Integration of Convolutional Neural Networks and Recurrent Neural Networks for Foliar Disease Classification in Apple Trees

Disha Garg, Mansaf Alam
Computer Science Department
Jamia Millia Islamia
New Delhi, India

***Abstract*—Automated methods intended for image classification have become increasingly popular in recent years, with applications in the agriculture field including weed identification, fruit classification, and disease detection in plants and trees. In image classification, convolutional neural networks (CNN) have already shown exceptional results but the problem with these models is that these models cannot extract some relevant image features of the input image. On the other hand, the recurrent neural network (RNN) can fully exploit the relationship among image features. In this paper, the performance of combined CNN and RNN models is evaluated by extracting relevant image features on images of diseased apple leaves. This article suggested a combination of pre-trained CNN network and LSTM, a particular type of RNN. With the use of transfer learning, the deep features were extracted from several fully connected layers of pre-trained deep models i.e. Xception, VGG16, and InceptionV3. The extracted deep features from the CNN layer and RNN layer were concatenated and fed into the fully connected layer to allow the proposed model to be more focused on finding relevant information in the input data. Finally, the class labels of apple foliar disease images are determined by the integrated model for apple foliar disease classification, experimental findings demonstrate that the proposed approach outperforms individual pre-trained models.**

Keywords—Artificial intelligence; deep learning; transfer learning; convolutional neural network; recurrent neural network; LSTM

I. Introduction

Agriculture began in neolithic era and has continued to this day. However, the situation of agriculture is deteriorating day by day as a result of harsh variations in climatic conditions and human malpractice. Diseases in trees, animals, and grains are dangerous and can kill agriculture if not detected early. Various plant diseases can spoil the surrounded plants and trees. But due to a lack of knowledge and expertise, it is not possible to diagnose such diseases. As a result, experts are doing everything possible to assist farmers in overcoming these diseases.

Apple (Maluspumila) is commercially the most important fruit and is fourth among the most widely produced fruits in the world [1]. The growth of apple trees is affected by the assault of numerous foliar diseases such as scab, black rot, and rust [2]. Because of resemblance in appearance in the early stages of tree growth, correctly classifying and identifying all forms of foliar diseases in apple trees is a difficult challenge for farmers. Early disease identification is important for effective and timely disease control. Incorrect and/or delayed diagnosis may result in disease transmission because small insects can damage the whole tree and rapidly it can become a larger and more costly issue.

The manual classification of these diseases can be unnecessarily cumbersome, time-consuming, and costly. Speeding up this method has many advantages, including lower costs, reduced effort. Various plant disease classification systems are designed to support non-expert and non-botanist users in automatically distinguishing plant diseases. Because of this, cost and time can be reduced with accurate identification. Machine learning (ML) [3][4] and deep learning (DL) [5-11] have received a lot of interest in recent years for automatic disease and pest detection in trees and plants. Usama Mokhtar et al. [3] implemented SVM-based disease detection of tomato leaves. They used the dataset of 800 healthy and unhealthy leaves for disease classification. Authors performed feature extraction using Grey-Level Co-occurrence matrix (GLCM and achieved a classification accuracy of 99.83%. Shima Ramesh et al. [4] implemented plant disease detection, and they achieved the highest classification accuracy (70.14%) using the random forest technique among the various machine learning algorithms. These machine learning techniques use manual feature extraction, but deep learning algorithms solve this problem effectively.

For automatically detecting crop diseases, convolutional neural networks (CNNs) have become common. Moreover, numerous studies on the identification of apple diseases also have been published in the literature using pre-trained CNN models since 2017. Table I summarizes the contents of these articles.

The problem with these models is that these models cannot extract some relevant image features of the input image. In this paper, the performance of combined CNN and RNN models is evaluated by extracting relevant image features. For this, transfer learning was used in which the pre-trained weight parameters on the ImageNet dataset are transferred to CNN and combined with the extracted features of RNN in the model.

TABLE I. STUDIES TO CLASSIFY APPLE LEAVE DISEASE USING PRE-TRAINED CNN MODEL

Reference	Task	Dataset	Method	Accuracy	Pros and Cons
(Peng Jiang et al. 2019) [1]	Real time apple leaf diseases classification	26,377 images of apples diseased leaves	VGG-INCEP	97.14 %	Detects disease spots of various sizes on the same leaf
(Yong Zhong, Ming Zhao 2020) [12]	Identify apple leaf diseases	2462 images of 6 apple leaf diseases	DenseNet-121	93.71 %	Cannot overcome the impact of data imbalance
(Xiaofei Chao et al. 2020) [13]	Early diagnosis and accurate identification of apple tree leaf diseases	2970 images of diseased and healthy leaves	Xception and DenseNet (XDNet)	98.82 %	The diversity of dataset is not enough
(GuanWang et al. 2017) [15]	Image-Based Plant Disease Severity Estimation	50,000 images of healthy and diseased crops-38 class labels	VGG16 trained with transfer learning	90.4%	Lacking number of parameters to compare the results
(Bin Liu et al. 2017) [16]	Identify apple leaf diseases	13,689 images of diseased apple leaves	AlexNet, image generation technique	97.62 %	Image generation technique can enhance robustness of CNN model

Contributions of the CNN-RNN combined network are as follows:

- To classify leaf diseases, Pretrained CNN models and RNN were used.
- The performance of combined network was evaluated.
- The suggested integrated model performs better than pre-trained networks, according to the findings of the experiments.

The remaining part of the article is laid out as follows: Section 2 presents a review of current scholarly publications relevant to this article. Section 3 contains the methodology, dataset collection and description, data preprocessing, an integrated network, and its layers. In Section 4, the training and fine-tuning of an integrated model is described. Section 5 contains the experimental setup as well as the model classification findings. In Section 6, the proposed model is compared with existing techniques. Section 7 concludes the paper.

A. Motivation

When the current work is examined on image classification, it was observed that researchers have paid a lot of attention to combining multiple models. Deep models with multi-modality perform better in terms of classification accuracy, precision, recall, and f1 scores. JananiVenugopalan et al. used a combination of models to diagnose Alzheimer's disease at an early stage[17].Md. Zabirul Islam et al. [18] used X-ray images to combine CNN and LSTM for the identification of a novel coronavirus (COVID-19).Aydin Kaya et al. [19] used the combination of RNN and CNN models for plant classification. It is also found that deep feature concatenation is extremely efficient. This sparked the interest in combining RNN and CNN, as well as the notion of concatenating features from pre-trained CNN and RNN models.

II. RELATED WORK

Many pieces of research on the identification of plant diseases have been carried out so far. Genetic algorithms [20] [21], artificial neural networks [22], Bayes classifiers [23], and fuzzy logic [24] have been used in previous studies to identify and classify plant leaf diseases having a higher level of accuracy.

DL has been extensively researched in recent years for disease and pest identification in plants [9] [10] [25] [26]. Previous research has also shown that CNN is ideally suited for high-accuracy detection and diagnosis of plant diseases automatically. Some of these studies used CNN-based models that had been pre-trained, while others created their models based on CNN or pre-trained models.

Alvaro Fuentes et al. (2017) [5] proposed a faster region-based CNN for tomato diseases and pests recognition in real-time. Authors utilized a huge data-set for testing and training, comprising of 5000 images to develop these systems. These systems are capable of detecting nine distinct diseases and pests. The authors used a faster RCNN, but this model has a poor identification rate and a lot of pattern variability in certain disease groups.

Belal A. M. Ashqar and Samy S. Abu Naser (2018) [6] used deep learning approach to develop tomato leaves disease detection system. They used 9000 images of diseased and healthy tomato leaves for training of deep CNN to classify 5 diseases. They achieved 99.84 % accuracy. To perform detection and diagnosis of diseases in plants, Konstantinos P. Ferentinos (2018) [7] developed a convolutional neural network model using diseased and healthy 87,848 images of plant leaves. They trained various models. Among all CNN models, VGG achieved an accuracy of 99.53% with the best performance. The authors used the VGG model on diseased images and had a decent success rate, but this model is far from being a generic method.

Aravind Krishnaswamy Rangarajan et al. (2018) [8] have performed the classification of disease in tomatos by using pre-trained DL and its algorithm. Authors have used AlexNet and VGG16 pre-trained architectures on PlantVillage consisting of 13262 images and achieved 97.29% classification accuracy for VGG16 net and 97.49% for Alex-Net. However, other transfer learning models need to be evaluated for this dataset.

Additionally, Geetha ramani G. and Arun Pandian J (2019) [9] proposed a model to identify diseases in plants using Deep

CNN. For training, they used a dataset that included 39 different plant leaf classes as well as background images. Authors used the following 6 types of data augmentation methods: principal component analysis (PCA), gamma correction, flipping, noise injection, color augmentation, scaling, and rotation. They achieved a classification accuracy of 96.46% for this proposed model. Despite their high success rate, authors were unable to completely exploit important image features using deep CNN.

Moreover, Edna Chebet Too et al. (2018) [10] performed the comparison of InceptionV4, VGG16, ResNet with 101, 152, 50 layers, and Dense-Nets with layer number as 121. Among all structural designs, Dense-Nets achieved the testing accuracy of 99.75 percentages. With an increasing number of epochs, DenseNets has the potential to increase accuracy without overfitting. However, computational time can be improved.

TABLE II. PREVIOUS STUDIES BASED ON DEEP LEARNING FOR THE OTHER PLANT SPECIES

Author	**Plant species**	**Model used**	**Best Results**	**No. of images**
Mehmet MetinOzguven, Kemal Adem [27]	Sugar beet	Faster R-CNN	92.89%	155
Aravind-Krishna swamy Rangarajan, Raja Purushothaman, Aniirudh Ramesh [8]	Tomato	AlexNet VGG16 net	97.49% 97.29%	13,262
Geetharamani G., ArunPandian J [9]	Different plant disease	Deep CNN	96.46%	61,486
Karthik R et. al. [28]	Tomato	Attention embedded residual CNN	98%	95999
Sladojevic et al. [20]	Plant leaves	CNN based model	96.3%	2 589
Honghua Jiang et.al [29]	weed and crop recognition	GCN-ResNet-101	97.8%	6000
Gensheng et al. [26]	tea leaf	VGG16	90%	4 980
Ashqar et al. [6]	Tomato	Deep CNN	99.84%	9000
Jihen Amara et al. [11]	Banana	LeNet	98.61%	3700
Srdjan Sladojevic et al. [30]	Different plant disease	deep CNN	96.3%	30880
Sharada P. Mohanty et al. [31]	Different crops	GoogLeNet	99.35%	54306

One of the most significant gaps of the current studies in the field of plant disease detection is a significant decrease in classification performance [12, 15, 26] of the models on real images collected in fields compared to images taken in a controlled environment so there should be availability of large public datasets [14] [27]. Table II summarizes the findings of related investigations.

III. METHODOLOGY

The overall method for detecting foliar diseases in an apple tree dataset is represented in Fig. 1 in several phases. The apple foliar disease dataset was utilized as the training data for the CNN-RNN network. The training accuracy, validation accuracy, training loss, and validation loss were calculated at each epoch. Confusion matrix, accuracy, precision, recall, and f1-score were used to evaluate the performance of the proposed system. It is also observed the minimum training and minimum validation loss and corresponding accuracies.

A. Dataset

High-resolution and real-life symptom images of different apple foliar diseases have been manually captured with angles, illumination, noise, and surfaces. This dataset for the Plant-Pathology Challenge is made available and open to the group through “https”. It’s an integral component of FGVC (Fine-grained visual categorization)which is a workshop at CVPR(Computer -Vision and Pattern- Recognition) 2020.This dataset has four types of classes (a) Apple scab, (b) Black rot (c) Cedar apple rust (d) healthy. Sample images of each class are given in Fig. 2. Proposed model, however, is not limited to this dataset and can be used on a variety of plant disease datasets. Table III shows the count of total images in the dataset, which has previously been separated into the train set and test set.

B. Preprocessing

Dataset is enhanced to increase the model accuracy and reduce over-fitting. Apple foliar disease dataset contains RGB images of arbitrary sizes. All of the images were scaled to a resolution of 299 by 299 pixels and to be compatible with the initial values of the network, all pixel values were divided by 255.After that, sample-by-sample normalization was carried out. The efficiency of end-to-end training can be greatly improved by normalization. Finally, the training images are subjected to a variety of random augmentations, such as random rotation, shearing, zooming, and flipping.

C. Development of the Integrated Model

A proposed integrated CNN-RNN model is based on pre-trained CNN. The models mainly contain different layers of pre-trained CNN models, one RNN layer/LSTM layer and one softmax layer. The images are re-sized to fit the input layers of the pre-trained models such as Xception, VGG16, and InceptionV3. After this, deep features were extracted from these pre-trained models and then different shape of the output is found for different pre-trained models. For inceptionv3 the output shape is found as (None, 8, 8, 2048). For VGG16, the output shape is received as (None, 9, 9, 512) and for the Xception model (None, 10, 10, 2048) is received as an output shape. The input is provided to the LSTM layer using the reshape technique. After analyzing the time characteristics, a softmax layer/fully connected layer predicted class labels of the input images that were categorized into four categories: scab, black rot, cedar apple rust, healthy.

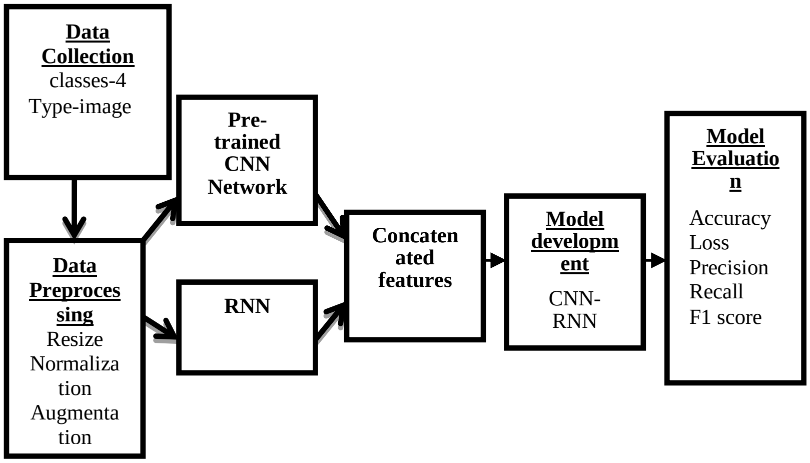


Fig. 1. Proposed System.

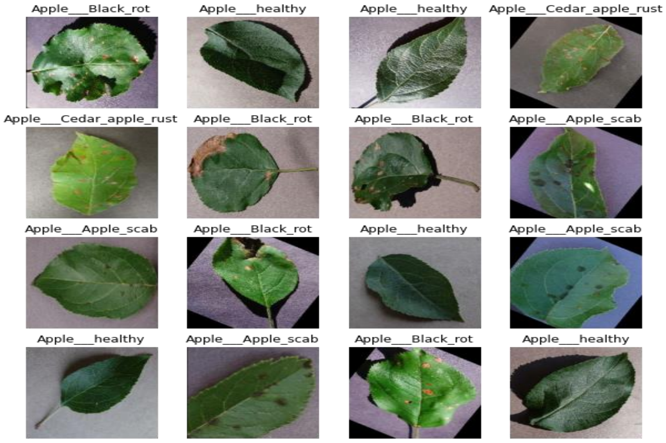


Fig. 2. The Apple Foliar Dataset Sample for Four Classes.

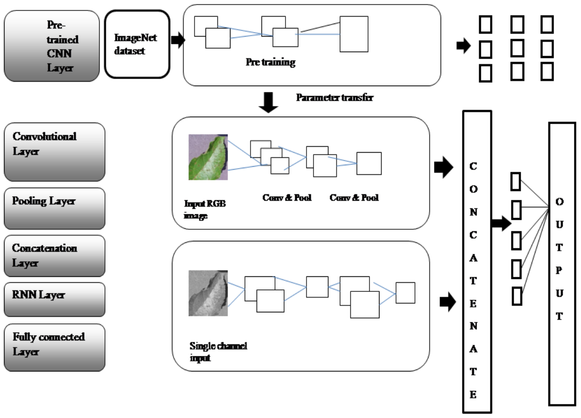


Fig. 3. Integrated Model.

TABLE III. IMAGE COUNT IN THE DATASET

Dataset	Apple scab	Black rot	Cedar apple rust	Healthy	Total
Training	4032	3974	3520	4016	15542
Testing	1008	994	880	1004	3886
Total	5040	4968	4400	5020	19428

RNNs can handle non-sequential data as well as sequential data, according to several pieces of research [32]. Improved RNN models, such as LSTMs allow for long sequence training, conquering issues such as vanishing gradients. Vanishing gradients is a problem where a deep neural network is not able to transmit valuable gradient information from the output to the input end of the model. RNN methods for sequentially processing the data of variable-length and fixed sizes have been presented in a few recent papers [33]. LSTM may be used to collect the images having discriminating regions for fine-grained classification [34]. LSTM (Long Short-Term Memory networks) has a "processor" that determines whether or not the information is useful. The Integrated model is illustrated in Fig. 3 with four layers: Pre-trained CNN layer, RNN layer, concatenation layer, softmax, or fully connected layer.

All layers of an integrated model are briefly described as follows:

1) Pre-trained CNN layer: For initializing weights of CNN model, pre-trained weight parameters are used. ImageNet is used for pre-training and it is a common dataset to develop different architectures. It is broad enough (1.2 million images) to construct a generic model. Transfer learning indicates that these pre-trained networks can generalize to images that are not part of the ImageNet dataset. By fine-tuning the model, changes can be made to it that was not previously possible.

2) Convolution layer: Convolution operations with feature maps are performed with this layer using convolution windows of various sizes. The effect of applying a convolution over an image is a feature map. The row index and column index of resulted matrix is denoted by u and v respectively. The number of weight parameters varies as the window size changes. Convolution and pooling are two types of computations that can be done in the early layers of CNN. The convolution layer has the following operation as in (1):

$$S\ (u,\ v) = (M*F)\ (u,\ v) = \sum\sum M\ (u+m, v+n)\ F\ (m,\ n) \qquad (1)$$

In (1), M indicate the input matrix. F represents the m×n size 2D filter of and S denotes the 2D feature map output. M*F denotes the convolutional layer operation. Rectified linear unit (ReLU) is the activation function which is used after the convolution operation to improve nonlinearity in feature maps. ReLU calculates activation by keeping the threshold at zero. It may be mathematically represented as in (2):

$$f(y) = \max(0, y) \tag{2}$$

3) Pooling layer: To minimize the number of parameters, this layer down samples a given input representation. By using max-pooling which is the most frequent approach, the size of resulted image can be reduced by retaining image information. As a convolution layer, the pooling layer also uses filters of different sizes.

4) RNN/LSTM Layer: An input layer, a hidden layer, and an output layer are present in RNN and CNN. The most significant aspect of RNN is the interaction of these hidden layers. The input and hidden layer nodes are linked. Output layer takes the outcome of the hidden layer. Hidden layer receives information from the output layer node. There exist various adjacent hidden layer nodes.

In this article, LSTM is used which is a form of RNN. LSTM is made up of memory blocks and these blocks are recurrently connected blocks. Each has 3 multiplicative gates (input, output, forget gate) and one or two memory cells that are recurrently connected. The input gate monitors how much data it should read, forget gate monitor whether it should forget the value of the current cell, and the output gate monitor whether it should output the new cell value.

5) Concatenation layer: The concatenation layer concatenates features derived from the CNN and RNN. Feature concatenation is a powerful technique for combining many features to improve classification accuracy.

6) Softmax layer or fully connected layer: The features of RNN and CNN are concatenated and forwarded to the softmax layer, which generates the input image class labels.

D. Long Short-term Memory (LSTM)

A cell with three gates (input, output, and forget) makes up an LSTM unit. LSTM can choose which information is lost and which is remembered. An LSTM layer is made up of memory blocks, which are recurrently connected blocks. It can be assumed that these blocks are a differentiable variant of a computer's memory chips. Every block has one or more than one recurrently linked memory cells, as well as 3 multiplicative units. These units continuously provide analogs of reading, Reset and Write operations for the cells. The following are the gate equations:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3}$$

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{4}$$

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$gt = tanh(W_i[h_{t-1}, x_t] + b_i) \tag{6}$$

i_t,$f_{t,}$ and o_t represent input gate, forget gate, and output gate respectively. σ denotes the sigmoid function. The weight for the respective gate(x) neurons is represented by W_x. The output of the previous LSTM block (at timestamp t-1) is h_{t-1}. x_t is the input at the current timestamp. bx is the bias for the corresponding gate(x). In (3), h_{t-1} and x_t are passed through the sigmoid layer and used to decide that which part of the `information is needs to be added. (6) Obtains new information when h_{t-1} and x_t are passed through the tanh layer in (8):

$$c_t = i_t g_t + f_t c_{t-1} \tag{7}$$

$$h_t = o_t \tanh(c_t) \tag{8}$$

The new and previous cell states are denoted by ct and c_{t}-$_1$, respectively.g_t is the current information. g_t and C_{t-1} are combined in (7). The final output O_t is calculated in (5) and it gets multiplied with c_t which transfers new information through the tanh layer.

IV. CNN-RNN Network Training and Fine-tuning

To classify apple foliar diseases, the combination of CNN models and RNN was used. There were a training phase and a testing phase in the proposed method illustrated. In the training process, firstly on the ImageNet dataset, CNN model was pre-trained and the parameters of this pre-trained network were used for initializing the new CNN model, this is called fine-tuning. When the entire layers of CNN have been frozen, the RNN model was trained. After the completion of training, layers of the CNN model were unfrozen and the whole CNN-RNN model was trained.

Test images that were pre-processed were fed into the CNN-RNN model. By using the softmax layer, the classification results were obtained. In this experiment, the proposed model is built on two layers: Conv and pool. These are the first few layers of pre-trained models, already trained on ImageNet dataset. In this proposed model, the above two layers were moved to the same place to allow the transmission of features. The CNN transfer learning model uses RGB input, whereas RNN uses a single-channel input. The aforementioned layers were trained on jointly on the apple foliar disease dataset. The training process was terminated after predetermined epochs.

V. Experimental Results

The proposed work is implemented in two parts:

Part 1: Implementing transfer learning using CNN

Part 2: Implementing RNN

RNN uses randomly initialized parameters while CNN uses pre-trained weight parameters. During the training phase, the gradient of the cross-entropy loss function is used to iteratively update these weights. Firstly, all layers of CNN were frozen then RNN and the last classification layer of CNN were trained. Then training samples are calculated with the help of an RMSProp optimizer. Then we defrosted all layers of CNN and the whole CNN-RNN model was trained then Adam optimizer was used to calculate the samples of training. For the whole network, 0.0001 was used as a learning rate.

Since the problem is multi-class, the categorical cross-entropy/softmax loss function is used. Keras and TensorFlow2.5.0with python 3 were used on an Intel(R) Core(TM) i5-2.9 GHz CPU for implementing models. Furthermore, we were carried out the experiments on an NVIDIA Tesla T4 GPU with 2.32 GB of RAM.

A. Evaluation Indices

The following indices were used to compare the performance of the individual and integrated systems: True Positive (TP): The class of interest was correctly categorized. True Negative (TN): Classified correctly as not the class of interest. False Positive (FP): The class of interest was incorrectly categorized. False Negative (FN): Not the class of interest was incorrectly categorized. Accuracy is represented in (9), Precision is represented in (10), Recall is represented in (11) and F1 score is represented in (12). Mathematical expression of all these metrics is represented below:

$$\text{Accuracy} = \frac{(\text{TP+TN})}{(\text{TN+FP+TP+FN})} \tag{9}$$

$$\text{Precison} \;= \frac{(\text{TP})}{(\text{TP+FP})} \tag{10}$$

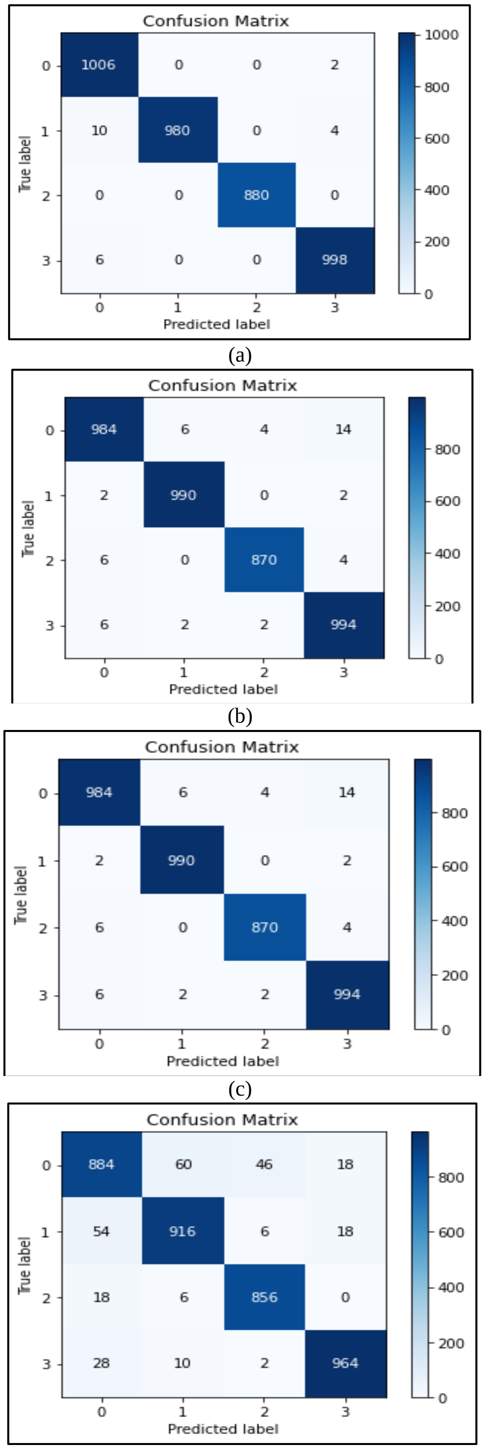
$$\text{Recall} = \frac{\text{TP}}{(\text{TP+FN})} \tag{11}$$

$$\text{F1 score} \; \frac{2*\text{TP}}{(2*\text{TP+FP+FN})} \tag{12}$$

B. Result Analysis and Discussion

Fig. 4 illustrates the confusion matrix of the pre-trained CNN model and the proposed CNN-RNN models for apple foliar disease classification. Fig. 4(a) represents Xception-LSTM, Fig. 4(b) represents Xception, Fig. 4(c) represents VGG16-LSTM, Fig. 4(d) represents VGG16, Fig. 4(e) represents InceptionV3-LSTM and Fig. 4(f) represents Inception V3. In the confusion matrix, apple_scab is represented as class 0, black_ rot is represented as class 1, cedar_ apple_ rust is represented as class 2, and healthy leaves are represented as class 3. The Xception model misclassified 44 of 3886 test images, whereas the proposed Xception-LSTM model misclassified 22 of them. Among the 3886 test images, 266 were misclassified by the VGG16 model, whereas the proposed VGG16-LSTM model misclassified 45 of them. Similarly, 60 were misclassified by the InceptionV3 model, whereas the proposed InceptionV3-LSTM model misclassified a total of 14 images. The proposed integrated CNN-RNN networks outperformed the individual pre-trained CNN networks in terms of TP and TN values, as well as reduced FN and FP values. Table IV shows the observed results of pre-trained CNN models and CNN-RNN models in terms of minimum training loss with corresponding training accuracy, and minimum validation loss with corresponding validation accuracy and Xception-LSTM accuracy and loss are depicted in Fig. 5(a) and 5(b).

Fig. 6(a) and 6(b) depict accuracy and Xception loss, respectively. VGG16-LSTM accuracy and loss are depicted in Fig. 7(a) and 7(b). VGG16 accuracy and loss are depicted in Fig. 8(a) and 8(b). InceptionV3-LSTM accuracy and loss are depicted in Fig. 9(a) and 9(b). InceptionV3 accuracy and loss are depicted in Fig. 10(a) and 10(b). Performance evaluation for pre-trained CNN models and CNN-RNN models may be seen in the loss and accuracy figures.

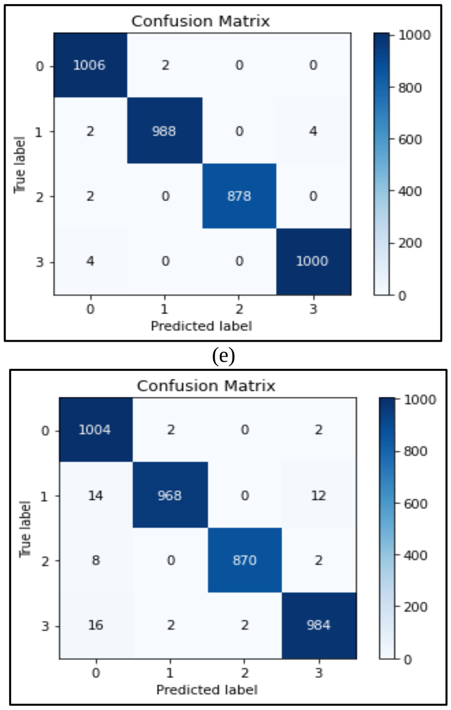


(a)

(b)

(c)

(d)



(e)

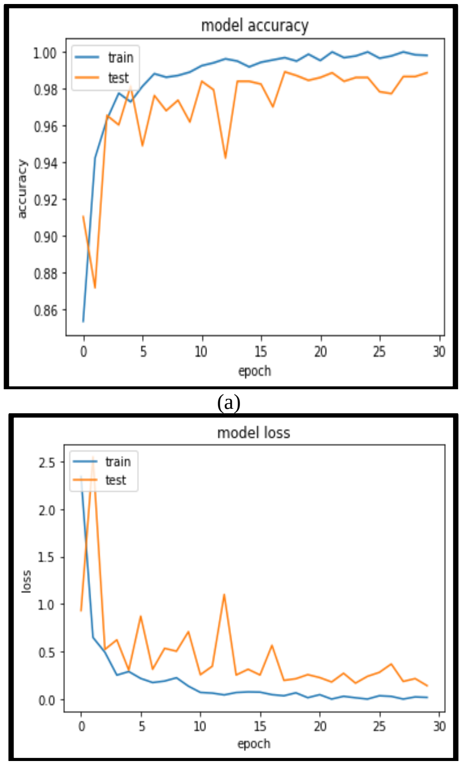
(f)

Fig. 4. (a) Xception-LSTM (b) Xception (c) VGG16-LSTM (d) VGG16 (e) InceptionV3-LSTM (f) InceptionV3.

TABLE IV. COMPARISON OF INTEGRATED MODELS AND INDIVIDUAL MODELS IN TERMS OF MINIMUM LOSS AND CORRESPONDING TRAINING AND VALIDATION ACCURACY

Model	Training		Validation	
	Loss	**Accuracy**	**Loss**	**Accuracy**
Xception	0.0144	99.87	0.1408	98.86
Xception-LSTM	0.001	99.94	0.0111	99.69
VGG16	0.0345	98.87	0.206	93.08
VGG16-LSTM	0.0028	99.94	0.0468	99.07
InceptionV3	0.0464	99.59	0.3042	98.24
InceptionV3-LSTM	0.0011	99.94	0.0027	99.9

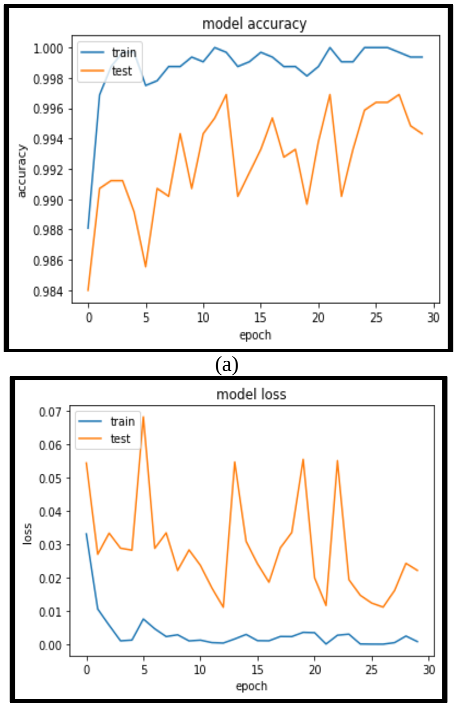
Among the total number of 30 epochs, Minimum training loss and corresponding training accuracy of the Xception were found at .0144 and 99.87% respectively at epoch 20 whereas .0010 and 99.94 for Xception-LSTM model at epoch 10. The minimum validation loss and corresponding validation accuracy of Xception model were found .01408 and 98.86% respectively at epoch 30 whereas .0111 and 99.69 for the Xception-LSTM model at epoch 13. Similarly, the minimum training loss and corresponding training accuracy of the VGG16 model were found at .0345 and 99.87% respectively at epoch 27 whereas .0028 and 99.94 for the VGG16-LSTM model at epoch 22. The minimum validation loss and corresponding validation accuracy of the VGG16 model were found at .206 and 93.08% respectively at epoch 25 whereas .0468 and 99.07 for the VGG16-LSTM model at epoch 29. Minimum training loss and corresponding training accuracy of the InceptionV3 model were found .0464 and 99.59% respectively at epoch 19 whereas .0011 and 99.94 for InceptionV3-LSTM model at epoch 12. The minimum validation loss and corresponding validation accuracy of the InceptionV3 were found .3402 and 98.24% respectively at epoch 10 whereas .0027 and 99.9 for InceptionV3-LSTM model at epoch 10.



(a)

(b)

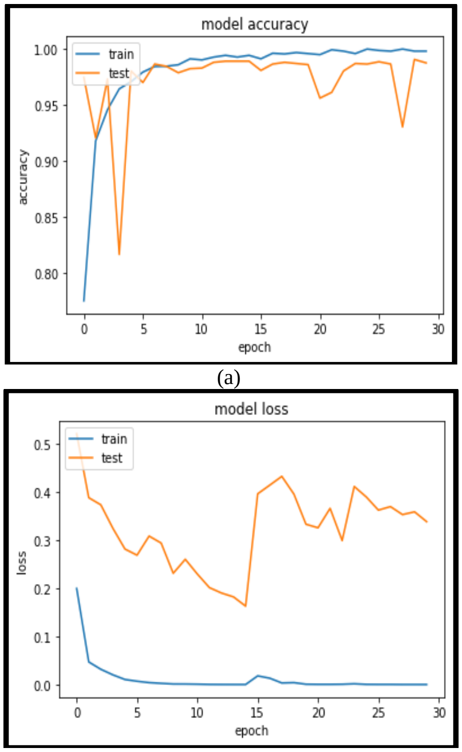
Fig. 5. (a) Accuracy (Xception-LSTM) (b) Loss (Xception-LSTM).



(a)

(b)

Fig. 6. (a) Accuracy (Xception) (b) Loss (Xception).

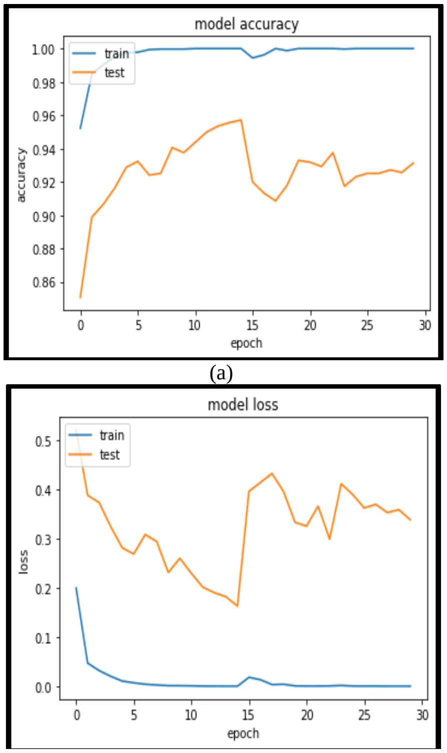


(a)

(b)

Fig. 7. (a) Accuracy (VGG16-LSTM), (b) Loss (VGG16-LSTM).

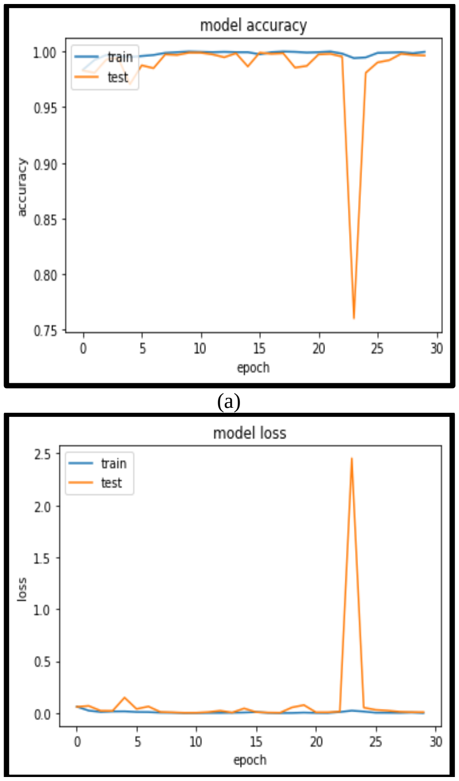
Results presented in Table IV shows that combined models have given the highest accuracy when the loss was minimum as compared to individual model. It can be seen in both the cases, training results as well as validation results.



(a)

(b)

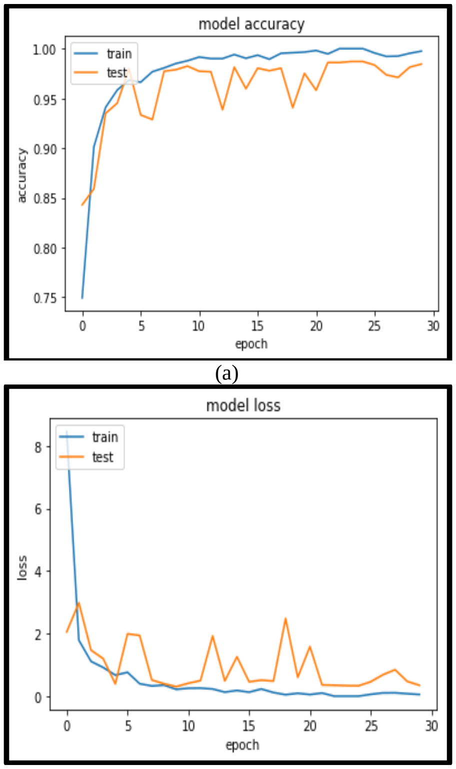
Fig. 8. (a) Accuracy (VGG16) (b) Loss (VGG).



(a)

(b)

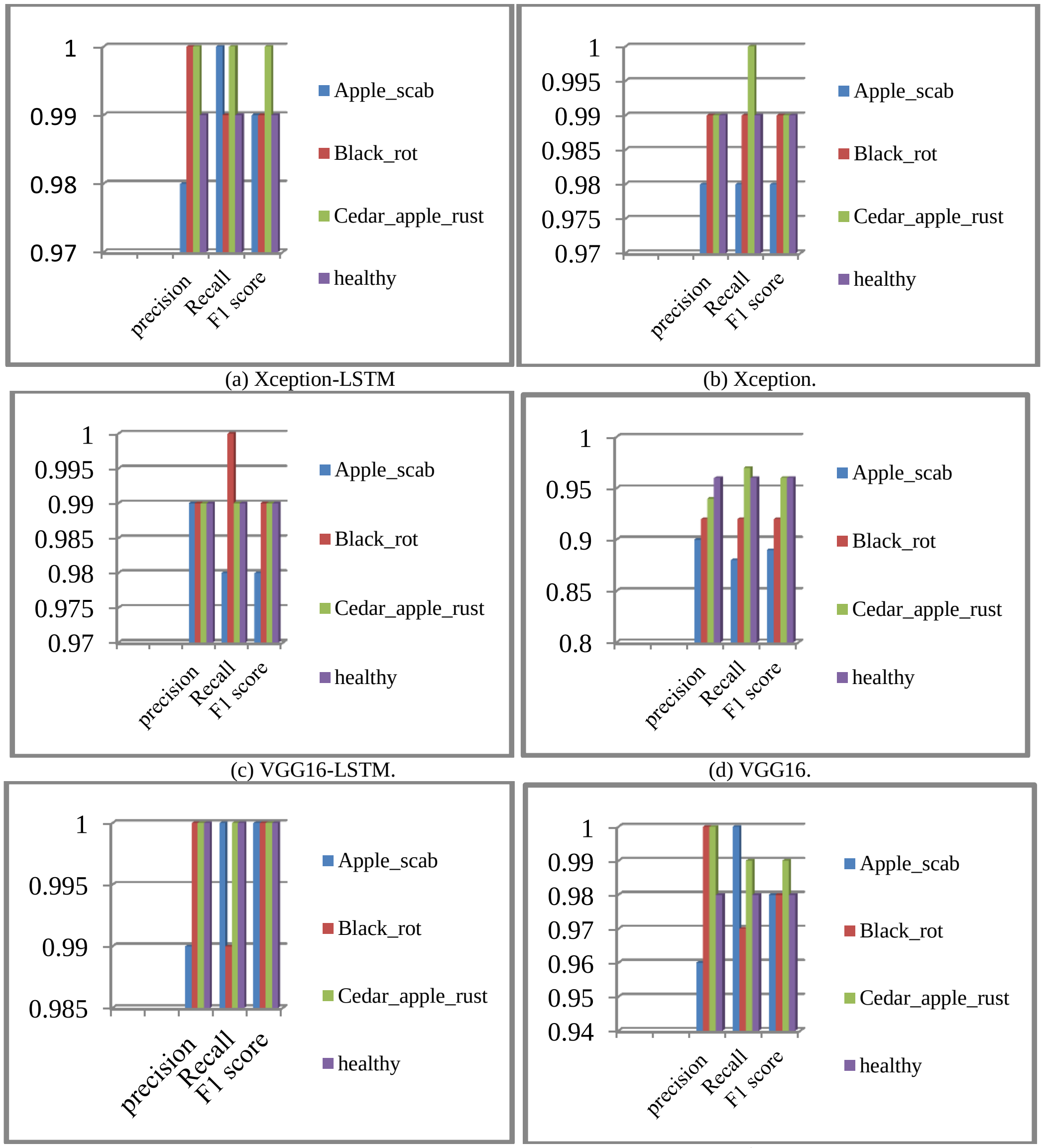
Fig. 9. (a) Accuracy (InceptionV3-LSTM) (b) Loss (InceptionV3-LSTM).



(a)

(b)

Fig. 10. (a) Accuracy (InceptionV3) (b) Loss (InceptionV3).



(a) Xception-LSTM

(b) Xception.

(c) VGG16-LSTM.

(d) VGG16.

(e) InceptionV3-LSTM.

(f) InceptionV3.

Fig. 11. Precision, Recall, and f1 Score of CNN-LSTM Models and pre-trained CNN Model.

Fig. 11 depicts precision, recall, and f1 score findings. As per the results, integrated models achieved good score in terms of precision, recall, and F1 score as compared to individual Pretrained CNN models. Xception scored 98.75% precision, 99 % recall, and 98.75% f1 score for all four classes, whereas Xception-LSTM achieved 99.25 % precision, 99.5 % recall, and 99.25% f1 score for all four classes. Similarly, VGG16 model achieved 93% precision, 93.25% recall and 93.25% recall and VGG16-LSTM model achieved 98.75% precision, 99% recall and 98.76% f1 score. InceptionV3 achieved 98.5% precision, 98.5% recall, 98.25% f1 score and InceptionV3-LSTM achieved 99.75% precision, 99.75% recall, 99.9% f1 score for all the classes.

VI. Comparison with Existing Methods

Many deep CNN architectures have recently been suggested for the classification of apple diseases. Earlier, Densenet-121, XDNet (Xception and DenseNet), VGG-INCEP, VGG16 model, AlexNet were used. For comparisons, pre-trained models were randomly selected such as Xception, VGG16, and Inceptiontv3 for apple diseases detection. As seen in Table V, the suggested approach outperformed all other pre-trained models. Among three experiments, InceptionV3-LSTM produces the highest accuracy score as 99.8%.The benefits of the suggested integrated model are as follows:

- Pre-trained CNN models were used for reducing training complexity that saved time to train a model.
- The deep features in images can be extracted and classified automatically using proposed system.
- Based on the apple foliar disease dataset, proposed approach performed the best in comparison to current approaches in terms of classification.

TABLE V. A COMPARISON OF THE PROPOSED SYSTEM TO EXISTING METHODS FOR CLASSIFYING APPLE LEAF DISEASES

Author	Deep Learning model	Overall Accuracy
(Yong Zhong, Ming Zhao 2020) [12]	Densenet-121	93.71%
(Xiaofei Chao et al. 2020) [13]	XDNet (Xception and DenseNet)	98.82%
(Peng Jiang et al. 2019) [1]	VGG-INCEP	97.14%
(Marko Arsenovic et al. 2019) [14]	Pretrained CNN models	93.67%
(Saraansh Baranwal et al. 2019) [2]	GoogLeNet	98.42%
(GuanWang et al. 2017) [15]	VGG16model	90.4%
(Bin Liu et al. 2017) [16]	Deep CNN based on AlexNet	97.62%
Proposed work	**Xception-LSTM**	**99.5%**
	VGG16-LSTM	**99%**
	InceptionV3-LSTM	**99.8%**

The main purpose of this research is to achieve good results in detecting foliar diseases in apple trees. The experimental results revealed that the proposed CNN-RNN network outperforms the individual CNN models.

VII. CONCLUSION

To create combined neural network architecture for apple foliar disease image classification, the features of CNN and RNN models are merged in this work. When comparing separate models to the integrated models, the assessment findings reveal that the integrated models generated better outcomes. InceptionV3-LSTM received the maximum accuracy score of 99.8%Second highest accuracy score of 99.5% was found for Xception-LSTM and the lowest accuracy score was found for VGG16-LSTM as 99%.Despite this performance, these integrated models may have some improvement also. Training on 30 epochs was performed. We can increase the number of epochs in the future. Another improvement can be investigating the success of these models for other agricultural applications, such as weed identification and plant classification, other plant disease detection. Other deep neural network models will be utilised in future studies to detect apple leaf illnesses in real time, such as Faster RCNN (Regions with Convolutional Neural Network), YOLO (You Only Look Once), and SSD (Single Shot MultiBox Detector). Furthermore, more forms of apple leaf illnesses and thousands of high-quality natural images of apple leaf diseases must be collected in order to identify more diseases in a timely and efficient manner.

REFERENCES

[1] Jiang, P., Chen, Y., Liu, B., He, D., & Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. IEEE Access, 7, 59069-59080.

[2] Baranwal, S., Khandelwal, S., &Arora, A. (2019, February). Deep learning convolutional neural network for apple leaves disease detection. In Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India

[3] Mokhtar, U., El Bendary, N., Hassenian, A. E., Emary, E., Mahmoud, M. A., Hefny, H., &Tolba, M. F. (2015). SVM-based detection of tomato leaves diseases. In Intelligent Systems' 2014 (pp. 641-652). Springer, Cham.

[4] Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., &Vinod, P. V. (2018, April). Plant disease detection using machine learning. In 2018 International conference on design innovations for 3Cs compute communication control (ICDI3C) (pp. 41-45). IEEE.

[5] A. Fuentes, S. Yoon, S.C. Kim, Park D.S., A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition, Sensors 17 (2022) (2017).

[6] B.A.M. Ashqar, S.S. Abu-Naser, Image-based tomato leaves disease detection using deep learning, Int. J. Acad. Eng. Res. 2 (12) (2018).

[7] K.P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Comput. Electron. Agric. 145 (2018) (2018) 311–318.

[8] A.K. Rangarajan, R. Purushothaman, A. Ramesh, Tomato crop disease classification using pre-trained deep learning algorithm, ProcediaComput.Sci. 133 (2018) (2018) 1040–1047.

[9] G. Geetharamani, P.J. Arun, Identification of plant leaf diseases using a nine-layer deep convolutional neural network, Comput. Electr. Eng. 76 (2019) (2019) 323–338.

[10] E.C. Too, L. Yujian, S. Njuki, L. Yingchun, A comparative study of fine-tuning deep learning models for plant disease identification, Comput. Electron. Agric. 161 (2019) (2019) 272–279.

[11] J. Amara, B. Bouaziz, A. Algergawy, A deep learning-based approach for banana leaf diseases classification, BTW Workshop, Stuttgart, 2017, pp.79–88.

[12] Zhong, Y., & Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. Computers and Electronics in Agriculture, 168, 105146.

[13] Chao, X., Sun, G., Zhao, H., Li, M., & He, D. (2020). Identification of Apple Tree Leaf Diseases Based on Deep Learning Models. Symmetry, 12(7), 1065.

[14] Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., &Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. Symmetry, 11(7), 939.

[15] Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. Computational intelligence and neuroscience, 2017.

[16] Liu, B., Zhang, Y., He, D., & Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry, 10(1), 11.

[17] Venugopalan, J., Tong, L., Hassanzadeh, H. R., & Wang, M. D. (2021). Multimodal deep learning models for early detection of Alzheimer's disease stage. Scientific reports, 11(1), 1-13.

[18] Islam, M. Z., Islam, M. M., &Asraf, A. (2020). A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Informatics in medicine unlocked, 20, 100412.

[19] Kaya, A., Keceli, A. S., Catal, C., Yalic, H. Y., Temucin, H., &Tekinerdogan, B. (2019). Analysis of transfer learning for deep neural network based plant classification models. Computers and electronics in agriculture, 158, 20-29.

[20] Arya, M. S., Anjali, K., &Unni, D. (2018, January). Detection of unhealthy plant leaves using image processing and genetic algorithm with Arduino. In 2018 International Conference on Power, Signals, Control and Computation (EPSCICON) (pp. 1-5). IEEE.

[21] Singh, V., & Misra, A. K. (2015, March). Detection of unhealthy region of plant leaves using image processing and genetic algorithm. In 2015 International Conference on Advances in Computer Engineering and Applications (pp. 1028-1032). IEEE.

[22] Awate, A., Deshmankar, D., Amrutkar, G., Bagul, U., &Sonavane, S. (2015, October). Fruit disease detection using color, texture analysis and ANN. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 970-975). IEEE.

[23] Kusumo, B. S., Heryana, A., Mahendra, O., &Pardede, H. F. (2018, November). Machine learning-based for automatic detection of corn-plant diseases using image processing. In 2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA) (pp. 93-97). IEEE.

[24] Cintra, M. E., Meira, C. A., Monard, M. C., Camargo, H. A., & Rodrigues, L. H. (2011, November). The use of fuzzy decision trees for coffee rust warning in Brazilian crops. In 2011 11th International Conference on Intelligent Systems Design and Applications (pp. 1347-1352). IEEE.

[25] H. Gensheng, W. Haoyu, Z. Yan, W. Mingzhu, A low shot learning method for tea leaf's disease identification, Comput. Electron. Agric. 163 (2019) (2019) 104852.

[26] Hu, G., Wu, H., Zhang, Y., & Wan, M. (2019). A low shot learning method for tea leaf's disease identification. Computers and Electronics in Agriculture, 163, 104852.

[27] Mehmet MetinOzguven a, Kemal Adem, Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithm.

[28] Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., &Menaka, R. (2020). Attention embedded residual CNN for disease detection in tomato leaves. Applied Soft Computing, 86, 105933.

[29] Jiang, H., Zhang, C., Qiao, Y., Zhang, Z., Zhang, W., & Song, C. (2020). CNN feature based graph convolutional network for weed and crop recognition in smart farming. Computers and Electronics in Agriculture, 174, 105450.

[30] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, in: Computational Intelligence and Neuroscience, Vol. 2016, Hindawi Publishing Corporation, 2016, p. 11

[31] S.P. Mohanty, D.P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Front. Plant Sci. 7 (2016).

[32] Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks.[online] karpathy. github. io. Online verfügbarunter: http://karpathy. github. Io/2015/05/21/rnn-effectiveness/[Zugriff 13 Aug. 2018]. TH Köln Gustav-Heinemann-Ufer, 54, 50968.

[33] Lee, S. H., Goëau, H., Bonnet, P., &Joly, A. (2020). Attention-based recurrent neural network for plant disease classification. Frontiers in Plant Science, 11.

[34] Zhao, B., Wu, X., Feng, J., Peng, Q., & Yan, S. (2017). Diversified visual attention networks for fine-grained object classification. IEEE Transactions on Multimedia, 19(6), 1245-1256.