Comparison of Deep Convolutional Neural Networks Models for Plant Disease Identification

Megha Gupta, Nupur Prakash

Abstract—Identification of plant diseases has been performed using machine learning and deep learning models on the datasets containing images of healthy and diseased plant leaves. The current study carries out an evaluation of some of the deep learning models based on convolutional neural network architectures for identification of plant diseases. For this purpose, the publicly available New Plant Diseases Dataset, an augmented version of PlantVillage dataset, available on Kaggle platform, containing 87,900 images has been used. The dataset contained images of 26 diseases of 14 different plants and images of 12 healthy plants. The CNN models selected for the study presented in this paper are AlexNet, ZFNet, VGGNet (four models), GoogLeNet, and ResNet (three models). The selected models are trained using PyTorch, an open-source machine learning library, on Google Colaboratory. A comparative study has been carried out to analyze the high degree of accuracy achieved using these models. The highest test accuracy and F1-score of 99.59% and 0.996, respectively, were achieved by using GoogLeNet with Mini-batch momentum based gradient descent learning algorithm.

Keywords—Comparative analysis, convolutional neural networks, deep learning, plant disease identification.

I. INTRODUCTION

AGRICULTURE plays a vital role in every economy worldwide. The economic development of any country depends on its agricultural production. Traditionally, farmers follow ancestral farming patterns and practices especially in developing countries like India. A large section of human population is involved in farming as it is the basic need of human beings. Agriculture sector accounts for 14% of GDP in India.

A number of activities are involved for ensuring a good yield in agriculture, including soil analysis, quality of seed used, major nutrient requirement, etc. However, a major challenge faced in crop production is in the form of plant diseases, which can result in reduction of both the quality and quantity of overall yield. It is estimated that pests and diseases lead to loss of 20 – 40% of global food production constituting a threat to food security [1]. Insect pests, bacteria, fungi and viruses are the main reasons for crop diseases. Infection with these organisms can affect the overall growth of the plants. In addition to the stunted growth, there can be decreased production of fruits, leaf deformity and increased leaf falls, etc. The organisms spread from one crop field to another field, and viruses may be transmitted with seeds from one place to another place. Traditionally, plant diseases are detected by experts having knowledge and experience in the field. However, finding an expert and contacting them is a tedious, time consuming and expensive task. It may sometimes take long time, making the eradication of the disease difficult. Now with technological advances, automatic disease detection is being attempted using image processing and computer vision techniques, machine learning and deep learning techniques. Most widely used machine learning techniques for image classification are support vector machines (SVM), artificial neural networks (ANN), and random forest.

H. Rahman et al. (2017) [2] reported random forest classifier as the best machine learning method for classification of healthy and diseased plants while comparing different machine learning approaches for identification of healthy and non-healthy plant leaves of cabbage (382 images), citrus (539 images) and sorghum (262 images). H. B. Prajapati et al. (2017) [3] presented a prototype system using K-means clustering and Support Vector Machine for detection and classification of rice diseases, namely Bacterial leaf blight, Brown spot, and Leaf smut, and achieved 93.33% accuracy on training dataset and 73.33% accuracy on the test dataset. P. Alagumariappan et al. (2020) [4] developed a real time decision support system integrated with a camera sensor module for identification of plant diseases and demonstrated that Extreme Learning Machine performed better than Support Vector Machine with linear and polynomial kernel.

Convolutional neural networks (CNN, or ConvNet), a class of deep neural networks, employ a specialized kind of linear operation called convolution, and have been used for analysis of visual images. These are being used in applications such as image and video recognition, image classification, medical image analysis, natural language processing, recommender systems, etc. Basic architecture of CNN is composed of five parts, namely input, convolution layer, pooling layer, fully connected layer, and output layer [5]. The convolution layer comprises of a number of filters having dimensions less than the dimensions of the input. The convolution operation is carried out as depicted in Fig. 1, resulting in an output known as convoluted output. If k number of filters are applied, then the output is a convolution layer of depth 'k', which is also called as 'k feature maps'. Each cell in the feature map corresponds to a neuron. The pooling layer is applied on the convoluted layer in order to reduce its dimension, while fully connected layers connect every neuron in one layer to every neuron in other layer. This is similar to the traditional multi-layer perceptron

Megha Gupta is with the University School of ICT, GGS Indraprastha University, Delhi, India (corresponding author, phone: +919811183692; e-mail: guptamegha98@gmail.com).

Nupur Prakash (Prof.) is with the University School of ICT, GGS Indraprastha University, Delhi, India (e-mails: nupurprakash1@gmail.com, nupur@ipu.ac.in).

World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
Vol:15, No:9, 2021

neural network (MLP). In convolutional layer, neurons receive input from only a restricted, typically square shaped (e.g., size 5x5), subarea of the previous layer. The input area of a neuron is denoted as its receptive field. Hence, the receptive field for a fully connected layer is its entire previous layer. On the other hand, the receptive area is smaller than the previous layer for the convolutional layer. The vector of the weights is known as filters. It represents a particular feature of the input.
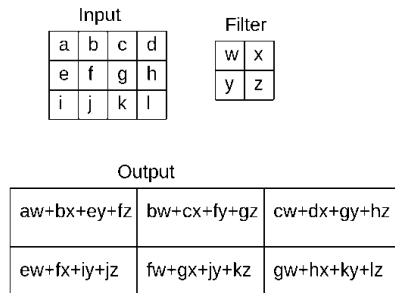

Fig. 1 Convolution operation

The main advantage of CNN is that many neurons can share the same filter. This reduces the memory footprint because in place of each receptive field having its own bias and vector weighting, all receptive fields share a single vector of weights as filter.

S. P. Mohanty et al. (2016) [6] used 54,306 images of diseased and healthy plant leaves collected under controlled conditions, available for public use in the PlantVillage dataset, to train two deep convolutional neural networks, AlexNet and GoogLeNet. They reported that GoogLeNet performed better than AlexNet and achieved an accuracy of 99.35%. K. P. Ferentinos (2018) [7] reported an error rate of 0.47% with VGG on their test dataset while comparing five convolutional neural networks architectures (AlexNet, AlexNetOWTBn, GoogleNet, Overfeat and VGG) for the identification of plant diseases using dataset containing 87,848 images, taken both laboratory conditions and real field conditions. J. Boulent et al. (2019) [8] carried out a survey of 19 studies that used CNNs for identification and classification of plant diseases, to identify some of the major shortcomings and issues in these works. In six of these studies, models for multiple crops and diseases were trained, while in the rest the specialized model focused on a single crop. J. Chen et al. (2020) [9] studied transfer learning of the deep CNNs using VGG-19 and achieved validation accuracy of 91.83% on the dataset containing 500 rice images and 466 maize images of diseased leaves.

In this paper, a comparative analysis of various deep CNN models for plant disease identification has been carried out.

II. DATASET AND METHOD

A. Dataset Description

The 'New Plant Diseases Dataset' [10], which is an augmented version of publicly available PlantVillage dataset [11] has been used in the comparative analysis. This dataset comprises of 87,867 images and is available on Kaggle platform. The dataset is already divided into training and validation sets in the ratio of 80:20. Apart from these images, another folder containing 33 images is available for testing. However, as the images for testing are less, all the images from training set, validation set and testing set are combined into one folder. This new folder is named as 'plant disease identification dataset (updated)' which contains total 87,900 images. The total number of classes are 38 (Fig. 2) for 14 unique plants. The total number of diseased plants are 26, as against 12 healthy plants. This new dataset is divided into three parts: training, validation and test set, in the ratio of 70:10:20.

The data related to 14 unique plants is considered, these are, Tomato, Bell pepper, Orange, Corn (maize), Grape, Potato, Strawberry, Peach, Apple, Squash, Blueberry, Cherry (including sour), Raspberry, Soybean. The number of images in each class are shown in Table I.

B. Methodology

The dataset 'plant disease identification dataset (updated)' is pre-processed. Standard and well-known CNN architectures (AlexNet, ZFNet, VGGNet, GoogLeNet, ResNet) are trained, validated and tested on the dataset. Mini-batch momentum based gradient descent is used as the learning algorithm. Various evaluation metrics such as Accuracy, Loss, Precision, Recall and F1-Score, are used for comparing performance of these CNN models.

III. CNN MODELS

A. AlexNet

Alex Krizhevsky et al. [12] proposed in 2012 one of the first deep CNN architecture, the AlexNet, that showed some excellent results for image classification and recognition tasks. It has an eight-layer architecture consisting of five convolution layers and three fully connected layers. It is operated with 3-channel images (RGB images) that are (224x224x3) in size. Max pooling is used for subsampling. To improve the convergence rate by alleviating the problem of vanishing gradient to some extent, ReLU has been used as a non-saturating activation function. In the initial layers, large size filters (11x11 and 5x5) are used, which are followed by smaller size (3x3) filter size for rest of the layers, and 3x3 kernels for max pooling. Ninety-six filters of size 11x11 with stride 4 are used in first conv layer, followed by 256 filters of size 5x5 with stride 1 in second conv layer, 384 filters of size 3x3 with stride 1 in third and fourth conv layers, and 256 filters of size 3x3 with stride 1 in fifth conv layer. Each convolution layer is followed by batch normalization. Dropout is also used in the first two fully-connected layers to reduce overfitting. Softmax as activation function is employed for output layer and number of trainable parameters is 60 million (M).

B. ZFNet

Zeiler and Fergus in 2013 [13] proposed another CNN model, the ZFNet, while working on AlexNet. It was developed with the idea to quantitatively visualize the network activity and monitor performance of the CNN model by interpreting the neurons activation. Experimental validation of this idea of feature visualization was carried out by the authors on AlexNet.

World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
Vol:15, No:9, 2021

It was noted that in the first and second layers of the network, only a few neurons were active while others were inactive. Based on these findings, CNN topology was adjusted and parameter optimization was carried out. The learning of CNN model was maximized by decreasing filter sizes and stride to retain maximum number of features in initial two convolutional layers.

TABLE I
NUMBER OF IMAGES FOR EACH CLASS OF THE DATASET

Class	Scientific Name	Number of images
Apple (Apple scab)	*Venturia inaequalis*	2523
Apple (Black rot)	*Botryosphaeria obtusa*	2484
Apple (Cedar rust)	*Gymnosporangium juniperi-virginianae*	2204
Apple healthy	-	2510
Blueberry healthy	-	2270
Bell pepper (Bacterial spot)	*Xanthomonas campestris*	2391
Bell pepper healthy	-	2485
Cherry healthy	-	2282
Cherry (Powdery mildew)	*Podoshaera clandestine*	2104
Corn (Common rust)	*Puccinia sorghi*	2387
Corn (Gray leaf spot)	*Cercospora zeae-maydis*	2052
Corn healthy	-	2324
Corn (Northern leaf blight)	*Exserohilum turcicum*	2385
Grape (Black measles (Esca))	*Phaeomoniella aleophilum, Phaeomoniella chlamydospora*	2400
Grape (Black rot)	*Guignardia bidwellii*	2360
Grape healthy	-	2115
Grape (Leaf blight)	*Pseudocercospora vitis*	2152
Orange Haunglongbing (Citrus greening)	*Candidatus Liberibacter spp.*	2513
Peach (Bacterial spot)	*Xanthomonas campestris*	2297
Peach healthy	-	2160
Potato (Early blight)	*Alternaria solani*	2429
Potato healthy		2282
Potato (Late blight)	*Phytophthora infestans*	2424
Raspberry healthy	-	2226
Soybean healthy	-	2527
Squash (Powdery mildew)	*Erysiphe cichoracearum*	2170
Strawberry (Leaf scorch)	*Diplocarpon earlianum*	2218
Strawberry healthy	-	2280
Tomato (Bacterial spot)	*Xanthomonas campestris pv. Vesicatoria*	2127
Tomato (Early blight)	*Alternaria solani*	2406
Tomato healthy	-	2411
Tomato (Late blight)	*Phytophthora infestans*	2314
Tomato (Leaf mold)	*Passalora fulva*	2352
Tomato (Mosaic virus)	-	2238
Tomato (Septoria leaf spot)	*Septoria lycopersici*	2181
Tomato (Target spot)	*Corynespora cassiicola*	2284
Tomato (Two-spotted spider mite)	*Tetranychus urticae*	2176
Tomato (Yellow leaf curl virus)	-	2457

Fig. 2 Example of leaf images from the 'plant disease identification dataset (updated)' dataset. (Left to right, then top to bottom) Apple scab, Corn (Common rust), Grape (Black rot), Blueberry healthy, Apple (Black rot), Apple healthy, Grape healthy, Apple (Cedar rust), Cherry healthy, Cherry (Powdery mildew), Corn (Gray leaf spot), Corn healthy, Corn (Northern leaf blight), Grape (Black measles (Esca)), Grape (Leaf blight), Orange Haunglongbing (Citrus greening), Peach (Bacterial spot), Peach healthy, Bell pepper (Bacterial spot), Bell pepper healthy, Potato (Early blight), Potato healthy, Potato (Late blight), Raspberry healthy, Soybean healthy, Squash (Powdery mildew), Strawberry healthy, Strawberry (Leaf scorch), Tomato (Bacterial spot), Tomato (Early blight), Tomato healthy, Tomato (Late blight), Tomato (Leaf mold), Tomato (Septoria leaf spot), Tomato (Two-spotted spider mite), Tomato (Target spot), Tomato (Mosaic virus), Tomato (Yellow leaf curl virus)

It also has an eight-layer architecture, five convolution and three fully connected layers, and is operated using three-channel images (RGB images) that are (224x224x3) in size. Max pooling is used for subsampling and ReLU is employed as in AlexNet. In the first convolutional layer, ninety-six filters of size 7x7 with stride 2 are used, while in second convolutional layer, 256 filters of size 5x5 with stride 2 are used. Rest of the convolutional layers are similar to AlexNet in filter size and stride, only the number of filters is increased. 512, 1024, and

World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
Vol:15, No:9, 2021

512 filters are used in third, fourth and fifth convolutional layers, respectively. Each convolution layer is followed by batch normalization and dropout is used in the first two fully-connected layers to reduce overfitting. Softmax as activation function is employed for output layer and the number of trainable parameters is 104 M.

C. VGGNet

VGGNet was proposed by Simonyan and Zisserman in 2014 [14]. Based on the ZFNet findings, the authors replaced the large-size filters (11x11 and 5x5 filters) with small size filters consisting of a stack of 3x3 filters and demonstrated that such use of small size filters can induce the effect of the large size filter. This also decreased the computational complexity by decreasing the number of parameters, and set a new trend to work with smaller size filters in CNN models. Four different ConvNet configurations (VGG11, VGG13, VGG16, VGG19) were proposed. The input to the model is a fixed-size 224 x 224 RGB image. It has same back-to-back multiple convolutional layers and some intermediate max pooling layers. Throughout the network the filter size is 3x3. The convolution stride is fixed to 1 pixel. The spatial padding is 1 for 3x3 convolutional layers so that the spatial resolution is preserved after convolution. Max-pooling is performed over a 2x2 pixel window with stride 2. Number of trainable parameters are 133 M in VGG-11 and VGG-13 is 133 M, 138 M in VGG-16, and 144M in VGG-19.

D. GoogLeNet

GoogLeNet (also known as InceptionNet) was proposed by Christian Szegedy et al. [15] in 2014 with the objective to achieve high accuracy with lesser computational cost. A new concept of 'inception block' was introduced which while using split, transform and merge idea, carried out multi-scale convolutional transformations. Fig. 3 shows the basic architecture of inception block. It consists of filters of different sizes (1x1, 3x3, and 5x5) so as to have spatial information at different scales, both at fine and coarse grain level. This idea helped in the learning of various types of variations present in the same category of images having different resolutions.

It is a 22 layers deep network. Inception module is the building block of GoogLeNet, and full architecture is a series of these Inception modules. Inception Module has 1x1 convolutions followed by 3x3 and 5x5 convolutions, 3x3 max pooling followed by 1x1 convolution, and directly 1x1 convolutions. Inception Module differ in number of filters across different layers. Each Inception layer counts for two convolutional layers. Number of trainable parameters is 6 M (approx.), hence 12 times less parameters than AlexNet. It introduced 1x1 convolutions which reduces the number of computations.
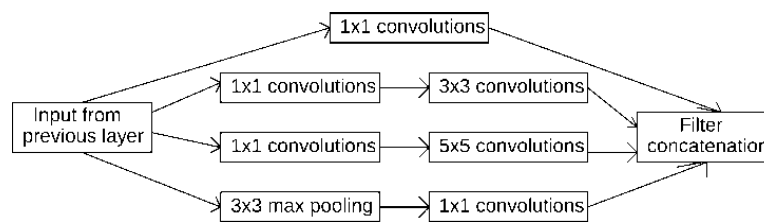

Fig. 3 Inception Module

E. ResNet

Kaiming He et al. in 2015 [16] proposed a revolutionary CNN model, the ResNet. It introduced the concept of residual learning in CNNs. Use of residual connections led to development of an efficient methodology for the training of deep networks (Fig. 4). Three different ResNet models were proposed (ResNet-50, ResNet-101, ResNet-152).

There are mostly 3x3 size filters in the convolutional layers. Two simple design rules are followed: (1) the layers have the same number of filters for the same output feature map size; and (2) the number of filters is doubled if the feature map size is halved, to preserve the time complexity per layer.

Even though the ResNet-152 has less computational complexity, it is 20 times and 8 times deeper than AlexNet and VGG respectively. After every three layers, the residual connection (shortcut connection) is added. These shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers. Down-sampling is performed by conv3_1, con4_1, and conv5_1 with a stride of 2. Number of trainable parameters are 23 M, 42 M, and 58 M in ResNet-50, ResNet-101 and ResNet-152 respectively.
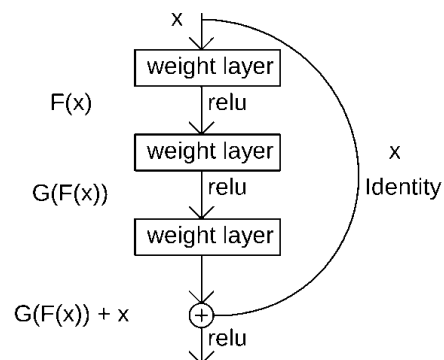

Fig. 4 Residual block

IV. IMPLEMENTATION

For implementing these deep CNN models, the Google drive is mounted on Google Colaboratory notebook [17]. The zip file of 'Plant disease identification (updated)' dataset is extracted from the google drive and the Python libraries are imported including NumPy, Pandas, Matplotlib, PyTorch [18], etc. Thereafter, Mlflow [19] is installed and used for logging metrics and hyperparameters.
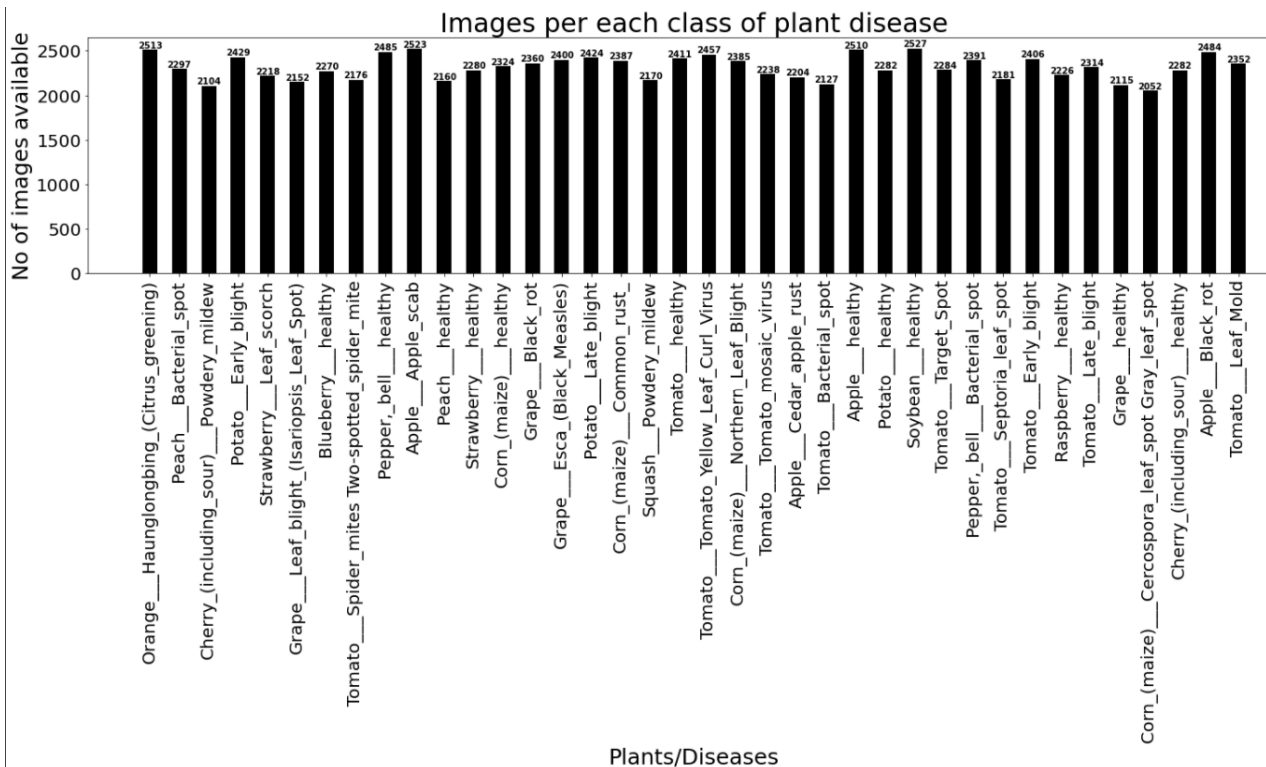
World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
Vol:15, No:9, 2021

Fig. 5 Number of images per class of the dataset

A. Pre-Processing Data

The number of unique plants, number of diseases and the number of images present for each class are calculated and plotted as shown in Fig. 5.

The images have been resized to 224 x 224 (from 256 x 256) using *Resize()* function that is available in *transforms* package of *torchvision* library. The Dataset is split into three parts: training set (61,530 images), validation set (8,790 images) and testing set (17,580 images).

B. Training and Testing

The architectures of AlexNet, ZFNet, VGGNet, GoogLeNet, and ResNet has been created using PyTorch. Since this problem deals with classification, therefore cross entropy loss function has been used. The training loop is defined (fit function). All the models are trained by calling the fit function. Hyperparameters are taken as: batch size = 32, learning rate = 0.001, momentum = 0.9, number of epochs = 25. After each epoch, training loss, training accuracy, validation loss and validation accuracy are calculated and logged. After training, test accuracy and test loss are calculated. Precision, recall and F1-score are also calculated.

V. RESULTS

A. Discussion

This section presents the comparative analysis of deep learning convolutional based neural network architectures in order to select the best model for the augmented version of the PlantVillage dataset i.e., '*Plant Disease identification*

(updated)' dataset. Training accuracy, validation accuracy, training loss, validation loss, testing accuracy, testing loss, precision, recall and F1-score were used for evaluating the results obtained. The performance metric, F1-score, is considered as the most important evaluation metric. It is the harmonic mean of Precision and recall. Therefore, for the task of identification of plant diseases, the model that achieved the highest F1-score was considered to be most suitable. It was observed that training the models for about 25 epochs the training as well as the validation accuracy and loss were converged. The performances of deep learning architectures are represented by line graphs (Figs. 6 and 7). Table II shows the overall performance of deep CNN architectures.

B. Comparative Analysis of Deep CNN Models

As presented in Fig. 6 and Fig. 7, the performance of deep CNN architectures indicates that overfitting and underfitting has not occurred. Since the dataset used is balanced unlike PlantVillage dataset, the models achieved higher accuracy and low loss rate after training them for 25 epochs. Overall, for the purpose of comparative study ten different CNN architectures were considered. From Table II, Figs. 6 and 7, a few observations were made:

- The GoogLeNet model attained the highest F1-score, precision, recall, test accuracy and lowest test loss among all the models. It achieved second highest validation accuracy i.e., 0.0002 value less than the highest validation accuracy attained by VGG13 model and second lowest validation i.e., 0.004 value more than the lowest validation loss attained by again VGG-13 model. Therefore, for the

World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
Vol:15, No:9, 2021

augmented version of the PlantVillage dataset, GoogLeNet can be considered as the best CNN architecture among all the models used for doing analysis. It implies that the concept of adding Inception blocks in the architecture is useful for obtaining higher identification results. Moreover, this model has least number of trainable parameters i.e., 6 M among all the models considered. Therefore, the training is faster for this model as compared to others.
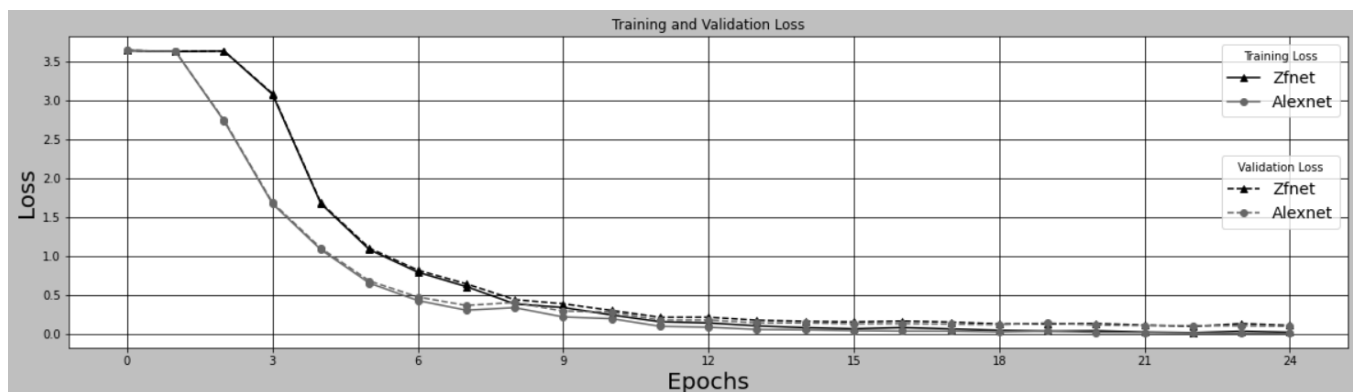
- VGG-13, VGG-19 and ResNet-50 attained the second highest F1-score, precision and recall. However, VGG-16 and ResNet-152 attained the third highest F1-score, precision and recall i.e., 0.001 value less than the second highest F1-score, precision and recall. ResNet-50 attained third highest validation accuracy followed by VGG-19, VGG-16, ResNet-152, VGG-11 and ResNet-101. ResNet-152 and VGG-11 models attained same validation accuracy. VGG-19 attained the third lowest validation loss followed by ResNet-50, VGG-16, ResNet-152 and VGG-11. ResNet-50 achieved the second highest test accuracy and second lowest test loss followed by VGG-19, VGG-13, VGG-16, ResNet-152 and VGG-11. Therefore, all these models gave comparable results. As ResNet-50 has 23 M trainable parameters that is around 6x times less than the number of trainable parameters for all the four models of VGG analyzed. Since ResNet-101 and ResNet-152 have more number of convolution layers, and therefore has more number of trainable parameters as compared to ResNet-50, which in turn increased the training time. Therefore, ResNet-50 can be considered as second best model for plant disease identification dataset.
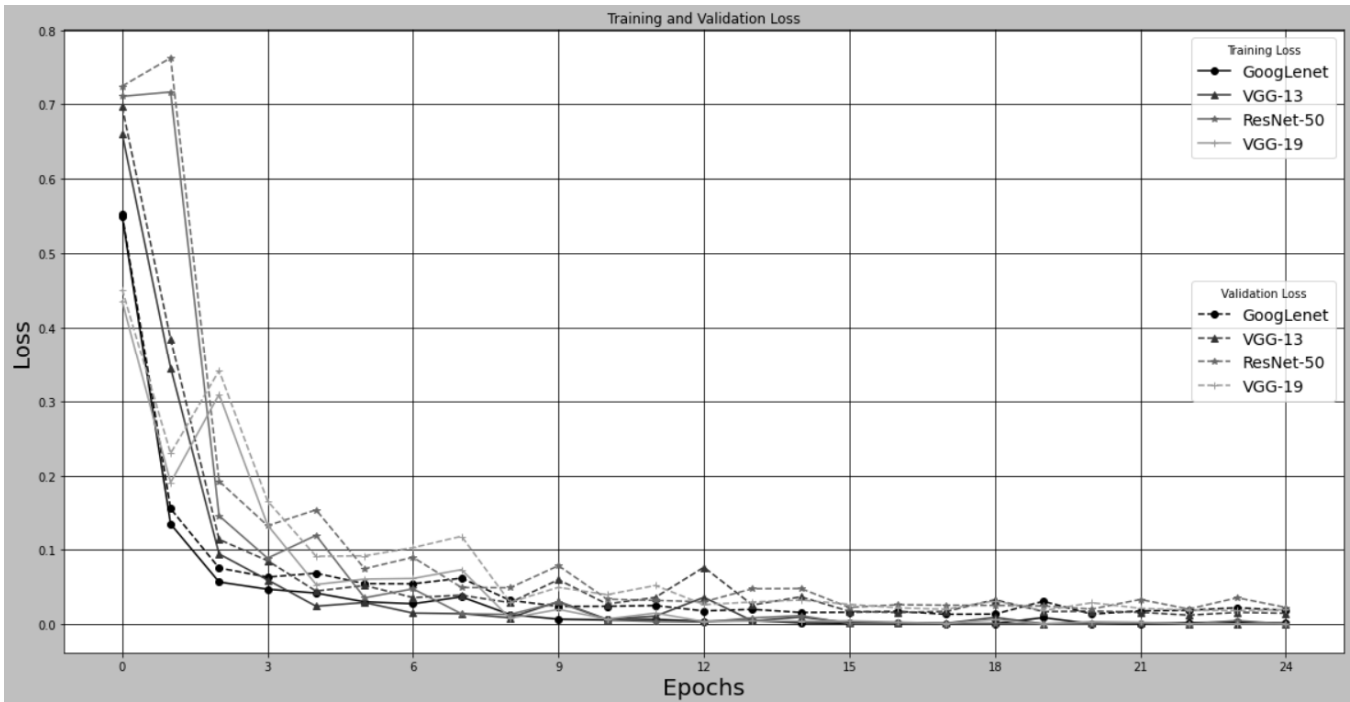
- VGG-13, VGG-19 and ResNet-50 attained comparable results. However, the number of trainable parameters in VGG-13 is less than VGG-19. Therefore, VGG-13 can be considered as third best model for plant disease identification dataset.

- ZFNet attained the least F1-score, precision and recall among all the models analyzed, followed by AlexNet. Both the models took five to six more epochs to converge as compared to other models analyzed as shown in the Figs. 6 and 7. They attained low validation and test accuracy and high validation and test loss among all the models.

TABLE II
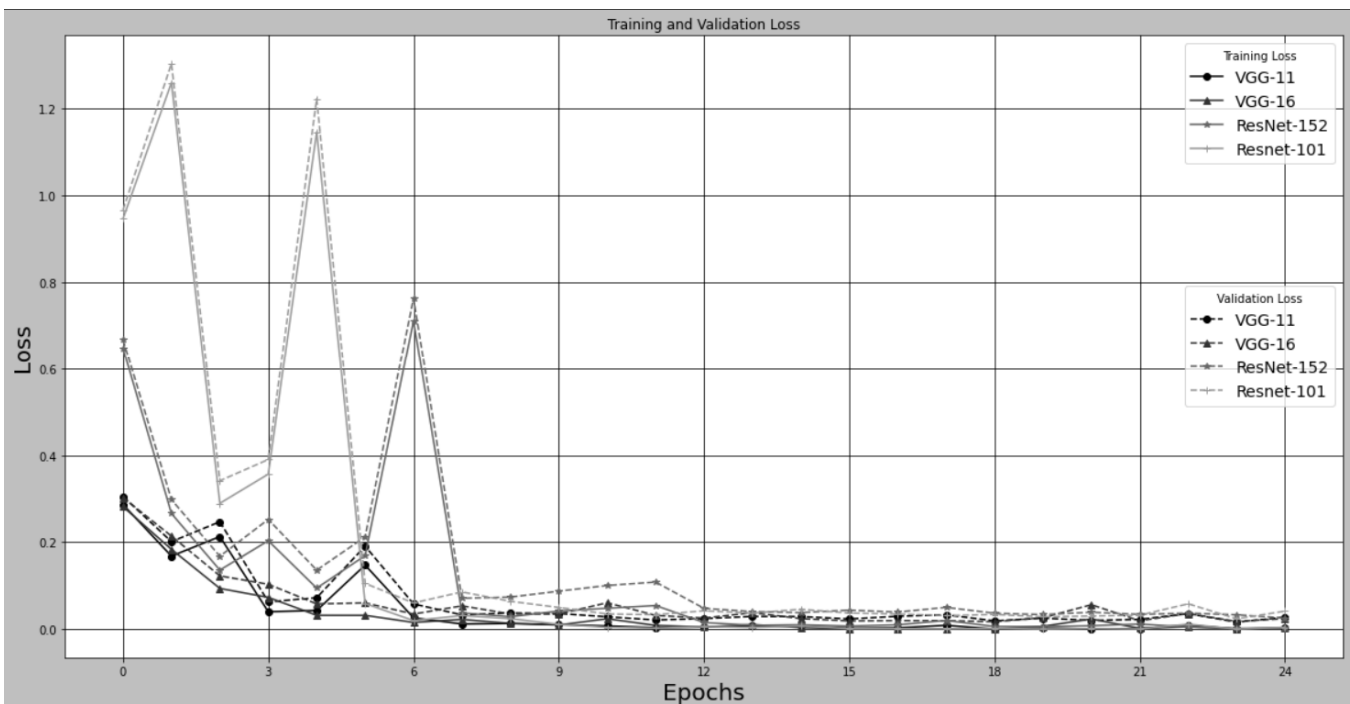PERFORMANCE OF VARIOUS DEEP LEARNING ARCHITECTURES

Deep Learning Architectures	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Precision	Recall	F1-score	Test Accuracy	Test Loss
AlexNet	0.9968	0.0110	0.9696	0.1019	0.968	0.967	0.967	0.9674	0.1128
ZFNet	0.9924	0.0250	0.9637	0.1139	0.962	0.961	0.961	0.9612	0.1269
VGG-11	0.9991	0.0026	0.9917	0.0277	0.992	0.991	0.992	0.9916	0.0326
VGG-13	0.9996	0.0010	**0.9950**	**0.0148**	0.994	0.994	**0.994**	0.9938	0.0199
VGG-16	0.9992	0.0022	0.9929	0.0244	0.993	0.993	0.993	0.9931	0.0247
VGG-19	0.9998	0.0006	0.9935	0.0197	0.994	0.994	**0.994**	0.9940	0.0200
GoogLeNet	0.9995	0.0015	**0.9948**	**0.0188**	**0.996**	**0.996**	**0.996**	**0.9959**	**0.0136**
ResNet-50	**0.9999**	**0.0004**	0.9938	0.0226	0.994	0.994	**0.994**	0.9944	0.0197
ResNet-101	0.9987	0.0046	0.9885	0.0416	0.988	0.988	0.988	0.9878	0.0407
ResNet-152	0.9992	0.0031	0.9917	0.0257	0.993	0.993	0.993	0.9927	0.0250

(a) AlexNet, ZFNet

World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
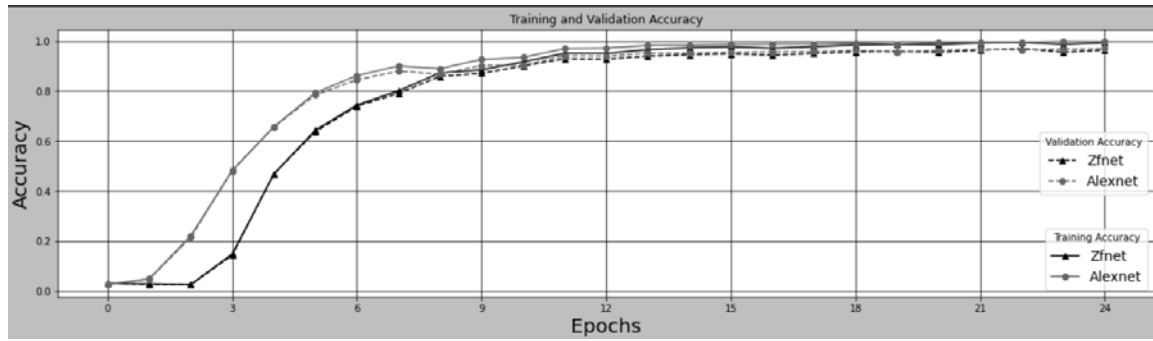Vol:15, No:9, 2021

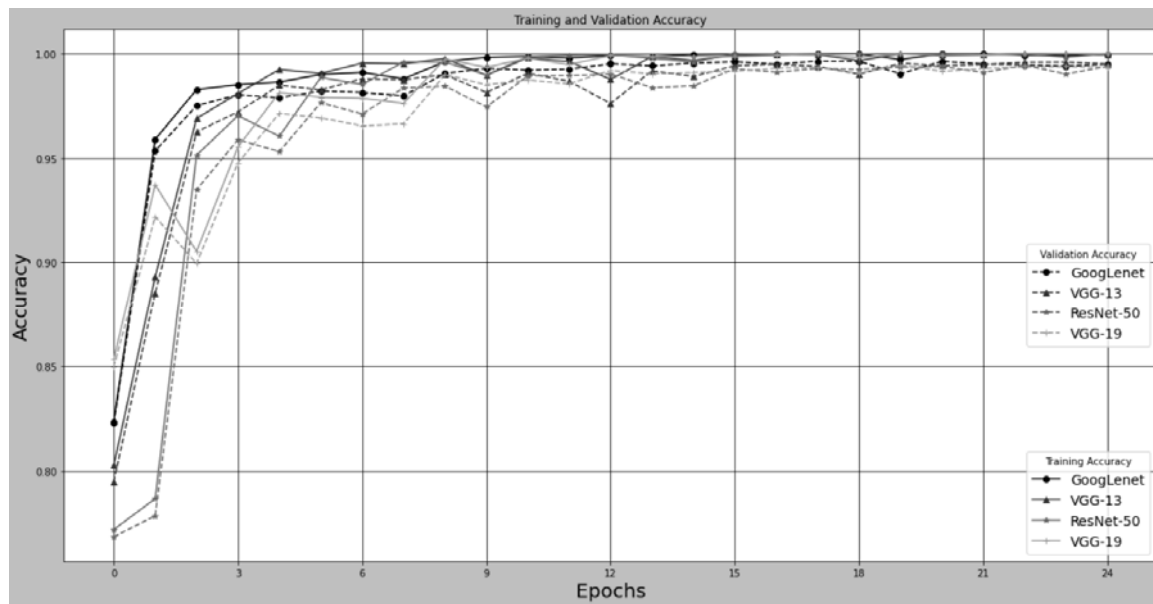(b) GoogLeNet, VGG-13, ResNet-50, VGG-19
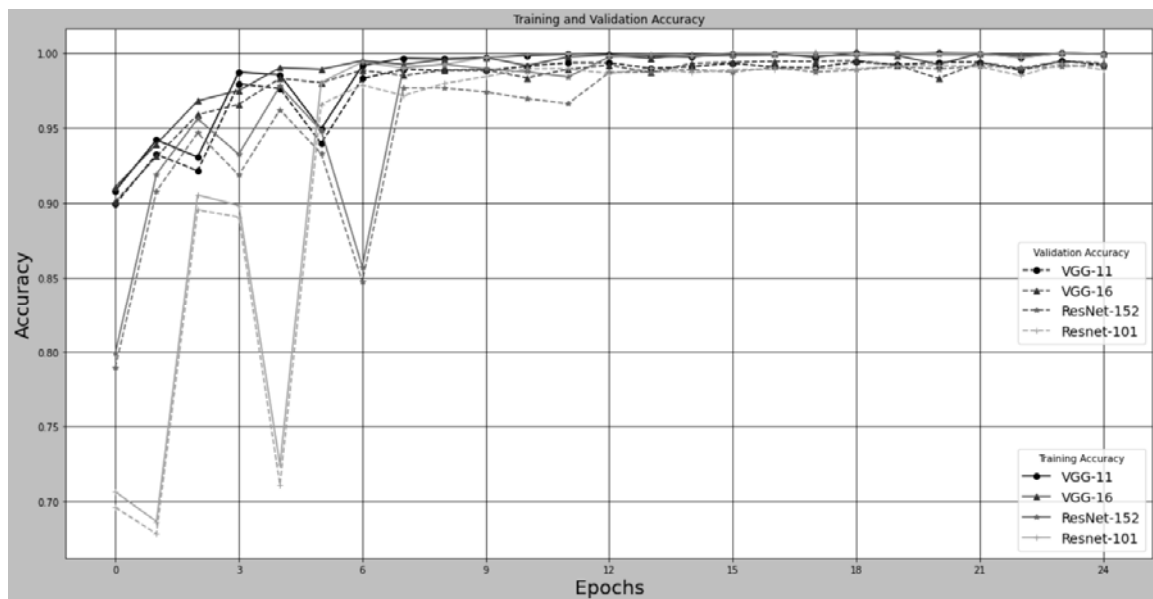

(c) VGG-11, VGG-16, ResNet-152, ResNet-101

Fig. 6 Training loss and Validation loss after each epoch of (a) AlexNet, ZFNet, (b) GoogLeNet, VGG-13, ResNet-50, VGG-19, (c) VGG-11, VGG-16, ResNet-152, ResNet-101

World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
Vol:15, No:9, 2021

(a) AlexNet, ZFNet

(b) GoogLeNet, VGG-13, ResNet-50, VGG-19

(c) VGG-11, VGG-16, ResNet-152, ResNet-101

Fig. 7 Training accuracy and Validation accuracy after each epoch of (a) AlexNet, ZFNet, (b) GoogLeNet, VGG-13, ResNet-50, VGG-19, (c)
VGG-11, VGG-16, ResNet-152, ResNet-101

World Academy of Science, Engineering and Technology
International Journal of Agricultural and Biosystems Engineering
Vol:15, No:9, 2021

VI. Conclusion

In this paper, a comparative analysis has been performed among various standard well known deep learning models for plant disease identification task. It was found that GoogLeNet attained the highest F1-score, test accuracy and lowest test loss and has lowest number of trainable parameters among all the models. Therefore, the most suitable model for the identification of plant diseases was found out to be GoogLeNet among all the models analyzed. ResNet-50 can be considered as second best model followed by VGG-13. However, in future the performance of other standard and modified versions of convolutional models can also be analyzed. Various deep learning based optimizers such as Rmsprop, Adam, Adagrad and Adadelta can be used in future to enhance the performance of the models for plant disease identification.

References

[1] "Plant Health and Food Security", *Food and Agriculture Organization of the United Nation, International Plant Protection Convention*, 2017.

[2] H. Rahman et al.., "A comparative analysis of machine learning approaches for plant disease identification," in *Advancements in Life Sciences – International Quarterly Journal of Biological Sciences*, pp. 120-126, Aug 2017.

[3] H. B. Prajapati, J. P. Shah and V. K. Dabhi, "Detection and Classification of Rice Plant Diseases," in *Intelligent Decision Technologies, IOS Press*, pp. 357-375, 29 Aug 2017.

[4] P. Alagumariappan et al.., "Intelligent Plant Disease Identification System Using Machine Learning," in *Eng. Proc.*, 14 Nov 2020.

[5] A. Khan, A. Sohail, U. Zahoora and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," in *Artificial Intelligence Review*, 21 Apr 2020.

[6] S. P. Mohanty, D. Hughes and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection", *Frontiers in Plant Science*, vol. 7, 2016.

[7] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," in *Computers and Electronics in Agriculture*, pp. 311-318, 2018.

[8] J. Boulent, S. Foucher, J. Théau and P. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases," in *Front. Plant Sci.*, 23 July 2019.

[9] J. Chen, J. Chen, D. Zhang, Y. Sun and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," in *Computers and Electronics in Agriculture*, 2020.

[10] S. Bhattarai, "New Plant Diseases Dataset," *Kaggle.com*, 2018. (Online). Available: https://www.kaggle.com/vipoooool/new-plant-diseases-dataset. (Accessed: 21- Apr- 2021).

[11] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv:1511.08060*, 2015.

[12] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.

[13] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *arXiv:1311.2901v3 (cs.CV)*, 28 Nov 2013.

[14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.21556v6 (cs.CV)*, 10 Apr 2015.

[15] C. Szegedy et al.., "Going deeper with convolutions," *arXiv:1409.4842v1 (cs.CV)*, 17 Sep 2014.

[16] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385v1 (cs.CV)*, 10 Dec 2015.

[17] "Google Colaboratory", *Colab.research.google.com*. (Online). Available: https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index#recent=true. (Accessed: 01- May- 2021).

[18] "PyTorch documentation – PyTorch 1.8.1 documentation," *PyTorch.org*. (Online). Available: https://pytorch.org/docs/stable/index.html. (Accessed: 03- May- 2021).

[19] "Quickstart – Mlflow 1.17.0 documentation," *Mlflow.org*. (Online). Available: https://www.mlflow.org/docs/latest/quickstart.html. (Accessed: 08- May- 2021).