Application of Deep Convolutional Neural Networks for Identification of Diseases on Potato Leaves

Senju Thomas Panicker [1,*], Neeradha C K [2]

[1] Control and Instrumentation Group, Centre for Development of Advanced Computing, Thiruvananthapuram, Kerala, India

[2] Department of Mathematics, Mar Ivanios College, Thiruvananthapuram, Kerala, India

Abstract

In any form of agriculture, especially precision farming, early detection of diseases in plants is essential for disease control so as to obtain desired yield. Traditionally, this process has been carried out manually by visual inspection, which is time consuming and requires the assistance of an expert or a seasoned farmer. With advances in technology and computer vision, automatic methods of detecting diseases have become a viable solution in the field of agriculture. Most of the plant diseases leads to visual changes on the leaves which are used by computer vision algorithms to identify diseases. This paper proposes the application of Convolutional Neural Network for identification of diseases on potato plants. The proposed Deep CNN model yields a test accuracy of 97.10%.

Keywords: Deep learning, Neural Networks, Deep CNN, Accuracy, Dropout, Confusion Matrix

1. Introduction

The economy and development of developing countries like India is dependent on agricultural production. In a country like India where majority of the population depends on agriculture for making living, the role of agriculture in economic development cannot be set aside. In India, about 70% of Indian economy relies on agriculture [1]. Any sort of damage to the crops, whether due to diseases or natural calamity, would lead to great loss in productivity and would eventually affect the economy.

India is notable to the world as the biggest maker of heartbeats, rice, wheat, flavors and zest items. Agriculture is the major source of employment to the country’s workforce. It provides employment to a large proportion of the population. The economic growth of the countries farming population depends not only on quantity but also on the quality of the agricultural products that they produce, which is directly related to the plant's growth and the yield. Disease in plants reduces the production and quality of the agricultural products and affects the framers economically.

In spite of new advance in agriculture, farmers end up spending lakhs of rupees on disease control, often without adequate technical support. Use of modern technology in early detection and disease control can help the farmers a lot. The excessive use of pesticides for disease control pollutes the water and soil environment and ruins the eco-system. The early detection of diseases helps in reducing the excessive use of pesticides and fungicides. It also prevents the further spread of the diseases.

One of the worst plant diseases that man has ever confronted is the Late Blight disease of potato caused by the fungus phytophthora infectants. Late blight of potato caused severe starvation and famine in 1840 which took the lives of around 1 million people in Ireland alone [2]. Such massive destruction of crops could be avoided if the disease detection is done in the early stage.

Most of the plant diseases are caused by fungi. The most affected part of the plant by the fungus attack is the leaf. Fungal diseases make prominent visible changes on the surfaces of the leaves. These changes include change in color or nature of the leaves. It can also result in wilting of leaves [3].

Farmers periodically keeps checking the leaves for any disease symptoms. However, manual detection of plant disease by checking the leaves is tedious process and can sometimes go wrong in exactly identifying the disease caused in leaves. Introducing computational methods for early disease detection will make detection easy and accurate. Analyzing and classifying the leaf images will help in automating the process of disease detection.

In modern agriculture, identification of diseases by analyzing images of plant leaves for disease symptoms is often made use of [4]. Artificial intelligence is also used to aid the process of disease detection. The most widely used machine learning algorithms for image classification are K-nearest neighbours (K-NN), support vector machine (SVM), logistic regression and decision tree [5]. Recently several deep learning algorithms based on convolutional neural networks are being used for image classification.

In this project, we use Deep CNN based deep learning model for identification of plant leaf disease from images of leaves. Deep learning is an extension of machine learning where more complexity and hierarchical data representations is added to the model. The Deep CNNs are being extensively used for various applications including classification of images, detection of objects from image and video, recognition of speech from an audio or video file, and natural language processing [6].

2. Literature Survey

Disease detection in plants can be achieved in many ways, either using conventional methods or by way of making use of modern artificial intelligence techniques such as machine learning and deep learning, which has seen accelerated growth owing to the digital revolution, which makes available huge amounts of data, and rapid advances in image processing and computing power of machines.

Machine learning is an efficient mechanism for decision making and provides a powerful framework which integrates expert knowledge into the system. The following literature sheds light on some of the of machine learning algorithms. They are widely used in many fields and agricultural automation is one of them. Yang CC, et al. proposed an image classification model for decision-making in agricultural field by employing learning approaches like decision trees and logistic regressions [7].

Artificial neural network (ANN) is a computation scheme which simulate the way in which the human brain performs information processing and analysis. ANN is the foundation on which artificial intelligence (AI) is built and helps provide solutions to problems which were considered highly difficult or even impossible by human or statistical standards. The self-learning capacity has given ANN the ability to produce better results as more data becomes available. Srinivasa Rao and M. S. Prasad Babu proposed a “feed forward neural network” for perceiving different bugs and diseases in leaves by employing back propagation [8]. Back propagation is method used in artificial neural networks to compute the gradient descent values which helps in minimizing the error between the actual value and predicted value.

Sankaran S et al. explained about conventional methods, based on spectroscopy, imaging, and volatile profiling, that was used for detection of diseases in plants [9]. The paper explains the benefits of these approaches and sheds light on the limitations associated. A hyperspectral image classification was proposed by Guo Y et al. where they used the highly popular K-NN algorithm with the guided filter technique [10].

Support Vector Machine (SVM) algorithm was used by Garcia-Ruiz F et al. and Wetterich CB et al. for extracting features related to citrus greening disease found in lemon trees [11, 12]. SVM classification and hyperspectral imagery was proposed by Rumpf T et al. [13] and Calderón R et al. [14], for early detection of diseases in plants. The SVM model proposed by Mokhtar U et al. identifies tomato leaves infected with the yellow leaf curl disease [15]. The classification model based on Support Vector Machine (SVM) used 200 images of infected and healthy tomato leaves for learning and produced an average classification accuracy of 90%. The authors in [16] used SVM for identification of plant genome associations with bacteria. Another application of SVMs was presented in [17] where it was used for identification of diseases in the leaves of vine plants

Sladojevic S et al. [18] and Ferentinos KP [19] used Deep CNN algorithm for identification of diseases in various plants by verifying leaf images. Lee SH et al. [20] and Grinblat GL et al. [21] developed a Deep CNN model for identification of different plants using images of plant leaves. The model can also be used for identification of diseases and pests associated with these plants. This technique was implemented for detection of diseases and pest in tomato plant [22]. The authors in [23] studied 40 different research works that were based on deep learning approaches and were applied to several agricultural challenges.

Deep CNN needs huge amount of data for training the model in order to produce accurate results. In several cases sufficient amount of data may not be available to train the model. The lack of data can be overcome by enhancing the dataset by making use of image augmentation techniques like rotation, image flipping, scaling, shifting, PCA colour augmentation and noise injection [24]. The use of the enhanced dataset will improve the performance of the model.

3. Convolutional Neural Network

Convolutional Neural Network, also known as ConvNet, is a category of neural networks that is applied in processing data that has topology in the form of a grid, such as in the case of images. For image classification applications, the ConvNet deep learning algorithm takes images as inputs, identify patterns from various aspects in the image, detect objects and segregate them into various classes. In contrast to traditional image classification algorithms and machine learning algorithms, ConvNet requires much lesser preprocessing and filters need not be predetermined or hand-engineered. Convolutional neural networks have the capacity to learn the required filters during the training process.

The CNN being a class of neural networks, its architecture is also similar to that of the linking between of neurons in the human cerebrum [25]. Individual neurons pick up stimuli only in a specific region of the visual area referred to as the receptive field. The individual neuron in a ConvNet process the data which is relevant to its receptive field. The layers are organized in a specific manner such that they will be able to detect simple patterns, such as lines, curves, etc in the initial layers, followed by more complex patterns, such as faces, objects, etc as we traverse to deeper layers. A typical CNN consists of mainly three layers: the convolutional layer, the pooling layer, and the fully connected layer, as shown in Fig.1

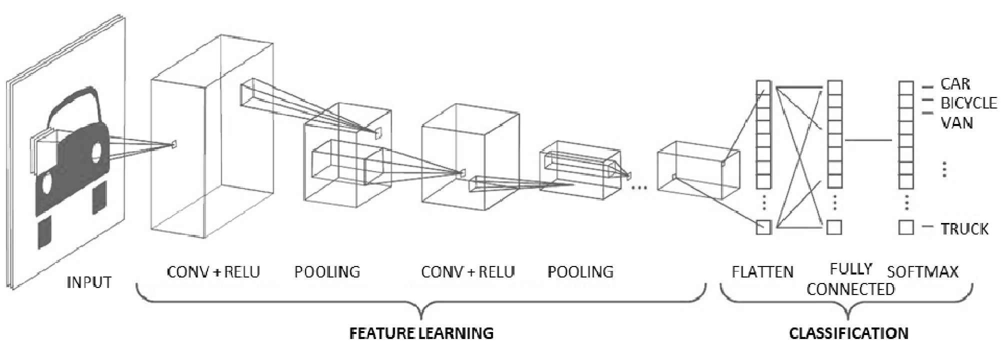


Fig.1 Architecture of CNN

3.1 Convolution Layer

As the name suggests, the convolution layer is the main part of CNN, which manages the major computational load of the network. However, in mathematical terms, the convolution layer performs a dot product between two matrices, instead of a convolution operation as the name suggests. One matrix is the is the restricted portion of the receptive field, for example the input image and the other matrix is the set of parameters that are to be learned. The matrix of learnable parameters is also known as kernel.

During training of the model, the convolutional layers create feature maps of the input by applying learned filters on the input images, in a systematic manner. Convolutional layers have proven to be highly efficient in identifying various features from the images. In CNN, the convolutional layers close to the input are effective in detecting simple features such as lines and curves while deeper conv layers identify high-order features, like shapes or specific objects, which are more abstract in nature.

In CNN, the convolution operation of a 2D image is performed by using continuous sliding convolution window, which generates a corresponding convolution result. In CNNs, each feature map is convoluted by multiple input feature graphs. The convolution operation on an input u at the ith convolutional layer can be represented as,

$$v_i = A(K_i * u) \tag{1}$$

where

$*$ indicates convolution,

K_i represents the convolution kernels of the layer,

$K_i = [K_i^1, K_i^2, \ldots, K_i^M]$, M is the number of kernels of the layer

A is the activation function

Each kernel K_i^m is an $\alpha \times \alpha \times D$ weight matrix where D is the number of input channels and F is the window size [26].

During the forward pass of the training, the kernel is moved horizontally and vertically along the image, which results in a minimized representation of the image. Convolution operation is performed between the image and kernel as the kernel is moved across the image. The convolution operation results in an activation map, which is the 2D representation of the image. The kernel or filter is moved along the image as per the stride specified. Although, the default stride of one position gives a more accurate result, a higher stride can assist in arriving at a conclusion faster.

The parameters that define the kernel are its filter size, stride and padding. The filter size refers to the length and width of the filter or kernel. The stride is the number of positions the filter slides after each convolution operation.

For an input $W \times W \times D$ fed to the convolution block which has M number of kernels with a filter dimension $\alpha \times \alpha \times D$, stride of δ and padding ρ, then the dimension of the output of convolution operation is $W_{out} \times W_{out} \times D_{out}$, where

$$W_{out} = \frac{W - \alpha + 2\rho}{\delta} + 1 \tag{2}$$

3.2 Activation Function

Even though the convolution operation is linear in nature, the images which are fed as input to the convolution layers are far from linear. For this reason, a non-linearity operation, known as activation, is introduced after the convolution operation. There are several types of activation functions which are used in ConvNets, out of which the sigmoid, tanh and ReLU activations are most popular.

The “Rectified Linear Unit (ReLU)” activation function is an unsaturated nonlinear function. It performs activation much similar to that of brain neurons. In contrast, the “sigmoid” and “tanh” activation functions are saturated non-linear functions whose performance is not as good as the ReLu activation. ReLU is more reliable and causes convergence of the learning process much faster. Hence, ReLu is the most preferred activation function in deep learning. It is mathematically represented as

$$A(v_i) = max(0, v_i) \tag{3}$$

For negative values, the ReLu generates zero as output and for positive input values, it outputs the same value itself, as shown in Fig.2.

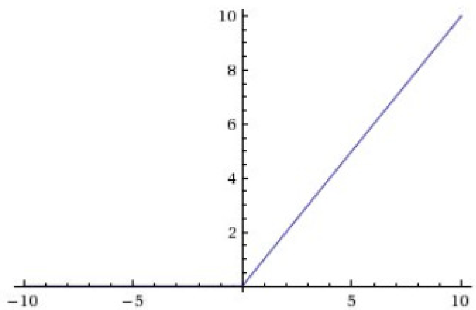


Fig.2. ReLu Activation Fuinction

3.3 Pooling Layer

Even though the convolutional layers offer considerable advantage of needing lesser parameters as compared to traditional neural network layers, the number of parameters increases exponentially as the number of convolution layers increases. The pooling layer is a layer that follows the conv layer in order to reduce the parameters that are passed on as input to the next layer. This effectively reduces the number of parameters that in the network. The pooling operation is performed by identifying the statistical characteristics of a region that can represent the characteristics of that region. The pooling operation offers some translation invariance by means of which objects would be recognizable irrespective of its position in the image.

The pooling layer features a 2-dimentional filter which performs pooling operation on the output of the convolutional layer. For a convolutional output of size $W_{out} \times W_{out} \times D_{out}$, pooling filter size of β and stride of φ, the pooling operation reduces the number of parameters to $P_{out} \times P_{out} \times D_{out}$ where

$$P_{out} = \frac{W_{out} - \beta}{\beta} + 1 \tag{4}$$

3.4 Dense Layer or Fully Connected Layer

The convolutional layers are followed by fully connected layers similar to that of traditional fully connected neural networks. The neurons in these layers are fully connected with all the neurons in the subsequent layer and the preceding layer, as shown in Fig.3. A dense layer obtains all the features learned from the previous layer. This layer performs matrix multiplication of the input with the weight parameters in the layer and adds a bias weight.

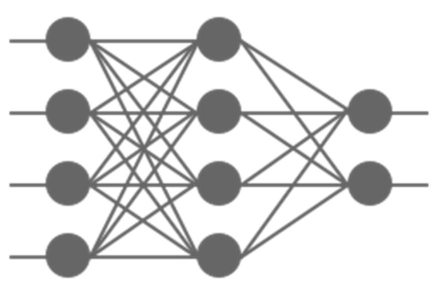


Fig.3. Fully connected layers

3.5 Optimization algorithm

Deep CNN is an extremely iterative process wherein the model is trained numerous times in order to identify the best one. The weights of the model are updated in each iteration using gradient descent optimization algorithm. It is a simple optimization technique used by machine learning algorithms in order to minimize the cost function during training. The model weights are updated in gradient steps while making using of all training data during each training step and hence the name batch gradient descent. The average of the gradients of all the training data is used to update the model parameters.

During the training operation, the model makes predictions on the training data which is then compared with the actual data to compute the error. Gradient descent works by updating the model parameters in small gradient steps so as to minimize the error. The model parameters, also known as weights or coefficients, is updated in smalls steps along the gradient of the error plot that slope down towards a minimum error value.

Since deep learning works on large datasets, the traditional batch gradient descent algorithm works very slowly and requires large amount of memory. This is solved by using the mini-batch gradient descent algorithm where dataset is split into small batches which are then used to train the model. The model parameters are updated for each batch. This greatly speedup the training process and also requires comparatively lesser memory. The mini-batch sizes are selected based on the memory available for GPU or CPU hardware. Batch size has a great effect on the learning process. Small values of batch size results in quick convergence at the cost of introducing noise in the training process. Large batch sizes results in slow converges of the learning process. However, large batch sizes give accurate estimates of the error gradient.

3.6 Hyperparameters

Every deep learning model has a set of parameters, which are pre-fixed before learning and are used to control the learning process. They are divided into two main categories – optimizer hyperparameters and model specific hyperparameters. The model or network specific hyperparameters comprises of number of layers, number of hidden units, dropout, and activation function. The optimizer parameters comprise of learning rate, mini-batch size, steps per epoch and number of epochs [27].

Increasing the number of layers increases the accuracy up to a point and having more hidden units in the first layer provides good accuracy. Dropout is a regularization technique that helps in avoiding overfitting by randomly dropping outputs of certain hidden units from the training process [28]. In this process some neurons are randomly set to zero during the forward pass [29]. Overfitting normally happens when a large neural network is trained using small training samples. Typical values of dropout range from 0.1 to 0.8. The learning rate defines the speed of the learning process.

Large learning rate provides for learning process to be fast but it has the disadvantage of not converging to an optimum result. In contrast, using a low learning rate will slow down the learning process but it helps in converging smoothly. Epoch is a hyperparameter which indicates the number of times the entire training set has been iterated through the various layers of the network. More the number of epochs, better is the accuracy. However, the accuracy stops to increase beyond a certain number of epochs.

4. Process Description

This section gives a walkthrough of the process involved in identification of leaf diseases in potato plant using deep convolutional neural network. The steps involve pre-processing, designing convolutional neural network model, training and validation of the model and testing of the model.

This section gives a walkthrough of the process involved in identification of leaf diseases in potato plant using deep convolutional neural network. The steps involve pre-processing, designing convolutional neural network model, training and validation of the model and testing of the model.

4.1 Data collection and pre-processing

The images of potato leaves, both healthy and diseased, where downloaded from the plant village dataset uploaded in Kaggle website [30]. From the dataset, 2152 images of potato plant leaves were used for developing the proposed ConvNet model. The dataset is categorized into healthy leaves, early blight disease affected leaves and late blight disease affected leaves. The dataset contained 1000 images of early blight affected leaves, 1000 images of late blight affected leaves and 152 images of healthy leaves. The sample images are presented in Fig. 4.

Early Blight in Leaf | Late Blight in Leaf | Healthy Leaf

Fig. 4. Potato leaves image dataset sample

Convolutional Neural Networks require a large dataset for effective training of the model and the number of images belonging to various classes of images must be proportionately even in order to prevent overfitting of the model. In the dataset obtained, the number of images of healthy potato leaf is considerably less than the images of diseased leaf images. Hence image augmentation technique involving image flipping, rotation, resizing, noise injection and contrast enhancement has been used to enhance the dataset. Application of image augmentation techniques resulted in 912 of images of healthy leaves. This resulted in a total of 2912 images belonging to 3 classes.

The images were then further grouped to form the training dataset and test dataset. The training set consisted of 750 images each of the 3 classes and the rest were selected for validation and testing.

3.2 Deep CNN Model

The Deep CNN Model proposed for identification of diseases in leaf images consist of 11 layers, with 4 convolutional layers, 4 pooling layers and 3 fully connected layers, as depicted in Fig.5. The initial layers consist of convolutional layers and pooling layers. The training dataset, consisting of images mentioned in section 3.1, is fed as input to the Deep CNN Model.

The convolutional layers perform multiple convolution operations on the input images and specific patterns or objects in the images. The initial layers identify common patterns like lines are curves which form the edges of the diseased part on the image. The deeper convolutional layers identify complex patterns like patches and discoloration.

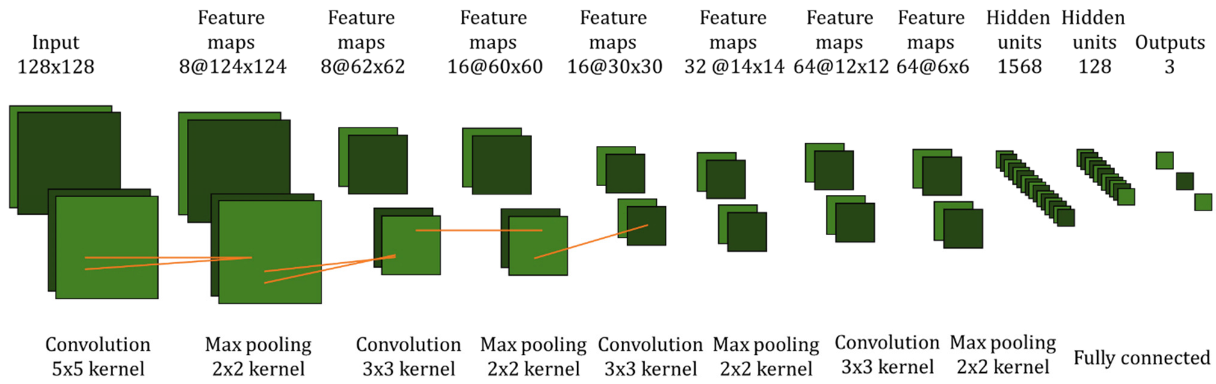
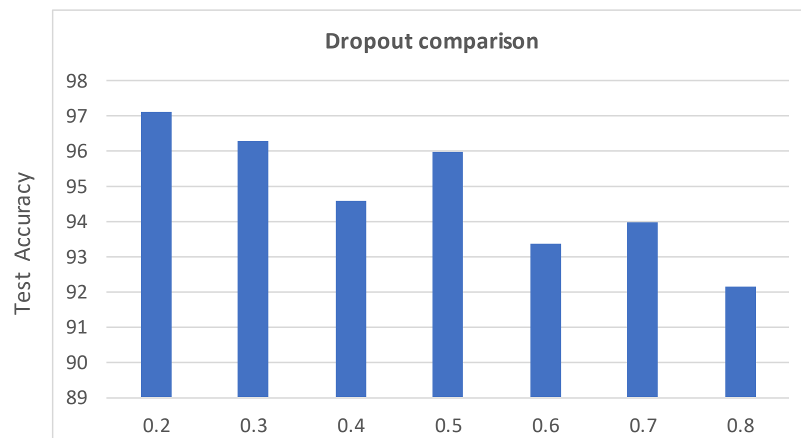


Fig.5. Proposed Deep CNN Model

The convolutional layers perform feature extraction from the images in the training dataset. The pooling layers reduce the dimension as we progress deeper into the layers [31]. The convolutional layers extract several lower-level features and transforms them into additional discriminative features. Each convolutional layer is followed by a max pooling layer which helps in reducing the number of model parameters. The set of convolution layers is followed by three fully connected layers, as shown in Fig.2.

Dropout regularization technique is used in order to avoid overfitting. Different dropout values ranging from 0.1 to 0.8 was employed in order to identify the best model. It was observed that the best test accuracy was realized with a dropout value of 0.2, as shown in Fig.6.



Droput	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Accuracy	97.10	96.27	94.61	95.99	93.37	93.97	92.13

Fig.6. Validation accuracy for various dropouts

The model was trained using various combination of hyperparameters to identify the optimum values. The model was trained using multiple epochs and the best accuracy was obtained when the model was trained for 300 epochs with a mini-batch size of 32 and 100 steps per epoch. As described earlier, a dropout value of 0.2 yielded the best results with optimum learning rate of 0.0007.

During training, the model achieved a best validation accuracy of 98.49%. On completion of training, training accuracy and validation accuracy is plotted, as depicted in Fig.7., in order to identify the effectiveness of the learning process

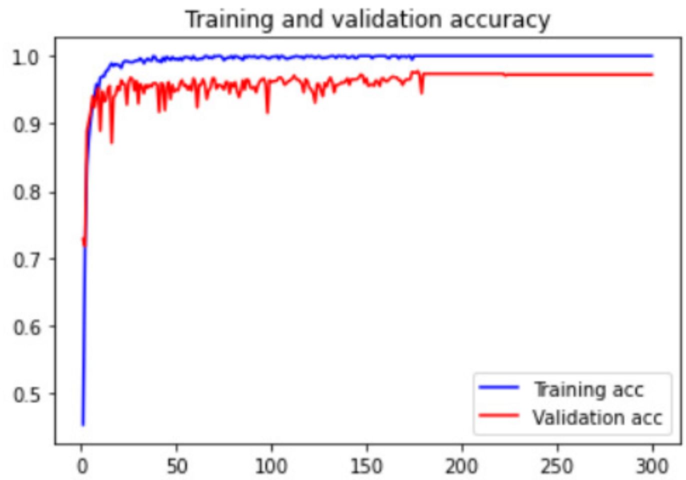


Fig.7. Plot of training and validation accuracy

The model was also trained using the non-augmented dataset, which contained lesser number of images. This resulted in a validation accuracy of 87.15% for the same set of hyperparameters. Once the training and validation process was completed, the trained model was used to test on test dataset. The testing process resulted in an accuracy of 97.10%

5. Results

The proposed ConvNet model was trained, validated and tested for identifying diseases associated with potato leaves. The model was trained using the enhanced dataset. After extensive training, the model yielded a highest test accuracy of 97.10%. The confusion matrix for the test result is given in Fig.8.

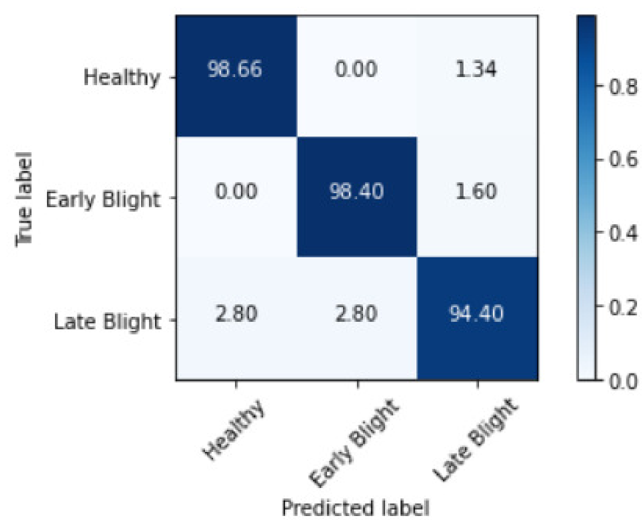


Fig.8 Confusion Matrix from the test data

6. Conclusion

Deep Learning is an effective method for detection of diseases in plant leaves and can help farmers in early detection of diseases. The results prove that the proposed ConvNet model can be used to identify the type of disease from the leaf images with accuracy. The study also showed that Deep CNN model trained with augmented dataset provided better accuracy as compared with original set which had lesser images. The ConvNet model proposed in this paper achieved a best testing accuracy of 97.10%. The Deep CNN model developed in this work will be integrated to a mobile application which will provide the farmers an easy way of identifying the disease with the help of their mobile phones. We also plan to extend this research to other type of plants and also provide mitigation solutions by incorporating augmented reality to the mobile app.

References

[1] http://www.fao.org/india/fao-in-india/india-at-a-glance/en/

[2] E. Charles Nelson and Jean Beagle Ristaino, “The Potato Late Blight pathogen in Ireland, 1846: reconnecting Irish specimens with the Moore–Berkeley correspondence”, Archives of Natural History 38(2):356-359, October 2011

[3] https://www.canr.msu.edu/news/signs_and_symptoms_of_plant_disease_is_it_fungal_viral_or_bacterial#:~:text=A%20symptom%20of%20plant%20disease,dahliae.

[4] Khirade SD, Patil AB, “Plant disease detection using image processing”, “ 2015 International Conference on Computing Communication Control and Automatio”, 2015, pp. 768–71. doi: 10.1109/ICCUBEA.2015.153.

[5] El Houby EMF, “A survey on applying machine learning techniques for management of diseases”, J Appl Biomed, 16(3),pp 165–74, 2018, doi: 10.1016/j.jab. 2018.01.002

[6] Wason R, “Deep learning: evolution and expansion”, Cogn Syst Res, 52, pp701–8, 2018, doi: 10.1016/j.cogsys.2018.08.023.

[7] Yang CC, Prasher SO, Enright P, Madramootoo C, Burgess M, Goel PK, et al, “Application of decision tree technology for image classification using remote sensing data”, Agric Syst ,76(3) pp 1101–17, 2003, doi: 10.1016/S0308-521X(02)0 0 051-3.

[8] M. S. Prasad Babu, B. Srinivasa Rao, “Leaves recognition using back propagation neural network-advice for pest and disease control on crops”, IndiaKisan 2007:13. http: //www.indiakisan.net/web/leafrecgnition.pdf (accessed October 12, 2019).

[9] Sankaran S, Mishra A, Ehsani R, Davis C, “A review of advanced techniques for detecting plant diseases”, Comput Electron Agric,72(1), pp 1–13,2003, doi: 10. 1016/j.compag.2010.02.007.

[10] Guo Y, Han S, Li Y, Zhang C, Bai Y. K, “Nearest neighbor combined with guided filter for hyperspectral image classification”, “Procedia Comput Sci”, 129, pp 159–65,2018, doi: 10.1016/j.procs.2018.03.066.

[11] Garcia-Ruiz F, Sankaran S, Maja JM, Lee WS, Rasmussen J, Ehsani R, “Comparison of two aerial imaging platforms for identification of Huanglongbing- infected citrus trees”, Comput Electron Agric , 91,pp 106–15,2013, doi: 10.1016/j.compag.2012.12.002.

[12] Wetterich CB, de Oliveira Neves RF, Belasque J, Ehsani R, Marcassa LG, “Detection of Huanglongbing in Florida using fluorescence imaging spectroscopy and machine-learning methods”, Appl Opt, 56(1), pp 15–23, 2017, doi: 10.1364/AO.56.0 0 0 015.

[13] Rumpf T, Mahlein A-K, Steiner U, Oerke E-C, Dehne H-W, Plümer L, “Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance”, Comput Electron Agric, 74(1), pp 91–9, 2010, doi: 10.1016/j.compag.2010.06.009.

[14] Calderón R, Navas-Cortés JA, Zarco-Tejada PJ, “Early detection and quantification of verticillium wilt in olive using hyperspectral and thermal imagery over large areas”, Remote Sens, 7(5), pp 5584–610, 2015, doi: 10.3390/rs70505584.

[15] Mokhtar U , Ali MAS , Hassanien AE , Hefny H, “Identifying two of tomatoes leaf viruses using support vector machine”, In: Mandal JK, Satapathy SC, Kumar Sanyal M, Sarkar PP, Mukhopadhyay A, editors. Information Systems Design and Intelligent Applications, New Delhi, Springer India, pp. 771–82, 2015.

[16] Martínez-García PM, López-Solanilla E, Ramos C, Rodríguez-Palenzuela P, “Prediction of bacterial associations with plants using a supervised machine- learning approach”, Environ Microbiol, 18(12), pp 4847–61, 2016, doi: 10.1111/1462-2920.13389.

[17] Pantazi XE , Moshou D , Tamouridou AA , Kasderidis S, “Leaf disease recognition in vine plants based on local binary patterns and one class support vector machines”, In: Iliadis L, Maglogiannis I, editors. Artificial Intelligence Applications and Innovations. Cham: Springer International Publishing; pp. 319–27, 2016.

[18] Sladojevic S, Arsenovic M, Anderla A, Dubravko Culibrk DS, “Deep neural networks based recognition of plant diseases by leaf image classification”, Comput Intell Neurosci 2016, pp 1–12, 2016, doi: 10.1155/2016/3289801 .

[19] Ferentinos KP, “Deep learning models for plant disease detection and diagnosis”, Comput Electron Agric, 145, pp 311–18, 2018, doi: 10.1016/j.compag.2018.01. 009 .

[20] Lee SH, Chan CS, Wilkin P, Remagnino P, “Deep-plant: plant identification with convolutional neural networks”, “2015 IEEE International Conference on Image Processing”, pp. 452–6, 2015, doi: 10.1109/ICIP.2015.7350839.

[21] Grinblat GL, Uzal LC, Larese MG, Granitto PM, "Deep learning for plant identification using vein morphological patterns", Comput Electron Agric, 127 pp 418–24, 2016, doi: 10.1016/j.compag.2016.07.003 .

[22] Fuentes A, Yoon S, Kim SC, Park DS, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition", Sensors 2017, 17(9), 2022, doi: 10.3390/s17092022.

[23] Kamilaris A, Prenafeta-BoldúFX, "Deep learning in agriculture: a survey", Comput Electron Agric 2018, 147, pp 70–90, 2018, doi: 10.1016/j.compag.2018.02.016 .

[24] Zhang Y-D, Dong Z, Chen X, Jia W, Du S, Muhammad K, et al, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation", Multimed Tools Appl, 78(3), pp 3613–32, 2019, doi: 10.1007/s11042- 017- 5243- 3 .

[25] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[26] G. Wang, Y. Sun, and J. X. Wang, "Automatic image-based plant disease severity estimation using deep learning", Computational Intelligence and Neuroscience, pp. 1-8, 2017.

[27] Evgin Goceri AG, "On The Importance of Batch Size for Deep Learning", Yildirim Kenan, editor, "International Conference on Mathematics", An Istanbul Meeting for World Mathematicians Minisymposium on Approximation Theory, Minisymposium on Mathematics Education, pp. 100–1, 2018.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from over_tting", J. Mach. Learn. Res, vol. 15, no. 1, pp. 1929-1958, 2014.

[29] Wang S-H, Tang C, Sun J, Yang J, Huang C, Phillips P, et al, "Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling", Front Neurosci, vol 12,818, 2018, doi: 10.3389/fnins.2018.00818.

[30] https://www.kaggle.com/abdallahalidev/plantvillage-dataset

[31] Kamilaris A, Prenafeta-BoldúFX. Deep learning in agriculture: a survey. Comput Electron Agric, vol 147, pp 70–90, 2018, doi: 10.1016/j.compag.2018.02.016 .