ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 11, DOI:10.29027/IJIRASE.v4.i11.2021.931-935,May 2021

Detection and Classification of Plant leaf Disease Using Convolutional Neural Network on Plant Village Dataset

V.APPLALANAIDU MAJJI
Department of Computer Science
Pondicherry university
Puducherry, India
naidu.lolugu@gmail.com

G.KUMARAVELAN
Department of Computer Science
Pondicherry university
Puducherry, India
gkumarvelanpu@gmail.com

Abstract— In the area of the image classification, the most recent generation of convolution neural networks (CNNs) produced extraordinary results. This paper explores a new approach to developing a model for identifying plant diseases, based on the classification of leaf images, using convolutional neural networks. The model is built by adjusting the number of epochs, changing the training and testing combinations, modifying the values of dropout, and rectified linear unit(Relu) functions using Keras deep learning frameworks. We have developed a convolutional neural network for 14 plants containing 38 diseases (including healthy) in which 200 images have taken per class among the plant village dataset of 44,016 images. This study automatically detected various plant leaf diseases with an average classification accuracy of 99.89 percentages.

Key Words: Plant diseases, Convolutional Neural Network, Deep learning.

I. Introduction

Progress in Artificial Intelligence Research now allows for the detection of plant disease from raw images automatically. Deep learning has seen in neural networks as a learning process. The main advantage of deep_learning is that it can select features from images automatically. In recent years, image recognition issues have seen a growing use of CNN models[1-3]. CNN is a deep neural network (DNN), which is based upon the human visual system and is used to process images. Various CNN pre-trained models have proposed for object recognition. Among them, LeNet[4], and AlexNet[5] have considered as a baseline for various tasks.
The aim of this study is as follows:
a) The proposed CNN method classifies 38 types of various plant leaf diseases with classification accuracy.
b) Recognition of different leaf diseases through automatic techniques will be helpful for farmers to decrease strenuous work to monitor a big farm.

In this paper, Section II elaborates on the Literature review about plant disease detection and classification. Section III depicts, materials and methods used in this study. Section IV describes the result and discussions. Section V gives a conclusion and further research.

II. LITERATURE REVIEW

Sladojevic et al. [6] used the CaffeNet model to detect plant diseases of various plants with their plant datasets. The authors used the data augmentation process for small datasets to train their models for real-world problems. Mohanty et al. [7] used AlexNet and GoogleNet to detect 14 crop diseases of plant village dataset. The main disadvantage is to improve CA's more diverse set of training data was needed. Amara et al. [8] proposed the LeNet model with the framework deeplearning4j to detect three diseases of banana plant leaves among 3700 images of plant village dataset. Siva et al. [9] developed a system to detect three diseases among 120 tobacco plant leaves of the self-collected dataset. Alvaro et al. [10] proposed the VGGNet and ResNet model to detect ten diseases of 5000 tomato plant leaves from the plant village dataset.

Wang et al. [11] proposed the VGGNet16 model with framework Keras to detect four diseases of apple plant leaves among 2086 images. HalilDrumus et al. [12] proposed the AlexNet and SqueezeNet model with framework Caffe for the mobile app to detect ten diseases of 5000 tomato plant leaves from the plant village dataset. This mobile app shows less accuracy as compared to other previous existing models. Oppenheim et al. [13] proposed VGGNet with framework MATLAB to detect five diseases of 2465 images of potato from their own collected dataset. Jain et al. [14] proposed the Spatial Locality method to detect two diseases of 1030 of pomegranate plant images. Arib et al. [15] recommended the CaffeNet model to detect thirteen diseases of 4511 paddy leaves.

ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 11, DOI:10.29027/IJIRASE.v4.i11.2021.931-935,May 2021

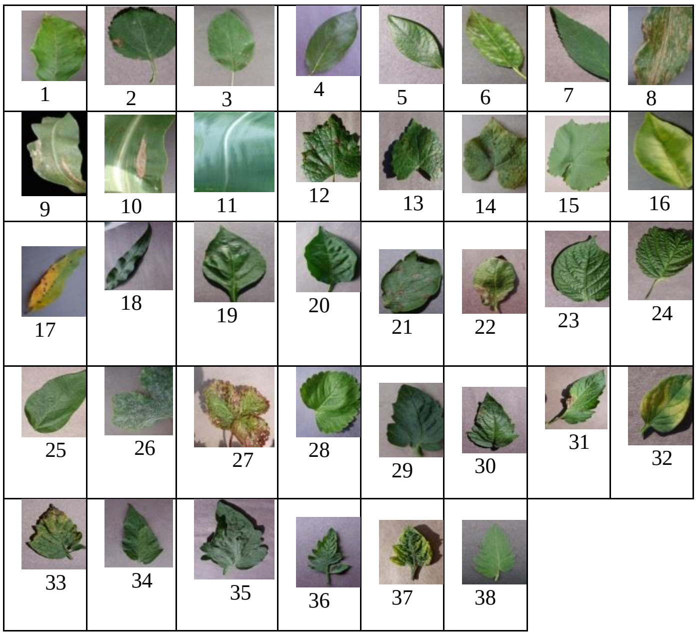

Fig.1. Various leaf disease and healthy images from the Plant Village dataset**(1)** AppleScab**(2)** Apple BlackRot**(3)** AppleCedarRust**(4)** Applehealthy **(5)** Blueberryhealthy **(6)** Cherry PowderyMildew**(7)** Cherryhealthy **(8)** CornGrayLeafSpot**(9)** CornCommonRust**(10)** Corn NorthernLeafBlight**(11)** Cornhealthy **(12)** GrapeBlackRot**(13)** GrapeBlackMeasles(Esca)**(14)** GrapeLeafBlight**(15)** GrapeHealthy **(16)** OrangeHuanglongbing **(17)** PeachBacterialSpot**(18)** Peachhealthy**(19)** PepperBacterialSpot**(20)** Pepperhealthy**(21)** Potato EarlyBlight**(22)** PotatoLateBlight**(23)** Potatohealthy **(24)** Raspberryhealthy **(25)** Soybeanhealthy **(26)** SquashPowdery Mildew **(27)** StrawberryHealthy **(28)** StrawberryLeafScorch**(29)** TomatoBacterialSpot,**(30)** TomatoEarlyBlight**(31)** TomatoLateBlight,**(32)** TomatoLeafMold **(33)** TomatoSeptoria LeafSpot **(34)** TomatoTwoSpottedSpiderMite, **(35)** TomatoTargetSpot, **(36)** TomatoMosaicVirus **(37)** TomatoYellowLeafCurlVirus **(38)** Tomatohealthy.

III. MATERIALS AND METHODS

A. Materials

Dataset: A benchmark dataset is necessary at every stage of disease detection and classification research, beginning the training stage to evaluating the work of identification algorithms. The number of images 44016 has collected from the Plant Village dataset[16]. It has divided into 38 different categories of 14 crops has shown in Fig.1. For the experimental purpose, we used 200 images per each class that means a total of 7121 images have loaded in which 6408 leaves have used for training, and the remaining 713 images have for testing.

B.Methodology:

Firstly, CNN has broadly used for object recognition problems. Nevertheless, it has extensively used in new domains such as text recognition, object tracking, detection of actions, visual detection of saliency, and labeling of sights. Now CNN has been extensively used to detect various plant leaf diseases.

The steps involved in this model as follows.

In the preprocessing, images from their original size have rescaled to the correct dimensions, if not the Keras library, unable to manage such large_scale images. In the case of plant village dataset no need to resize the dimensions because all the data are in a uniform size.

The Plant Village dataset has extended dynamically to prevent overlapping the sample. This method raises the total dataset. Numerous methods change the training data and adjust the array representations so that the data mark stays the same as data increase techniques. Some common increases are translations, horizontal flips, vertical flips, random cultivations, grayscales, turning, and jitters of color. By applying a few of these transformations to a training data set, it is simple to double or triple the number of training samples and construct an extremely robust model with no overlapping problems and more precision[17]. The applied changes are rotated, zoomed, adjusted height, and moved width.

The sequential API of Keras was used in the application when the input layer has used to insert one layer at a time. A convolutional (Conv2D) layer, like a collection of learning variables and filters, is the first layer. For conv2D layers, 32, 64, and 128 filters have selected and 1024 in the last fully connected layer. A portion of the image transforms (defined by the kernel size) using the conv2D_filter for the kernel. The kernel_filter matrix applied to the whole form. CNN can insulate features that are useful in all areas and construct task maps. The pooling layer is the second most important on CNN. In this case, MaxPool2D has used. This layer works as a filter of a downsample. The next two pixels are perceived, and the maximum value of both has calculated. Pooling has used to minimize device costs and also to some degree to eliminate overfitting. One must carefully select the pooling size as the pooling measurements increase, the smaller the sample[18].

The use of convolution and max-pooling layers allows convolution neural networks to combine the local features and to know more about the global characteristics of training images. The Dropout is regularization that arbitrarily overlooks the part of nodes in the layer (sets the weights to zero) on every exercise sample[19]. It removes a random portion of the network to learn certain distributed features. This approach eliminates overfitting and also improves widespread use.

The solution' ReLU' has the feature max(0,x) of activation. The Relu activation function has used to non-linearize the network. Using the "Flatten" layer in the model, final function maps have transformed into a single one-dimensional vector. The flattening step is necessary so that fully connected layers have used after certain convolutional/max pool layers. This layer incorporates all the local knowledge of the convolutional layers in the past defined. At last, one utilizes the apps with 1024 filters connected to two fully 'Dense ' layers. The probability distribution of neural network outputs for every class ' (Dense(38, activation="softmax ")) in this last layer[20].

IV.RESULTS AND DISCUSSIONS

Convolutional Neural Network algorithm for plant disease detection and classification has executed in python, and the conducting tests perform on google colab with inbuilt GPU: Tesla K80, 25.51GB of RAM size, and 68.40GB of hard disk size. Plant Village Dataset has used for this experimentation. The list of crops and their diseases has shown in Fig 2.

Plant Village Dataset: This dataset has downloaded from the website https://github.com/spMohanty/PlantVillage-Dataset. The dataset contains 14 crops and 38 diseases of corresponding plants. All the images are in JPEG format and have size 741MB. This dataset has 44016 images, and we selected 200 images from each folder. During classification, 7121 leaves have used for experimentation, out of which 6408 have for training, and the rest of 713 have used for testing.

Table I results indicate that improves the classification by increases the data for the training phase. The superlative trained model (train-90% and test-10%) classified 99.89% accuracy, and the poor set (train-10% and test-90%) classifies 97.02% accuracy.

Table II results indicate that the classification accuracy decreases by increasing more batch size for the training phase. The leading trained model (training-90% and testing-10%) with batch size 32 classified 99.89% accuracy and the lowest training set (training-90% and testing-10%) with batch size 128 classified 98.91% accuracy.

Table III results indicate the relationship between the testing accuracy of the improved model and dropout probability value is studied. In general, the dropout operation of probability value is choosing to be 0.5. Several experiments were performed from 0.5 to 0.7, with a 0.05 - percent difference. It has noted that the highest 99.89% and lowest 96% of classification accuracy achieved at the dropout value 0.5, and 0.7 respectively.

```
[INFO] Loading images ...
[INFO] Processing Apple___Apple_scab ...
[INFO] Processing Apple___Black_rot ...
[INFO] Processing Apple___Cedar_apple_rust ...
[INFO] Processing Apple___healthy ...
[INFO] Processing background ...
[INFO] Processing Blueberry___healthy ...
[INFO] Processing Cherry_(including_sour)___healthy ...
[INFO] Processing Cherry_(including_sour)___Powdery_mildew ...
[INFO] Processing Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot ...
[INFO] Processing Corn_(maize)___Common_rust_ ...
[INFO] Processing Corn_(maize)___healthy ...
[INFO] Processing Corn_(maize)___Northern_Leaf_Blight ...
[INFO] Processing Grape___Black_rot ...
[INFO] Processing Grape___Esca_(Black_Measles) ...
[INFO] Processing Grape___healthy ...
[INFO] Processing Grape___Leaf_blight_(Isariopsis_Leaf_Spot) ...
[INFO] Processing Orange___Haunglongbing_(Citrus_greening) ...
[INFO] Processing Peach___Bacterial_spot ...
[INFO] Processing Peach___healthy ...
[INFO] Processing Pepper,_bell___Bacterial_spot ...
[INFO] Processing Pepper,_bell___healthy ...
[INFO] Processing Potato___Early_blight ...
[INFO] Processing Potato___healthy ...
[INFO] Processing Potato___Late_blight ...
[INFO] Processing Raspberry___healthy ...
[INFO] Processing Soybean___healthy ...
[INFO] Processing Squash___Powdery_mildew ...
[INFO] Processing Strawberry___healthy ...
[INFO] Processing Strawberry___Leaf_scorch ...
[INFO] Processing Tomato___Bacterial_spot ...
[INFO] Processing Tomato___Early_blight ...
[INFO] Processing Tomato___healthy ...
[INFO] Processing Tomato___Late_blight ...
[INFO] Processing Tomato___Leaf_Mold ...
[INFO] Processing Tomato___Septoria_leaf_spot ...
[INFO] Processing Tomato___Spider_mites Two-spotted_spider_mite ...
[INFO] Processing Tomato___Target_Spot ...
[INFO] Processing Tomato___Tomato_mosaic_virus ...
[INFO] Processing Tomato___Tomato_Yellow_Leaf_Curl_Virus ...
[INFO] Image loading completed
```

Fig. 2. Loading the images from Plant Village Dataset

```
Epoch 1/200
200/200 [==============================] - 89s 443ms/step - loss: 0.1006 - acc: 0.9721 - val_loss: 0.1211 - val_acc: 0.9717
Epoch 2/200
200/200 [==============================] - 82s 410ms/step - loss: 0.0712 - acc: 0.9768 - val_loss: 0.3237 - val_acc: 0.9590
Epoch 3/200
200/200 [==============================] - 82s 409ms/step - loss: 0.0580 - acc: 0.9803 - val_loss: 0.1799 - val_acc: 0.9681
Epoch 4/200
200/200 [==============================] - 81s 407ms/step - loss: 0.0480 - acc: 0.9832 - val_loss: 0.2065 - val_acc: 0.9645
Epoch 5/200
200/200 [==============================] - 82s 409ms/step - loss: 0.0434 - acc: 0.9846 - val_loss: 0.0612 - val_acc: 0.9795
Epoch 6/200
200/200 [==============================] - 82s 408ms/step - loss: 0.0423 - acc: 0.9851 - val_loss: 0.1503 - val_acc: 0.9706
Epoch 7/200
200/200 [==============================] - 82s 408ms/step - loss: 0.0370 - acc: 0.9868 - val_loss: 0.0604 - val_acc: 0.9825
Epoch 8/200
200/200 [==============================] - 81s 407ms/step - loss: 0.0361 - acc: 0.9871 - val_loss: 0.0429 - val_acc: 0.9861
Epoch 9/200
200/200 [==============================] - 80s 402ms/step - loss: 0.0322 - acc: 0.9884 - val_loss: 0.2752 - val_acc: 0.9614
Epoch 10/200
200/200 [==============================] - 80s 401ms/step - loss: 0.0325 - acc: 0.9883 - val_loss: 0.0570 - val_acc: 0.9839
.
.
.
.
Epoch 190/200
200/200 [==============================] - 82s 410ms/step - loss: 0.0016 - acc: 0.9995 - val_loss: 0.0186 - val_acc: 0.9962
Epoch 191/200
200/200 [==============================] - 82s 410ms/step - loss: 0.0029 - acc: 0.9990 - val_loss: 0.0281 - val_acc: 0.9945
Epoch 192/200
200/200 [==============================] - 82s 410ms/step - loss: 0.0019 - acc: 0.9994 - val_loss: 0.0360 - val_acc: 0.9929
Epoch 193/200
200/200 [==============================] - 82s 409ms/step - loss: 0.0020 - acc: 0.9994 - val_loss: 0.0190 - val_acc: 0.9959
Epoch 194/200
200/200 [==============================] - 84s 418ms/step - loss: 0.0018 - acc: 0.9994 - val_loss: 0.0243 - val_acc: 0.9955
Epoch 195/200
200/200 [==============================] - 84s 418ms/step - loss: 0.0024 - acc: 0.9993 - val_loss: 0.0113 - val_acc: 0.9979
Epoch 196/200
200/200 [==============================] - 85s 424ms/step - loss: 0.0022 - acc: 0.9993 - val_loss: 0.0111 - val_acc: 0.9974
Epoch 197/200
200/200 [==============================] - 84s 421ms/step - loss: 0.0022 - acc: 0.9993 - val_loss: 0.0185 - val_acc: 0.9968
Epoch 198/200
200/200 [==============================] - 85s 423ms/step - loss: 0.0021 - acc: 0.9992 - val_loss: 0.0612 - val_acc: 0.9904
Epoch 199/200
200/200 [==============================] - 84s 422ms/step - loss: 0.0030 - acc: 0.9991 - val_loss: 0.0110 - val_acc: 0.9975
Epoch 200/200
200/200 [==============================] - 84s 422ms/step - loss: 0.0023 - acc: 0.9992 - val_loss: 0.0071 - val_acc: 0.9983
```

Fig. 3. Epochs with classification accuracies

ISSN(Online) : *2456-8910*
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 11, DOI:10.29027/IJIRASE.v4.i11.2021.931-935,May 2021

Table IV results indicate the lowest accuracy of 99.09% at 50 epoch(train-90% and test-10%). Meanwhile, the highest accuracy of 99.89% at the 200th epoch(train-90% and test-10%) as shown in Fig 3.

The classification accuracy of the proposed CNN algorithm has increased by increasing the data size(38diseases) over the existing CNN algorithm(15 diseases). The novel CNN model accuracy/loss of training and validation for 200 epochs has shown in Fig.4.

Table I. Training and Testing performance

S.No	Training	Testing	Classification accuracy
1	90	10	99.89
2	80	20	98
3	70	30	99.12
4	60	40	98.13
5	50	50	98.29
6	40	60	98.43
7	30	70	97.94
8	20	80	97.09
9	10	90	97.02

Table II Batch size performance

S.No	Batchsize	Training	Testing	classification accuracy
1	32	90	10	99.89
2	64	90	10	98.95
3	128	90	10	98.91

Table III. Dropout performance

S.No	Dropout	Training	Testing	classification accuracy
1	0.5	90	10	99.89
2	0.55	90	10	98.87
3	0.6	90	10	98.42
4	0.65	90	10	99
5	0.7	90	10	96
6	0.45	90	10	99.2

Table IV. Epoch performance

S.No	Epoch	Training	Testing	classification accuracy
1	25	90	10	99.35
2	50	90	10	99.09
3	100	90	10	99.59
4	200	90	10	99.89

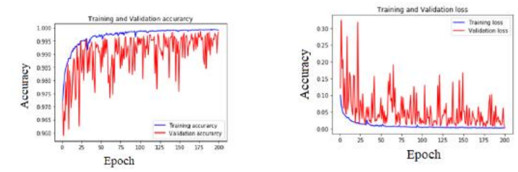

Fig.4. Model Accuracy of CNN model for the Plant Village Dataset

V. CONCLUSION

In this analysis, the neural networks model achieved high precision, 99.89 percent, when recognizing 38 forms of different leaves of plant disease. In the train-test collection, the classification accuracy obtained to a variety of sample settings with strong robustness(90% of the total data set employed for training and 10% for testing). This paper shows that enhanced recognition accuracy by adjusting the number of epochs, changing the training and testing combinations, modifying the values of dropout, and rectified linear unit(Relu) functions. In this paper, we used 200 images per each diseased class due to less RAM Supporting machine, and this is one of the drawbacks of this proposed system. In future research, we will use more RAM size to overcome the above limitation.

REFERENCES

[1]. H. A. Atabay, "Binary shape classification using convolutional neural networks". IIOAB J 7(5):332–336, 2016a.
[2]. A. K Reyes, J. C Caicedo, and J. E Camargo, "Finetuning deep convolutional networks for plant recognition". In CLEF (Working Notes).
[3]. A. M. J Hanson, A. Joy, and J. Francis, "Plant leaf disease detection using deep learning and convolutional neural network". International Journal of Engineering Science 5324.
[4]. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition". Proceedings of the IEEE 86(11):2278–2324. 1998.
[5]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks". In Advances in neural information processing systems,1097–1105. 2012.
[6].S Sladojevic, M Arsenovic, A Anderla, D Culibrk, and D Stefanovic "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image classification" Computational Intelligence and Neuroscience, Hindwai 2016.

[7]. S.P. Mohanty, D.P.Hughes and M. Salathe "Using Deep Learning for Image Based Plant Disease Detection" Journal Frontiers in Plant Science, Frontiers,2016.

[8]. J. Amara, B.Bouaziz and A Algergawy "A Deep Learning based Approach for Banana Leaf Diseases Classification" Lecture Notes in Informatics,2017.

[9]. S. K. Dasari, and V. Prasad " A novel and proposed comprehensive methodology using deep convolutional neural networks for flucured tobacco leaves classification" Springer, 2018.

[10]. A. Fuentes, S. Yoon, S. C. Kim, and D.S Park "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition**",** MDPI,2017.

[11]. G. Wang, Y. Sun and J. Wang "Automatic Image Based Plant Disease Severity Estimation Using Deep Learning", Computational Intelligence and Neuroscience, Hindwai, 2017.

[12].H. Durmu, E.O. Gunes, and M. Kirci "Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning"

[13]. D. OPPenheim and G. Shani "Disease Classification Using Convolutional Neural Networks", Advances in Animal Biosciences, 2017.

[14].L Jain, M.A. Harsha Vardhan, M. L. Nishanth and S. S Shylaja "Cloud-based system for Supervised Classification of Plant diseases using Convolutional Neural Networks", IEEE International Conference on Cloud Computing Emerging Markets, 2017.

[15]. A. A. Alfarisy, Q. Chen and M. Guo "Deep Learning Based Classification for Paddy Pests & Diseases Recognition", Association for Computing Machinery, 2018.

[16]. Plant Village Dataset downloaded from the website https://github.com/spMohanty/PlantVillage-Dataset.

[17]. Data Augmentation for deep Learning. (2019, August, 10) [Online]. Available: https://towardsdatascience.com/data-augmentation-for-deep-learning-4fe21d1a4eb9

[18]. Step wise Approach: CNN Model. (2019, July, 20) [Online]. Available: https://www.kaggle.com/sid321axn/step-wise-approach-cnn-model-77-0344-accuracy.

[19]. Understanding Dropout with the Simplified Math behind it. (2019, May, 10) [Online]. Available: https://towardsdatascience.com/simplified-math-behind-dropout-in-deep-learning-6d50f3f47275.

[20]. Fundamentals of Deep Learning – Activation Functions and When to Use Them? (2019, April, 10) [Online]. Available: https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/ .