imbalanced imagery. *Crop Pasture Sci.* **2022**. [CrossRef]
13. Shammi, S.; Sohel, F.; Diepeveen, D.; Zander, S.; Jones, M.G.; Bekuma, A.; Biddulph, B. Machine learning-based detection of freezing events using infrared thermography. *Comput. Electron. Agric.* **2022**, *198*, 107013. [CrossRef]
14. Amrani, A.; Sohel, F.; Diepeveen, D.; Murray, D.; Jones, M.G. Insect detection from imagery using YOLOv3-based adaptive feature fusion convolution network. *Crop Pasture Sci.* **2022**. [CrossRef]
15. Li, J.; Yin, J.; Deng, L. A robot vision navigation method using deep learning in edge computing environment. *EURASIP J. Adv. Signal Process.* **2021**, *1*, 1–20. [CrossRef]
16. Hasan, M.; Zahan, N.; Zeba, N.; Khatun, A.; Haque, M.R. A Deep Learning-Based Approach for Potato Disease Classification. In *Computer Vision and Machine Learning in Agriculture*; Springer: Singapore, 2021; pp. 113–126.
17. Xin, M.; Wang, Y. Image recognition of crop diseases and insect pests based on deep learning. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5511676. [CrossRef]
18. Habib, M.T.; Majumder, A.; Jakaria, A.Z.M.; Akter, M.; Uddin, M.S.; Ahmed, F. Machine vision based papaya disease recognition. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 300–309. [CrossRef]
19. Bhargava, A.; Bansal, A. Fruits and vegetables quality evaluation using computer vision: A review. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 243–257. [CrossRef]
20. Hasan, A.M.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G. A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric.* **2021**, *184*, 106067. [CrossRef]

21. Shammi, S.; Sohel, F.; Diepeveen, D.; Zander, S.; Jones, M.G. A survey of image-based computational learning techniques for frost detection in plants. *Inf. Process. Agric.* **2022**. [CrossRef]
22. Aravind, K.R.; Raja, P.; Ashiwin, R.; Mukesh, K.V. Disease classification in Solanum melongena using deep learning. *Span. J. Agric. Res.* **2019**, *17*, e0204. [CrossRef]
23. Maggay, J. Mobile-Based Eggplant Diseases Recognition System using Image Processing Techniques. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9*, 182–190. [CrossRef]
24. Xie, C.; He, Y. Spectrum and image texture features analysis for early blight disease detection on eggplant leaves. *Sensors* **2016**, *16*, 676. [CrossRef] [PubMed]
25. Ma, W.; Wang, X.; Qi, L.; Zhang, D. Identification of Eggplant Young Seedlings Infected by Root Knot Nematodes Using Near Infrared Spectroscopy. In Proceedings of the International Conference on Computer and Computing Technologies in Agriculture, Dongying, China, 19–21 October 2016; Springer: Cham, Switzerland, 2016; pp. 93–100.
26. Sabrol, H.; Kumar, S. Plant leaf disease detection using adaptive neuro-fuzzy classification. In Proceedings of the science and information conference, Las Vegas, NV, USA, 25–26 April 2019; Springer: Cham, Switzerland, 2019; pp. 434–443.
27. Wu, D.; Feng, L.; Zhang, C.; He, Y. Early detection of Botrytis cinerea on eggplant leaves based on visible and near-infrared spectroscopy. *Trans. ASABE* **2008**, *51*, 1133–1139. [CrossRef]
28. Krishnaswamy, R.A.; Purushothaman, R. Disease classification in eggplant using pre-trained VGG16 and MSVM. *Sci. Rep.* **2020**, *10*, 1–11. [CrossRef]
29. Arora, S.; Hu, W.; Kothari, P.K. An analysis of the t-sne algorithm for data visualization. In Proceedings of the Conference on Learning Theory, Stockholm, Sweden, 6–9 July 2018; PMLR: New York City, NY, USA, 2018; pp. 1455–1462.