Al-Nahrain Journal of Science
ANJS, Vol.24 (2), June, 2021, pp. 64-71

Classification of Plants Leaf Diseases using Convolutional Neural Network

Reem Mohammed Jasim Al-Akkam* and Mohammed Sahib Mahdi Altaei

Department of Computer Science, College of Science, Al-Nahrain University, Baghdad, Iraq

Article's Information

Received:
10-02-2021
Accepted:
10-04-2021
Published:
27-06-2021

Keywords:
Image preprocessing
Deep Learning
Convolutional Neural Network
Detecting Leaf Diseases

Abstract

Agriculture is one of the most important professions in many countries, including Iraq, as the Iraqi financial system depends on agricultural production and great attention should be paid to concerns about agricultural production. Because plants are exposed to many diseases and monitoring plant diseases with the help of specialists in the agricultural region can be very expensive. There is a need for a system capable of automatically detecting diseases. The aim of the research proposed is to create a model that classifies and predicts leaf diseases in plants. This model is based on a convolution network, which is a kind of deep learning. The dataset used in this study called (Plant Village) was downloaded from the kaggle website. The dataset contains 34,934 RGB images, and the deep CNN model can efficiently classify 15 different classes of healthy and diseased plants using the leaf images. The model used techniques to augment data and dropout. The Softmax output layer was used with the categorical cross-entropy loss function to apply the CNN model proposed with the Adam optimization technique. The results obtained by the proposed model were 97.42% in the training phase and 96.18% in the testing phase.

DOI: 10.22401/ANJS.24.2.09

*Corresponding author: reem.alakkam@gmail.com

1. Introduction

Agriculture is a significant source of economic growth. Each year, the Earth's population grows by around 1.6%, so does the demand for plant products of all kinds[1]. Plant protection against plant diseases plays a crucial role in meeting the increasing demand for food quality and quantity [2]. In terms of economical value, plant diseases alone cost the world economy about $ 220 billion yearly[3]. So part of the economy depends on agriculture. As a result, plant harm would drive to an enormous loss of productivity and, ultimately, would have an impact on the economy. Leaves, which are the most vulnerable component of plants, are the first to exhibit disease symptoms. The detection of plant diseases plays a major role in agriculture [4]. In this context, there are a large number of plant diseases. Proper and accurate diagnosis of plant diseases plays an important role in preventing loss of agricultural production. Typically, disease detection in plants is performed manually. Experts, such as botanists and agronomists, carry out these techniques, and these classical methods are also difficult and time-consuming [5,6]. Consequently, it is essential to automatically identify diseases utilizing progressed methods. In such manner, the agrarian businesses are zeroing in on artificial intelligence strategies.

In addition, deep learning (DL) created critical improvements in the plant leaf discovery of illnesses field of explores. This is because to ability extract the features automatically by deep learning algorithms. Among a few farming issues, the effective classification of plant sicknesses is necessary to improve the quality/amount of agrarian products. Therefore, there is an arising research subject for progressing agricultural computerization [7]. The purpose of this study is to use the deep Convolution Neural Networks (CNN) model to solve problems with plant leaf disease classification. The CNN is a type of deep learning algorithms. By adding more complexity and hierarchical representations of data into the model, deep CNNs have wide applications In image classification, natural language processing, object detection, speech recognition, and recommendation systems, etc. [8]. As follows, the remainder of the paper is structured. The associated works are presented in section 2. The materials and methods for identifying plant diseases are discussed in the section on 3. Section 4 gives the outcomes and the associated discussions. Finally, the findings and possible study directions are discussed in Section 5.

2. Related Works

The automated method of leaf disease is one of the most significant areas of research because it offers many benefits in saving plants. Much research is underway in this area. Here we take some of the work discussed below on the identification of plant leaf diseases using different advanced techniques. In the year 2015, a number of crop types, namely fruit crops and vegetable crops, have been adopted by J. D. Pujari et al. [9]. For each type of crop, distinct methods have been adopted. K-mean clustering for fruit crops is the type of segmentation used. Using ANN and nearest neighbor algorithms, texture features were oriented and graded, achieving an overall precision of 90.723 %. The Chan vase system for segmentation was used for vegetable crops. Local binary patterns obtain an overall average precision of 87.825 % for extracting texture characteristics and SVM and k-nearest neighbor algorithms for classification.

In 2017, Guan Wang et al. [10] used the Plant Village dataset and transfer learning to train several models on the same dataset, VGG16, VGG19, Inception-v3, and ResNet50.Equipped to diagnose the seriousness of the disease in a sequence of deep convolution neural networks. Systematically tested are the features of shallow networks trained from scratch and deep models tuned using transfer learning. The deep VGG16 model is the best model, trained using transfer learning, which gives an overall accuracy of 90.4% on the retention test set. R. Sangeetha, M. Mary Shanthi Rani[11], Describes the methodology for disease detection and prediction using deep-learning techniques for tomato plant leaves in 2019. Using an open database of 13,848 photos, which has included 7 separate groups of [plant, disease] combinations, including healthy tomato crops, model training was carried out. To predict safe and unhealthy leaves infected by two types of pathogens, septoria spots and bacterial spots, a convolution neural network has been used, which is well adapted for identification and prediction tasks. The model's efficiency is measured using precision, recall and F1-score and the model has achieved the accuracy, of 94.66 %. Experiments are carried out using the plant village dataset. The significantly high rate of success makes the model a very useful tool for early warning or advice.

A deep learning model using the CNN algorithm was sophisticated by Francis, Mercelin, and C. Deisy [12]. This model is dependent on four convolution layers, and each layer is followed by the pooling layer, the fully connected layers and

Even the Sigmoid feature have also been used to discover the probability of disease happening or not. This model classified the healthy and sick leaves of tomato and apple plants using a dataset consisting of 3663 images with 87 % classification accuracy.

3. Convolutional Neural Network

Convolutional neural network (CNN) was at first put forward by LeCun in the early1980s [13]. Convolutional neural networks (CNNs) are a special Scenario of feed forward neural networks. As they are often made up of neurons with learnable weights and biases, they are very similar to normal neural networks. But CNNs, in comparison to MLPs, allow the clear assumption that inputs, including images, have a particular structure. This allows this property to be encoded in the architecture, sharing the weights for each image position and making the neurons only respond locally [14]. The Convolutional Neural Network is the deep learning algorithm typically used for computer vision and image processing applications. Instead of using hand-defined feature extraction to obtain various image characteristics (e.g., shape, texture, color, etc.), deep learning and in particular the convolutional neural network takes an approach different to extract features from an image, in CNN, the features of the image are automatically and gradually learned from the training process, so the main advantage of CNN is that it allows you to skip the feature extraction step and instead focus on training the network to learn convolutional filters [15]. For several important reasons, the CNN is beyond classic models: First, the main interest in implementing CNN falls in the principle of using the definition of weight sharing, which greatly decreases the number of parameters that require training, resulting in enhanced generalization [16]. The CNN can be trained smoothly due to smaller parameters and does not over fit [17]. Secondly, the classification stage is integrated with the extraction stage of features[18], all of which utilize the learning process. Thirdly, the implementation of large networks using general artificial neural network (ANN) models is much more complicated than the implementation of CNN [19]. CNN's general model Comprises of five components of a convolution layer, an activation function, a pooling layer, a flattened layer, a fully connected layer. The following diagrams show the functionality of each component:

1. **Convolution layer:** By extracting the local characteristics of each layer and creating complex characteristics as they pass through the hidden layers, this layer serves as the extractor of features. In this portion, the image is passed over a succession of filters, or convolution kernels, creating new images called convolution maps). The outputs of neurons in this type of layers are calculated by applying the following equation [20]:

$$W_2 = \frac{(W1-F)+2P}{S} + 1 \quad \ldots(1)$$

$$H_2 = \frac{(H1-F)+2P}{S} + 1 \quad \ldots(2)$$

$$D_2 = F \quad \ldots(3)$$

where:

W_2: new width
H_2: new height
D_2: new depth
F: spatial extend of the filter (or called filter size such as (3×3)).
K: number of filters (such as (16, 32, 64)).
P: zero paddings (hyperparameter controlling the output volume).
S: stride (hyperparameter with which we slide the filter).

Figure 1 shows the activity of the convolution layer.

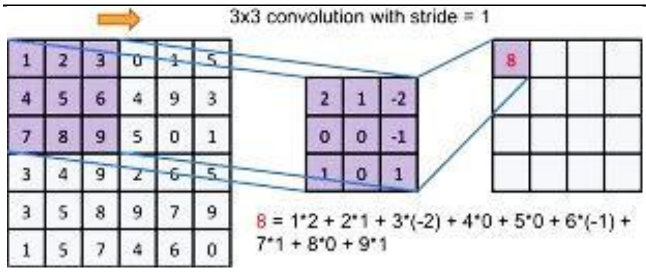

Figure 1. Convolution layer.

2. **Activation function:** The Rectified Linear Unit (ReLU) represented the activation function. The ReLU increased the nonlinear properties of the model and also the overall network. The output of the ReLU is the maximum input and zero, when the input value is non-negative, the output equals the input. the outputs of this function are calculated by applying the following equation:

$$F(x) = \max(x, 0) \quad \ldots(4)$$

3. **Max-pooling layer:** This layer includes a kernel used for down sampling of the input data. Feature maps from the convolutional layer are sampled down to a size specified by the pooling window size and the stride size of the pooling kernel. Figure 2 shows the operation of max-pooling layers. and the outputs of pooling layer are given by applying the following equations [20]:

$$W_2 = \frac{(W1-F)}{S} + 1 \quad \ldots(5)$$

$$H_2 = \frac{(H1-F)}{S} + 1 \quad \ldots(6)$$

$$D_2 = D_1 \quad \ldots(7)$$

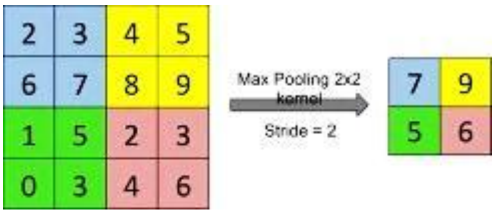

Figure 2. Max pooling layer.

4. **Flatten layer:** This layer was used to construct the one-dimensional from multidimensional array, usually used in the move from a layer of convolution to a fully connected layer.
5. **Fully connected layer (FC):** This layer is the basic layer where, before adding an activation function, every node in the fully connected layer multiplies each input by a learnable weight and outputs the sum of the nodes applied to a learnable bias. This layer takes the inputs from the previous layer.

4. Materials and Methods

Under the necessary subheadings, this section explains the dataset and the steps involved. Building a convolutionary neural network to effectively identify and predict leaf diseases in plants is the main objective of this paper. Target leaf spot, mosaic virus, yellow leaf curvature virus, bacterial spot, downy mildew, late blight and septoria leaf spot, etc. are common diseases affecting the leaves. Four main phases are involved in the proposed methodology, namely: data collection, pre-processing, training, and data prediction. The flow diagram is shown in Figure 3.

4.1 Dataset (image acquisition)

The dataset for the experiment called Plant Village was downloading from a Kaggle website with size 256×256 and JPG format, containing 34,934 RGB color space leaf images, the dataset contains 15 classes including healthy leaves is downloaded. The samples per class of the dataset are shown in Figure 4.

4.2 Images pre-processing

The image must be processed before being sent to the algorithm for testing and training purposes. To this end, there are several steps involved in preprocessing.

1. **Balancing:** Since the dataset imbalances, this method is implemented by selecting 1500 samples from each class.
2. **Resize:** The images have been resized by converting each image into a pixel array in 150×150 dimensions in order to make the model training possible computationally.
3. **Pixels normalization:** Neural networks process inputs using small weight values and the learning process can be interrupted or slowed down because inputs with large integer values. As a consequence, the pixel values are normalized from 0-255 to 0-1. This strategy helps to speed up the process of training.
4. **Labels converting:** convert each label into binary level.
5. **Data augmentation:** This performs stretch training dataset during training by applying random rotations, shifts, flips, crops, and sheers on image data set. This allows using a smaller data set and still achieving high results.
6. **Dataset splitting:** The dataset is splitting into 70 % for training and 30 % for validation.

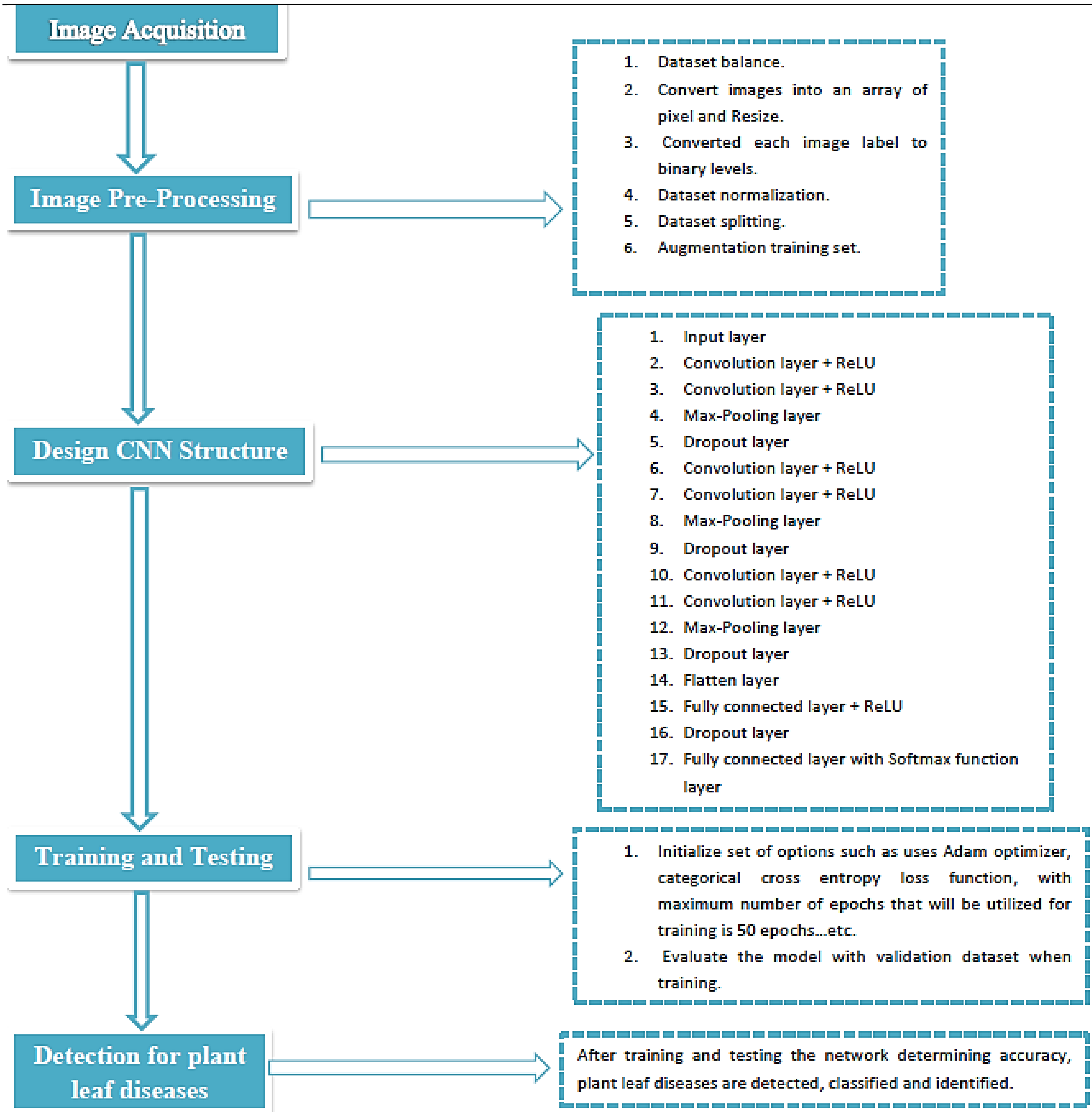

Figure3. Proposed method.

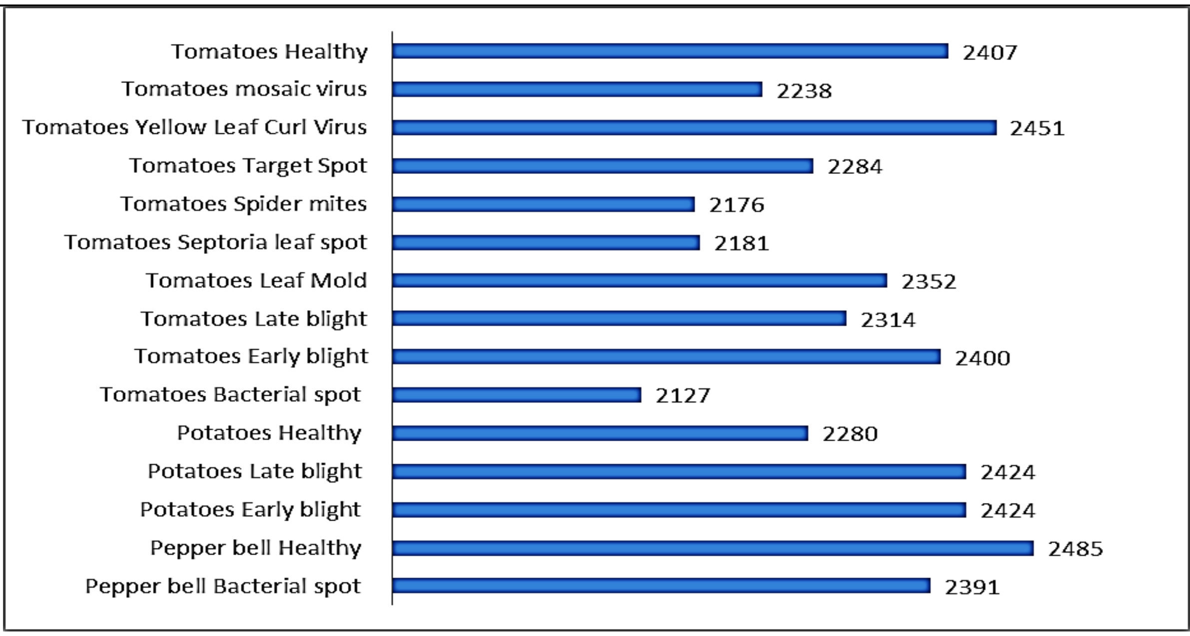

Figure4. Statistics of dataset.

4.3. Proposed CNN

The proposed CNN model has input layer, six convolutional layers, six non-linear layer, three max-pooling layers, a flatten layers, and two fully connected layers. The convolution and pooling layers labor as features extractors from the input images, while the fully connected layer as a classifier. The first and second convolution layers contain 16 filters with 3×3 kernel size. The third and fourth convolution layers contain 32 filters with 3×3 kernel size. The fifth and sixth convolution layers contain 64 filters with 3×3 kernel size. The convolution stride is one pixel without space padding for all convolution layers. A rectified linear unit (ReLU) operation follows any convolution layer. ReLU is a feature of activation that provides a solution to gradient vanishing. There is a max pooling layer with 2×2 window size and stride is two pixels and a dropout layer with a dropout rate of 0.25 after every two convolution layers. The term dropout refers to the random removal of units and their connections during training. Dropout is a technique used to reduce overfitting. After several iterate layers yet the data comes to the flatten step, i.e., transformed into a one-dimensional (1D) array of numbers (or vector) to form a vector of a single long element. And it is related to the final model of classification, called a fully connected layer. There is a fully connected layer containing 1024 units. It was followed by a rectified linear unit (ReLU) and a fully connected layer containing 15 units representing the number of classes followed by the softmax function. The softmax classifier input is a vector of features that result from the learning process, and the output is likely to belong to a given class image, there’s a dropout layer with a dropout rate of 0.5, between the two FC layers. The algorithm below to illustrate the mechanism of action CNN to make network converges.

Algorithm: CNN work.

Input: Model structure before training.
Output: Built CNN converges with optimum weight.
Steps:
Step 1: Provide input vector to the network.
Step 2: Perform convolution using filter to produce a feature map.
Step 3: Pass the obtained feature map through ReLU to introduce non-linearity.
Step 4: Apply pooling operation on obtained feature map, which introduces translation invariance.
Step 5: Apply dropout layer.
Step 6: Repeat Steps 2 to 5 for repetition of layers.
Step 7: The obtained feature maps are passed to fully connected layer for classification.
Step 8: Pass the output to a classifier such as softmax.
Step 9: Computer loss at the final layer and calculate gradient w.r.t. all the learnable parameters.
Step 10: Backpropagate the error component and update the parameters.
Step11: Perform the forward pass and repeat Steps 2 to 10 using updated parameters until network Converges with the optimum weight.
Step (13): End algorithm**.**

4.4. Training and testing phase

To classify 15 categories from Plant Village dataset, the model was trained. With random values from a Gaussian distribution, the weights of all layers are first initialized. The optimization was performed using Adaptive Moment Estimation (Adam) optimizer updates the weights of the network with initial learning-rate 0.001. Categorical cross entropy loss is utilized as a cost function; the accuracy is utilized as a metric. The CNN training requires the choice of several parameters and hyperparameters; as a result, the batch size was used where the batch size reduces the cost of computational and enhances the model's performance Because it introduces the dataset in each epoch as a subset based on batch size that identify. The batch size of 32 was used and the model was trained in 50 epochs, an epoch is a hyperparameters that determines a single pass into the full training set when training a deep learning model. During the model evaluation, the validation accuracy is calculated by using the test dataset. After the training and testing phase, the model will be able to detect and classified diseases.

5. Results

The framework for this study implemented using the jupyter notebook and python 3.7 programming language. Efficient libraries of Python, i.e., Keras, Numpy, TensorFlow, Scikit-learn, Pandas, open cv2, pickle, matplotlib, are used. Work was carried out on an HP machine with a Core i7-4800MQ CPU, 16 GB of RAM. The final assessment is performed on the validation set in the testing phase after the training process has been performed on the training set, figure 5. Show the accuracy and loss curves for training epoch 50 during the training process.

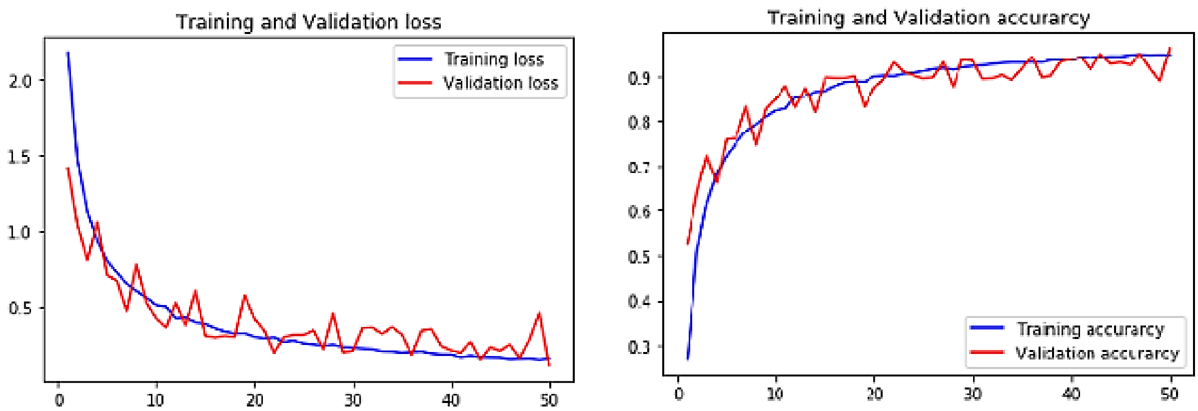

Figure5. Model accuracy and loss curves with 50 epochs.

As seen in the earlier figure, the accuracy of training and validation increases very quickly in the beginning. After about 20-50 epochs they begin to stabilize and grow very slowly, the loss Follows the same pattern; it decreases very rapidly and then stabilizes. For assessing the efficacy of the proposed method, accuracy measures are used. To measure the accuracy, the confusion matrix was used. Figure 6 displays the confusion matrix, giving an example of the number of images wrongly and correctly categorized by class.

As shown in the figure, the rate of misclassification is very low. The Tomatoes Late blight is misclassified more often as Potatoes Late blight and Tomatoes Early blight. In addition to it, Tomatoes Target Spotare is misclassified more often as Tomatoes Spider mites and Tomatoes Healthy. The reason for the misclassification is due to have similar features. The model performs very well for all other categories. The accuracy obtained after calculates from confusion matrix 97.42% on the training phase and 96.18% on the testing phase. By summing the diagonals representing the true positive (TP) and then dividing the outcome by the sum of all the numbers in the matrix representing the true positive (TP), the true negative (TN), the false positive (FP), and the false negative (FN), the accuracy was determined. The formula below was used to compute the model performance accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \qquad \ldots(8)$$

$$= \frac{422+442+462+452+452+492+422+402+402+372+472+402+442+452+442}{422+442+462+452+452+492+422+402+402+372+472+402+442+452+442+259}$$

$$= \frac{6530}{6789} = 96.18\ \%$$

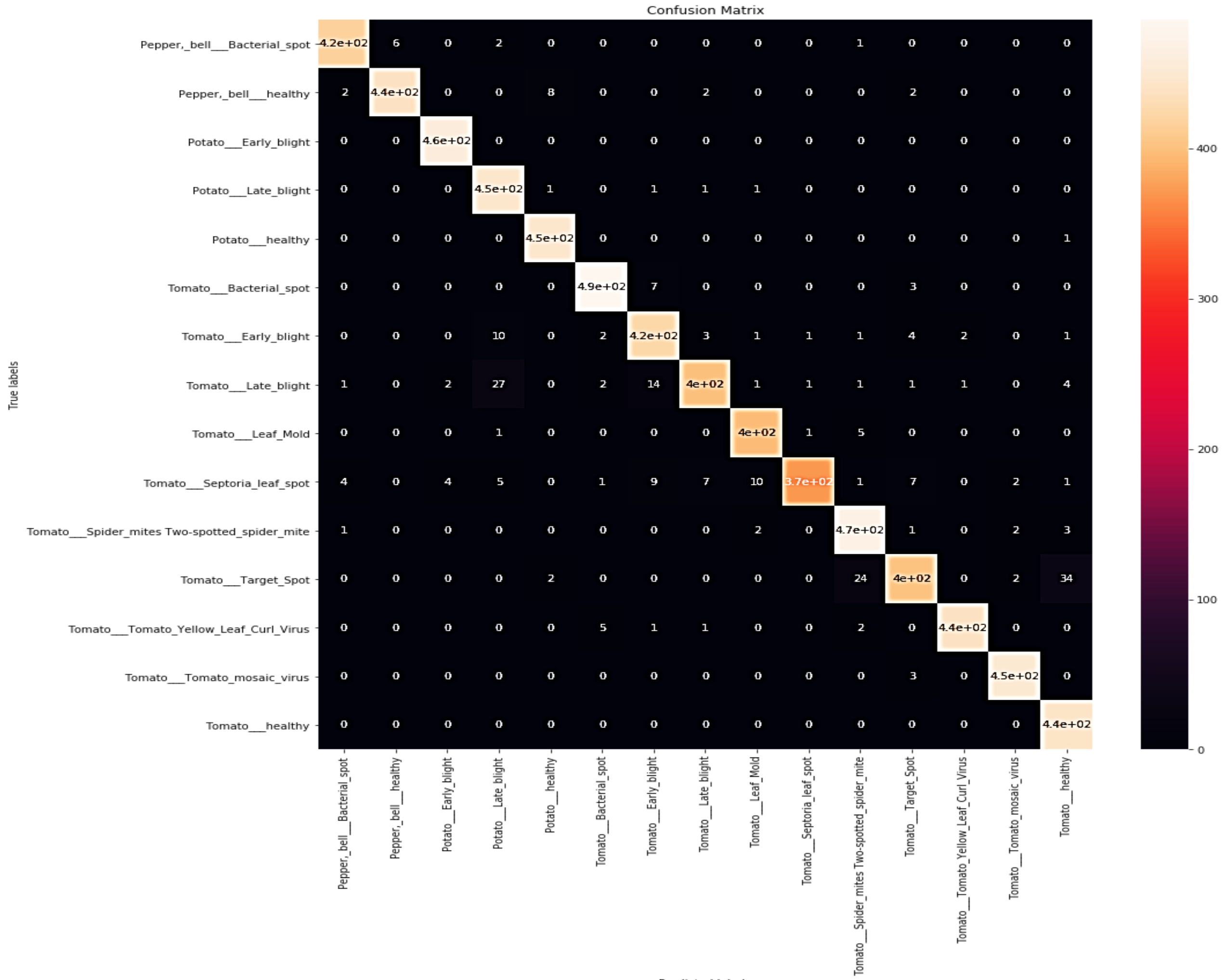

Figure6. Confusion matrix of the model.

Through confusion matrix can find out the accuracy of each class by computing precision as a shown in formula and table below.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \ldots(9)$$

Class labels	TP	FP	Precision
Pepper bell Bacterial spot	422	8	98%
Pepper bell Healthy	442	6	99%
Potatoes Early blight	462	6	99%
Potatoes Late blight	452	45	91%
Potatoes Healthy	452	11	98%
Tomatoes Bacterial spot	492	10	98%
Tomatoes Early blight	422	32	93%
Tomatoes Late blight	402	14	97%
Tomatoes Leaf Mold	402	15	96%
Tomatoes Septoria leaf spot	372	3	99%
Tomatoes Spider mites	472	35	93%
Tomatoes Target Spot	402	19	95%
Tomatoes Yellow Leaf Curl Virus	442	3	99%
Tomatoes mosaic virus	452	6	99%
Tomatoes Healthy	442	44	91%

6. Conclusions and Future Work

In this research, we introduced a convolution network focused on to the identification and classification of leaf diseases in plants. The suggested model can work as a decision-making method to help farmers recognize plant diseases. To detect and predict 12 diseases, the aim of this research is to apply deep neural network. Therefore, in detecting plant leaf diseases, CNNs have clear benefits. The findings of this study demonstrate the possibility of using CNN for the identification of plant leaf diseases, which would greatly increase the detection of diseases. While the accuracy of the model's disease classification is not 100 percent, it is possible to introduce changes to the current system in future research to strengthen the method and provide more efficient and precise guidance on disease control. This showed that with little computational effort, the proposed method would significantly support accurate detection of plant leaf diseases. Based on the results obtained, we intend to test more plant diseases with our model in our future work. In addition, we will focus on automatically estimating the severity of the detected disease, as this is major problems that can help farmers decide how to intervene to stop the disease.

References

[1] E.-C. O. H.-W. D. F. S. A. Weber; "Crop production and crop protection: estimated losses in major food and cash crops". Elsevier Science, 1994.
[2] R. N. S. A. P. R. Scott; "Plant disease: A threat to global food security," Annual Review of Phytopathology, 43(1), 83-116, 2005.
[3] Agrios, G.; "Plant pathology 5th edition", Elsevier Academic Press, 2005.
[4] Zhou R.; Kaneko S. I.; Tanaka F.; Kayamori M. and Shimizu M., "Disease detection of cercospora leaf spot in sugar beet by robust template matching", Computers and Electronics in Agriculture, 108, 58-70, 2014.
[5] Turkoglu M. and Hanbay D., "Apricot disease identification based on attributes obtained from deep learning algorithms", presented at the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018.
[6] TÜRkoĞLu M. and Hanbay D.; "Plant disease and pest detection using deep learning-based features", Turkish Journal of Electrical Engineering & Computer Sciences, 27(3), 1636-1651, 2019.
[7] Saleem M. H.; Potgieter J. and Arif K. M.; "Plant disease classification: A Comparative evaluation of convolutional neural networks and deep learning optimizers", Plants (Basel), 9(10), Oct. 6, 2020.
[8] G. G. and A. P. J; "Identification of plant leaf diseases using a nine-layer deep convolutional neural network", Computers & Electrical Engineering, 76, 323-338, 2019.
[9] Pujari J. D.; Yakkundimath R. and Byadgi A. S.; "Image processing based detection of fungal diseases in plants", Procedia Computer Science, 46, 1802-1808, 2015.
[10] Wang G.; Sun Y. and Wang J.; "Automatic Image-Based plant disease severity estimation using deep learning", Comput Intell Neurosci, 2017, 2917536, 2017.
[11] Pala T.; "Plant disease detection on leaf using deep learning", Master of Engineering with specialization in Computer Engineering, Atmiya Institute of Technology and Science, Technological University, 170030702009, 2019.
[12] Francis M. and Deisy C.; "Disease detection and classification in agricultural plants using convolutional neural networks-a visual understanding", presented at the In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), March 2019, 7-8, 2019.
[13] Geetha N. A. A. M.; "A review on deep convolutional neural networks", Presented at the 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017.
[14] Teuwen J. and Moriakov N.; "Convolutional neural networks," in Handbook of Medical Image Computing and Computer Assisted Intervention, 481-501, 2020.
[15] Mahdi A. S. and Dhannoon B. N.; "Color model based convolutional neural network for image spam classification", Al-Nahrain Journal of Science, 23(4), 44-48, 2020.
[16] Arel I.; Rose D. C. and Karnowski T. P.; "Deep machine learning-a new frontier in artificial intelligence research [research frontier]", IEEE computational intelligence magazine, 5(4), 13-18, 2010.
[17] Smirnov E. A.; Timoshenko D. M.; et al.; "Comparison of regularization methods for imagenet classification with deep convolutional neural networks", Aasri Procedia, 6, 89-94, 2014.
[18] LeCun Y.; Bottou L.; Bengio Y.; et al.; "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, 86(11), 2278-2324, 1998.
[19] Tivive F. H. C. and Bouzerdoum A.; "Efficient training algorithms for a class of shunting inhibitory convolutional neural networks", IEEE Transactions on Neural Networks, 16(3), 541-556, 2005.
[20] Jmour N.; Zayen S. and Abdelkrim A.; "Convolutional neural networks for image classification", in 2018 International Conference on Advanced Systems and Electric Technologies (IC–ASET), 397-402: IEEE, 2018.