



Identification of plant diseases using convolutional neural networks

Sachin B. Jadhav^{1,2} · Vishwanath R. Udupi³ · Sanjay B. Patil⁴

Received: 22 June 2019 / Accepted: 25 January 2020 / Published online: 7 February 2020
© Bharati Vidyapeeth's Institute of Computer Applications and Management 2020

Abstract Plant pathologists desire an accurate and reliable soybean plant disease diagnosis system. In this study, we propose an efficient soybean diseases identification method based on a transfer learning approach by using pretrained AlexNet and GoogleNet convolutional neural networks (CNNs). The proposed AlexNet and GoogleNet CNNs were trained using 649 and 550 image samples of diseased and healthy soybean leaves, respectively, to identify three soybean diseases. We used the five-fold cross-validation strategy. The proposed AlexNet and GoogleNet CNN-based models achieved an accuracy of 98.75% and 96.25%, respectively. This accuracy was considerably higher than that for conventional pattern recognition techniques. The experimental results for the identification of soybean diseases indicated that the proposed model achieved highest efficiency.

Keywords Deep CNN · Machine learning · AlexNet · GoogleNet · Detection

✉ Sachin B. Jadhav
sachinbjadhav84@gmail.com

Vishwanath R. Udupi
vrudupi1@gmail.com

Sanjay B. Patil
patilsbp@gmail.com

¹ Department of E and TC Engineering, Bharati Vidyapeeth's College of Engineering, Kolhapur, India

² VTU, Belagavi, Karnataka, India

³ Maratha Mandala Engineering College, Belgaum, Karnataka, India

⁴ Shri. Chattrapati Shivajiraje College of Engineering, Bhor, Pune, India

1 Introduction

Soybean crops are highly affected by diseases, which causes intense losses in agriculture economy [1]. For instance, bacterial blight, frog-eye leaf spot (FLS), and brown spot, are the most common diseases that cause considerable damage to crops and a decrease in yield [1, 2]. The proposed pretrained AlexNet and GoogleNet convolutional neural networks (CNNs) model was used for the classification of these three common diseases. Thus, accurate identification and diagnosis of soybean diseases are vital for high crop yield. In the naked eye approach, which is usually preferred by plant pathologists for detecting soybean diseases, subjective bias can occur because the decision is based on the experience and knowledge of experts [2].

To obtain accurate diagnosis results, several researchers have deliberated automated soybean diseases diagnosis based on digital image processing [3], pattern recognition [4], and computer vision [6]. Moreover, these advanced techniques are used to analyze various fruit and crop species, such as grapes [7], pomegranates [8], tomato [9], maize [6], and wheat [19].

Deep learning is a new trend in machine learning (ML), and it achieves state-of-the-art results in many research fields, such as computer vision, drug design, and bioinformatics [11, 17]. Deep learning enables the direct use of raw data without using handcrafted features [11, 12]. In recent years, deep learning has been extensively studied in computer vision, and therefore, a large number of related approaches have emerged [13]. Yang Lu et al. [16] proposed a system based on CNNs to recognize 10 common diseases which distinguish between rice blast, rice false smut, rice brown spot, rice bakanae disease, rice sheath blight, rice sheath rot, rice bacterial leaf blight, rice

bacterial sheath rot, rice seeding blight and rice bacterial wilt.; it achieves an accuracy of 95.45%.

Mohanty et al. [20] used pre-trained AlexNet CNN, for disease classification using transfer learning approach. The proposed system was able to classify 26 different diseases in 14 crop species using database of 54,306 images with a classification accuracy of 99.35%.

Konstantinos P. Ferentinos [21] presented convolutional neural network models to perform plant and disease, detection and classification task using simple leaves images of healthy and diseased plants, it achieves accuracy of 99.53%.

Aravind Krishnaswamy et al. [22] presented AlexNet and VGG16 CNNs models to identify tomato disease which achieves 97.29% for VGG16 net and 97.49% for AlexNet.

This study aims to introduce CNN transfer learning as an approach for classifying three soybean plant diseases according to sample leaf images. This study presents two main contributions in plant disease classification: (1) implementation of the transfer learning technique by using the already trained AlexNet and GoogleNet CNN models on a large data set and (2) identification of accurate disease symptoms in the soybean infected leaves by using the proposed AlexNet and GoogleNet CNN models, which could assist plant pathologists in diagnosing diseases.

2 Materials and methods

2.1 Materials

2.1.1 Data set

The proposed pretrained AlexNet and GoogleNet deep CNNs were used to classify defined test images from a test database. Data of soybean images were collected from soybean fields in Kolhapur district, Maharashtra, India. In this study, 80 testing data sets of soybean leaf images were used for testing the AlexNet and GoogleNet CNNs. The training data sets for AlexNet consisted of 199 bacterial blight disease images, 200 FLS disease images, 150 brown spot disease images, and 100 non-disease (healthy) images. The GoogleNet training data consisted of 150 bacterial blight disease images, 150 FLS disease images, 100 brown spot disease images, and 150 non-disease (healthy) images. We labeled black blight disease as class 1, Brown spot disease as class 2, Frogeye leaf spot disease as class 3, and healthy as class 4. Fig. 1 depicts the leaf samples of the testing data.

The summary of our data sets for the AlexNet and GoogleNet CNNs are provided in Tables 1 and 2, respectively. The total number of sample images in our data sets

was 649 for AlexNet and 550 GoogleNet. The images were fragmented into three disease categories and one non-disease category.

2.2 Methods

In this study, two pre-trained deep learning models, namely GoogleNet and AlexNet, were used for the classification of soybean diseases through the transfer learning approach. In the first phase of the study, the preprocessed images were applied as input to the proposed pretrained GoogleNet CNN architecture. The proposed models were retrained for classifying the four class categories of objects from the defined disease data set. The last layer was reconfigured and modified to the 4, which is set to the defined number of class categories (Fig. 3). The four class categories in this study consisted of three disease classes, namely bacterial blight, brown spot, and FLS, and one healthy class.

The last three layers of the GoogleNet model were modified. To increase the performance of the proposed models, some parameters of the CNNs were modified. The modification included setting the learning rate of the models as 0.0001 and setting the bias learning rate as 20 for the four fully connector layers. The minibatch size was set to 64, the number of epochs was fixed to 30, and the number of iterations was set to 150. A minibatch was obtained by splitting the training data set into batches, and the gradient descent was applied for a model coefficient update. This resulted in an overall classification accuracy of 96.25% with the GoogleNet deep neural network.

2.2.1 Architecture of the AlexNet and GoogleNet deep CNN models

The AlexNet and GoogleNet CNNs were tested in the experiment problem, which involved the identification of soybean plant diseases from their leaf images. A CNN passes a raw image through the network layers and provides a final class as an output. The proposed AlexNet and GoogleNet networks consisted of 25 and 145 layers, respectively, with each layer network learning to detect different features. Filters were then applied to each training image at different resolutions, and the output of each convolved image was used as the input to the next layer. Brightness and edge features were detected. The complexity of features that uniquely define the leaf object increases as the layers progress. Figure 2 shows the proposed pretrained AlexNet and GoogleNet general CNN model included three main neural layers, namely convolutional layers, pooling layers, and fully connected layers. The three commonly used neural layers are discussed as follows [15, 16]:

Fig. 1 Random samples output of testing AlexNet CNN model

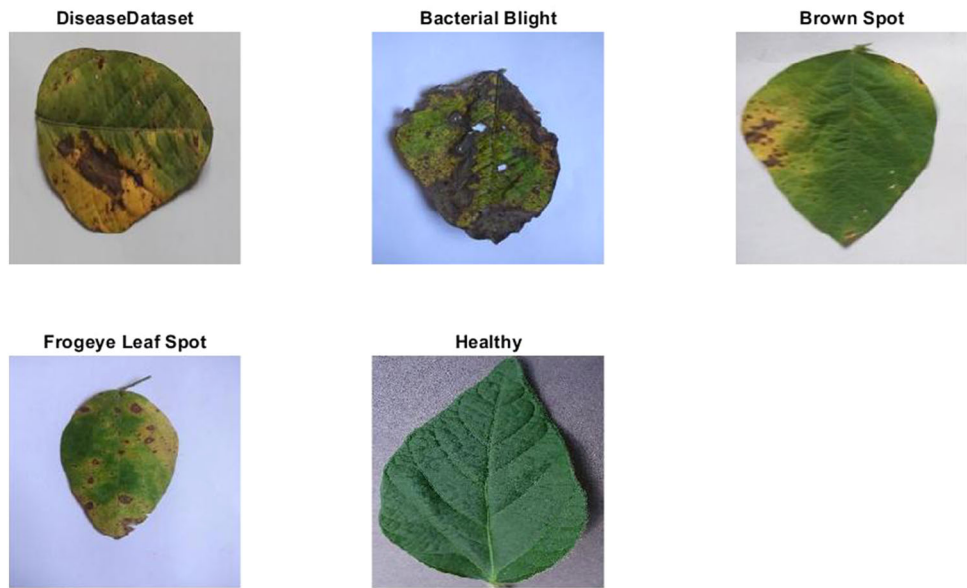


Table 1 Training and test data set of AlexNet CNN

Sr. no.	Disease class	Training samples	Test samples
1	Bacterial blight	199	20
2	Brown spot	150	20
3	Frogeye leaf spot	200	20
4	Healthy	100	20
Total		649	80

Table 2 Training and test data set of GoogleNet CNN

Sr. no.	Disease class	Training samples	Test samples
1	Bacterial blight	150	20
2	Brown spot	100	20
3	Frogeye leaf spot	150	20
4	Healthy	100	20
Total		550	80

2.2.2 Convolutional layers

Convolution layers process the input images through a set of convolutional filters, each of which activates certain features from the images. Generally, the convolutional layer output can be represented by Eq. (1)

$$M_j^p = f \left(\sum_{i \in M_j} M_i^{p-1} * k_{ij}^p + N_j^p \right) \tag{1}$$

where p represents the pth layer, k_{ij} denotes convolutional kernel, N_j denotes bias and M_j denotes a set of input maps. The various parameters of architecture, such as the bias and the weight of the kernel, are typically trained using unsupervised learning approach [13, 18]. The raw input image applied to convolutional layer through a set of filters, each of which activates certain features from the raw input image. In the convolutional layers, a CNN utilizes various kernels to convolve the whole raw input image as well as the intermediate feature maps, generating various feature maps.

2.2.3 Pooling layers

Pooling layers simplify the output by performing nonlinear down sampling, which reduces the number of parameters that the network must learn. In stochastic pooling, the probability p should first be computed for each region j according to Eq. (2)

$$P_i = \frac{\alpha_i}{\sum_{k \in S_j} \alpha_k} \tag{2}$$

where S_j is pooling region j , F is feature map, and i is the every element index inside region j . Stochastic St, is, used in pooling operation for each future map F , the stochastic (St) is expressed by:

$$a_{xy}^{p,k} = St(m, n, x, y) \varepsilon P \left(\alpha_{m,n}^{p-1,F} w(x, y) \right) \tag{3}$$

where α p, k x, y is the neuron activation at coordinate (x, y) in feature map F in p^{th} layer, $w(x, y)$ is the weighing function.

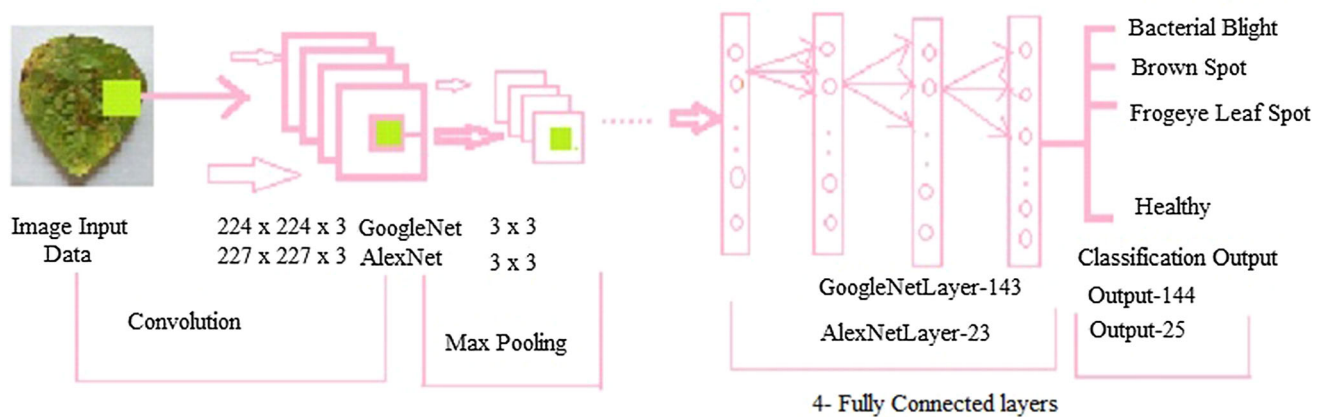


Fig. 2. Proposed AlexNet and GoogleNet CNN general architecture

2.2.4 Fully connected layers

Fully connected layers “flatten” the network’s 2D spatial features into a 1D vector that represents image-level features for classification purposes.

2.3 Image preprocessing and labeling

To improve the recognition accuracy of the proposed models during feature extraction, the final images intended to be used as the training and testing data sets for the proposed deep neural network classifier were preprocessed for consistency. A total of 649 and 550 soybean leaf sample images were preprocessed to input image dimensions of $227 \times 227 \times 3$ for the AlexNet architecture model and $224 \times 224 \times 3$ for the GoogleNet architecture model. Then, the preprocessed sample images from the training data set were used to train the AlexNet and GoogleNet CNN models. The output of random testing samples with data labeling through the AlexNet network is displayed in Fig. 1. To improve the recognition accuracy of the proposed models, the conventional ML model training parameters, such as the max epoch, minibatch size, and learning rate, were modified.

2.4 AlexNet and GoogleNet CNN training

Network training involves two stages: a forward stage and backward stage. First, the main goal of the forward stage is to represent the input image with the current parameters (weights and bias) in each layer. Then, the prediction output is used to compute the loss cost with the ground truth labels. Second, according to the loss cost, the backward stage computes the gradients of each parameter by using chain rules. All the parameters are updated according to the gradients and are prepared for the next forward computation. Network learning can be halted

after sufficient iterations of the forward and backward stages.

In feedforward pass stage, we consider a soybean disease multiclass task with N classes and T training samples. The squared-error function is given by

$$E^T = \frac{1}{2} \sum_{t=1}^T \sum_{k=1}^N (d_k^t - y_k^t)^2 \quad (4)$$

where d_k^t is the k th dimension of the t th pattern’s corresponding label, and y_k^t is the value of the k th output layer unit in response to the t th input pattern. We have used supervised learning techniques to train the proposed CNNs to learn classification of 4 various soybean diseases. Thus, from the image futures CNNs learned to recognize soybean diseases based on maximized activation neurons with stochastic response in next higher layer Regression is applied in soybean multiclass disease classification task. Suppose $H(m)$ and $J(m)$ are defined training dataset, then $\{(H(1), J(1)), \dots, (H(m), J(m))\}$, $J_i \in (1, 2, \dots, k)$. The probability of classifying m as class J is:

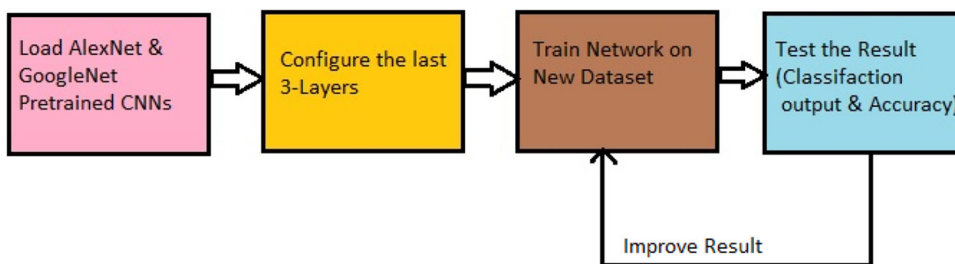
$$P(n^{(i)} = J | m^{(i)}; \theta) = \frac{e^{J^T m^i}}{\sum_{p=1}^k e^{J^T m^i}} \quad (5)$$

2.4.1 Retraining of pre-trained AlexNet and GoogleNet layers

In the pretraining phase, we have used trained deep architectures on a large data set, such as ImageNet, by using powerful machines [18, 19]. The objective of this phase was to initialize network weights for the next phase. Our aim was to use the advantages of these pretrained architectures to enhance the results in the proposed disease classification task.

Figure 3 depicts the process of retraining the AlexNet and GoogleNet models from the raw input image with the predicted output probabilities of each disease. The input

Fig. 3 Retraining process of AlexNet and GoogleNet CNNs model



images of the network were resized to 227×227 pixels for AlexNet and 224×224 pixels for GoogleNet respectively. The output results represent the probabilities of each disease. We proposed retraining the deep CNN for developing an image classification model from the data set described in Table 1.

We retrained the AlexNet and GoogleNet networks to classify four categories of soybean leaf diseases (Fig. 3) [15]. The following steps were involved in retraining the networks:

1. Loading the pretrained network.
2. Reconfiguring the last three layers to perform a new recognition task.
3. Training the model with new data.
4. Testing the performance result.

The architectures were reconfigured, modified, and adjusted to support the four defined classes shown in Tables 3, 4.

3 Experimental results and discussion

3.1 Results

3.1.1 Plot of the training progress

Our aim was to improve the performance accuracy of the model over time. Progress plots were obtained for the

Table 3 Architecture of retrained AlexNet model

Layer	Function	Filter size	Stride
Conv 1	Convolution	$11 \times 11 \times 3$	4
Pool 1	Max pooling	3×3	2
Conv 2	Convolution	$5 \times 5 \times 48$	1
Pool 2	Max pooling	3×3	2
Conv 3	Convolution	$3 \times 3 \times 256$	1
Conv 4	Convolution	$3 \times 3 \times 192$	1
Conv 5	Convolution	$3 \times 3 \times 192$	1
Pool 5	Max pooling	3×3	2

Table 4 Architecture of retrained GoogleNet model

Layer	Function	Filter size	Stride
Conv 1	Convolution	$11 \times 11 \times 3$	4
Pool 1	Max pooling	3×3	2
Conv 2	Convolution	$5 \times 5 \times 48$	1
Pool 2	Max pooling	3×3	2
Conv 3	Convolution	$3 \times 3 \times 256$	1
Conv 4	Convolution	$3 \times 3 \times 192$	1
Conv 5	Convolution	$3 \times 3 \times 192$	1
Pool 5	Max pooling	3×3	2

network training [15]. Figure 4 depict the training progress for the bacterial blight, disease category. Our model seems to have improved after the 50th iteration and then increased up to approximately 98% accuracy. It means the network is able to converge on a solution. We have modified the training options and the network configuration as a result of changing training parameter; we get a much better result more than 95% accuracy.

3.1.2 Inception layer and lgraph

Implementation of GoogleNet CNN for proposed soybean disease task, we have used inception model which was Google recently released a model called Inception v3 with Tensor flow. We have used this model by retraining the last 3 layers per our defined four category classification requirements. Inception modules are basically mini models inside the bigger model. The same Inception architecture was used in the GoogleNet model which was state of the art image recognition net [18, 19]. The inception model itself select type of convolution (1×1 or 3×3 or 5×5), done in parallel and concatenating the resulting feature maps before going to the next layer. Then each of the convolution’s feature maps will be passes through the mixture of convolutions of the current layer. This architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions. Figure 5 shows the graph of inception model of an

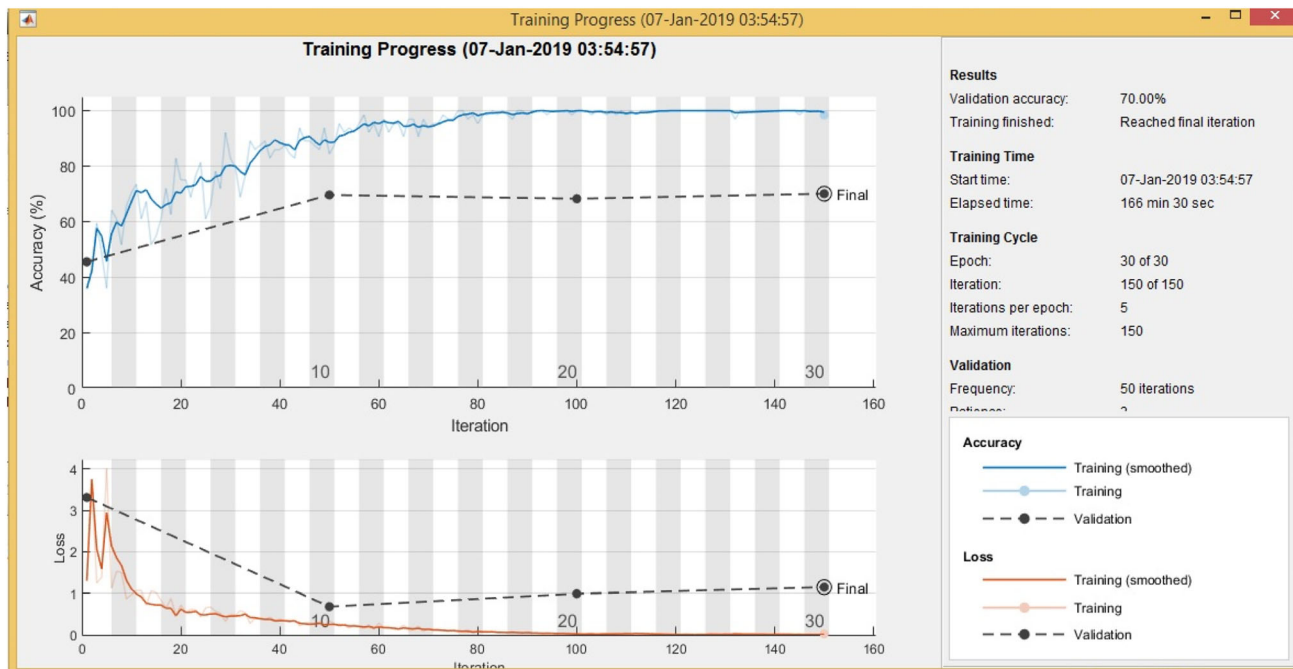
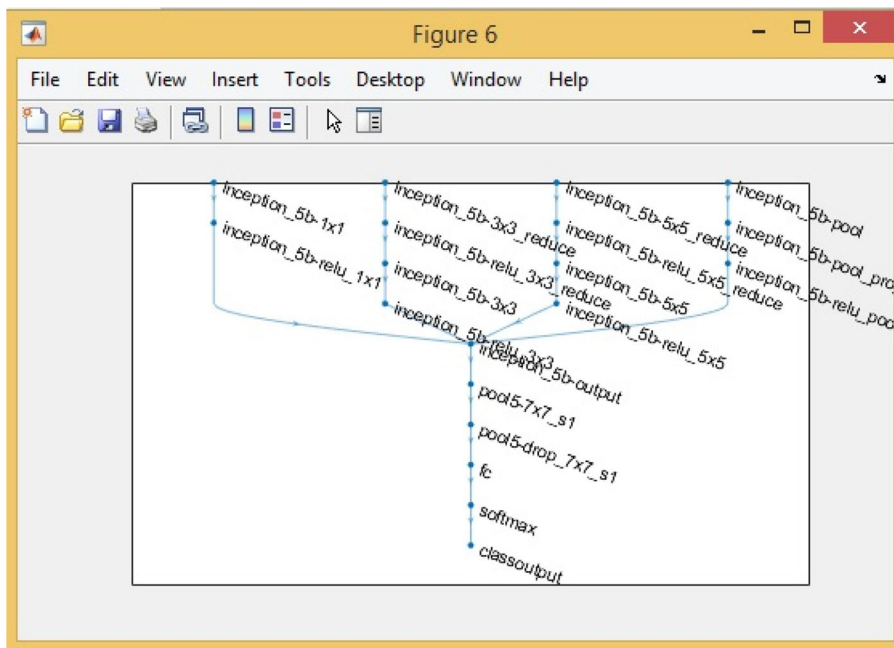


Fig. 4. Plot of training progress for bacterial blight class using GoogleNet model.

Fig. 5 Inception model graph of pre-trained GoogleNet architecture



entire inception module. We have used 1×1 , 3×3 , and 5×5 convolutions along with a 3×3 max pooling.

3.1.3 Soybean disease classification

Total 649 data samples for four data class are considered to train the AlexNet CNNs model and total of 80 data samples are considered to test the performance of system. From which, there are 1 data sample misclassified, 1 data in

class2 misclassified, and for all other class no data were misclassified, shown in Fig. 7 of confusion matrix of AlexNet CNN. So, classification accuracy for disease class 1 is 100%, disease class 2 is 95% disease class 3 is 100% and non-disease (healthy) class4 is 100% respectively for leaf with Bacterial Blight, leaf, leaf with Brown, spot and leaf with healthy summarized in Table 5.

Similarly, the training data set of the GoogleNet CNN model included 550 samples for the four data classes, and

Table 5 Classification result of AlexNet and GoogleNet CNN’s classification

Image dataset	Total testing data samples (actual)	Predicted samples using AlexNet CNN	% accuracy AlexNet CNN	Predicted samples using GoogleNet CNN	% Accuracy GoogleNet CNN
Bacterial blight	20	20	100	19	95
Frogeye leaf spot	20	19	95	19	95
Septoria brown spot	20	20	100	19	95
Healthy	20	20	100	20	100
Combined 4 Data class	80	79	98.75	77	96.25

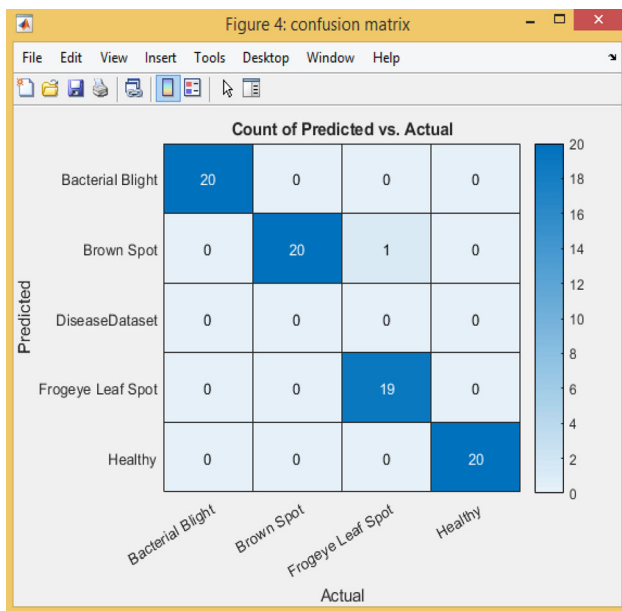


Fig. 6 Confusion matrix of AlexNet CNN (Predicted vs. Actual class)

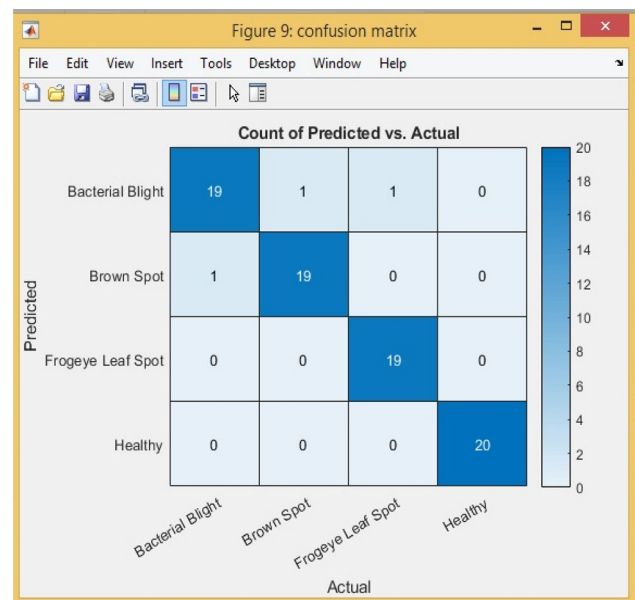


Fig. 7 Confusion matrix of GoogleNet CNN (Predicted vs. Actual class)

80 data samples were considered to test the performance of the system. Of these 80 samples, three were misclassified. One data sample each in class 1, class 2, and class 3 was misclassified, as depicted in confusion matrix of the GoogleNet CNN (Fig. 7). Thus, the classification accuracy for class 1, class 2, class 3, and class 4 were 95%, 95% 95%, and 100%, respectively. The classification results for bacterial blight, FLS, brown spot, and healthy leaves are summarized in Table 5.

3.1.4 Confusion matrix

The confusion matrix of the predicted and actual class categories obtained using the AlexNet CNN (Fig. 6) was used for classifying 80 disease sample test images into each class category. For class 1 (bacterial blight), class 3 (FLS), and class 4 (healthy), the values on the diagonal were 20,

which indicated that each test image in these categories was correctly classified. For class 2 (brown spot), 19 out of 20 samples were correctly classified and one sample was misclassified.

The confusion matrix of predicted and Actual class category classified using GoogleNet, shown in Fig. 7 for predictions of 80 disease sample test images in each class category. For healthy class category the values on the diagonal were 20, this would indicate that each test image in this category was correctly classified. For bacterial blight, brown spot and frogeye spot disease class categories out of 20 samples, 19 samples from each category were correctly classified, and 1 from each class category getting misclassified.

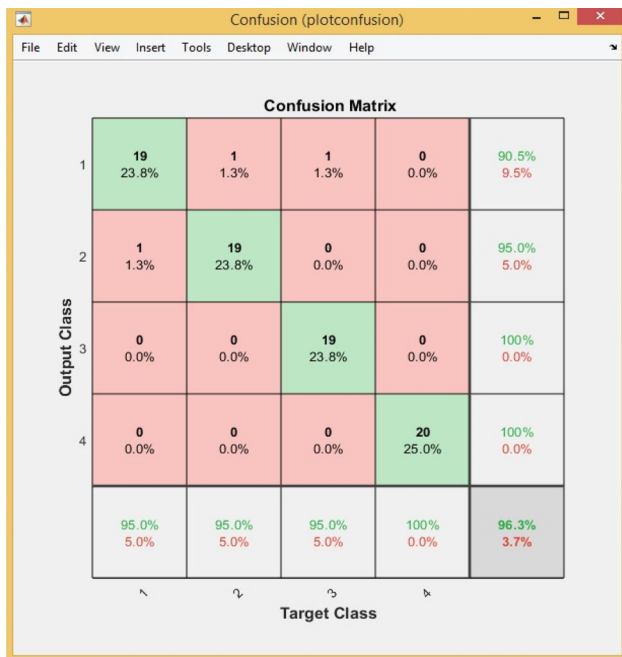


Fig. 8 Sophisticated confusion matrix of output class vs. Target class of GoogleNet CNN

3.1.5 Sophisticated Confusion Matrix

Figure 8 depicts the sophisticated confusion matrix of GoogleNet CNN which shows summary of results of soy-bean leaf disease classification for the 4 classes. The accuracy is improved and reaches to 96.3% which is a good amount as shown in diagonal gray box. It clearly shows that class 1 is one time misclassified as class 2 by 1.3%, class2 is one time misclassified as class 1 by 1.3%, and class 3 is one time misclassified as class 1 by 1.3% but class 4 is correctly classified as class 4 by 100%.

3.1.6 Comparative analysis with ML system

The performance of the proposed CNN model was compared with that of a previous ML system implemented by Al-Bashish and et.al.[1]. The comparison is presented in Table 7, which indicates that the proposed model outperformed the ML system.

Table 7 Comparative study of Proposed CNNs model with ML system

Particulars	Previous implemented system by Al-Bashish and et al. [1]	Proposed AlexNet and GoogleNet CNNs classifier model
disease detected	Early scorch, cottony mold and late scorch etc	Bacterial blight, brown spot and frogeye leaf spot
No. of samples per class	25	20
Feature extracted	Color and texture-manually extracted	Automatically extracted by model
Classifier	Neural network with back propagation	AlexNet and GoogleNet CNN Classifier
Accuracy	93%	98.75 and 96.25

Table 8 Training performance of AlexNet CNN

Epoch	Iteration	Time elapsed (hh: mm: ss)	Mini-batch accuracy	Mini-batch loss	Base learning rate
1	1	00:00:20	26.56%	1.9013	0.0010
7	50	00:15:28	82.81%	0.4054	0.0010
13	100	00:30:06	81.25%	0.4138	0.0010
19	150	00:44:45	98.44%	0.0355	0.0010
20	160	00:47:36	98.44%	0.0370	0.0010

Table 9 Training performance of GoogleNet CNN

Epoch	Iteration	Time Elapsed (hh: mm: ss)	Mini-batch accuracy	Validation accuracy	Mini-batch loss	Validation loss	Base learning rate
1	1	00:00:20	42.19%	46.36%	1.2863	1.9734	0.0010
10	50	00:15:28	89.06%	70.45%	0.3471	0.6279	0.0010
20	100	00:30:06	96.88%	72.73%	0.0878	0.7278	0.0010
30	150	00:44:45	100.00%	72.73%	0.0257	0.8870	0.0010

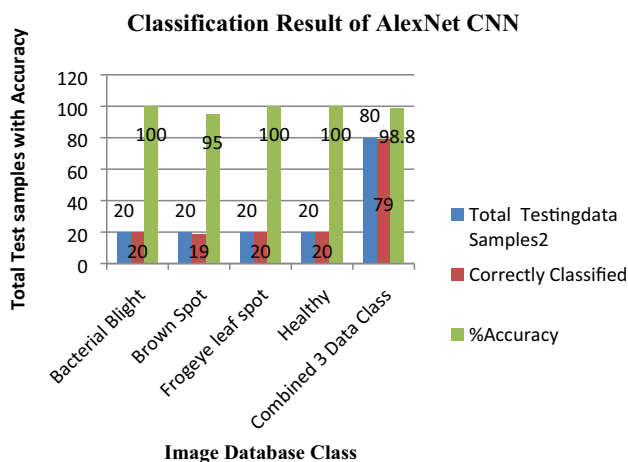


Fig 9 Classification Result of AlexNet CNN

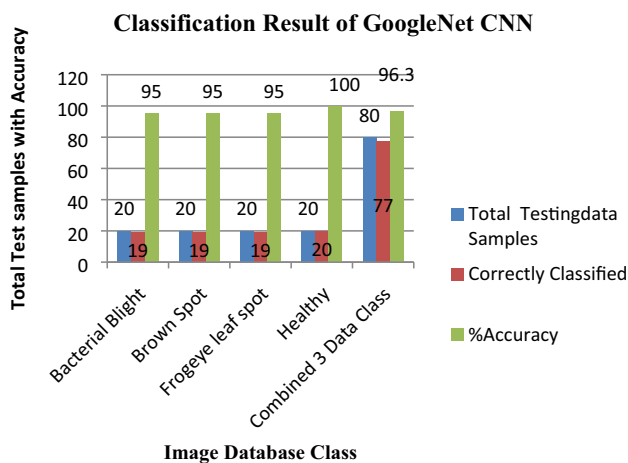


Fig. 10 Classification Result of GoogleNet CNN

3.1.7 Training performance and accuracy result

Tables 8 and 9 present the training performance of the AlexNet and GoogleNet CNNs with hyper parameter details. The tables indicate the elapsed time of training and the overall classification accuracy after testing new data.

In this study, 80 samples were considered for the AlexNet and GoogleNet CNNs. A total of 20 samples were tested in each disease class category. Figures 9, 10 depict the overall classification accuracy of the defined disease class categories when using the proposed CNN model.

4 Conclusion

In this study, we proposed a deep learning approach that involved using the AlexNet and GoogleNet CNN architectures to build a classifier model for the defined one non-disease and three disease classes (bacterial blight, brown

spot, and FLS). The classification accuracies for the AlexNet and GoogleNet CNN models were 98.75% and 96.25%, respectively. Classification was performed with the AlexNet and GoogleNet models by modifying various hyper parameters, such as the minibatch size, max epoch, and bias learning rate. Our experimental results indicate that the proposed deep convolutional neural network model outperformed the machine learning model in soybean disease classification. Future studies can attempt to increase the performance rate of the model by varying the minibatch size, bias learning rate, and weight.

Acknowledgements This work is supported by Dr. V.S.Patil, Plant Pathology, Zonal Agricultural Research Station, Kolhapur under Mahatma Phule Krishi Vidyapeeth, Rahuri, Maharashtra India.

References

- Al-Bashish D, Braik M, Bani-Ahmad S (2011) Detection and classification of leaf diseases using K-means-based segmentation and neural networks based classification. *Inform Technol J* 10(267–275):2011. <https://doi.org/10.3923/itj.2011.267.275>
- Barbedo JGA (2016) A review on the main challenges in automatic plant disease identification based on visible range images. *Biosys Eng* 144:52–60
- Barbedo JGA (2013) Digital image processing techniques for detecting, quantifying and classifying plant diseases. *SpringerPlus* 2(1):660–672
- S. Phadikar, J. Sil, Rice disease identification using pattern recognition techniques. In: *Proceedings of the IEEE International Conference on Computer and Information Technology (ICCIT)*, Khulna, Bangladesh, 2008, pp 420–423
- Asfarian A, Herdiyeni Y, Rauf A, Mutaqin KH (2014) A computer vision for rice disease identification to support integrated pest management. *Crop Prot* 61:103–104
- Khairnar K, Dagade R (2014) Disease detection and diagnosis on plant using image processing—a review. *Int J Comput Appl* 108(13):36–39
- Kim DG, Burks TF, Qin J, Bulanon DM (2009) Classification of grapefruit peel diseases using color texture feature analysis. *Int J Agric Biol Eng* 2(3):41–50
- Tejal Deshpande and K. S. Raghuvanshi (2014) Grading and identification of disease in pomegranate leaf and fruit. *IJCSIT* *Int J Comput Sci Inf Technol* 5(3): 4638–4645
- Chai Y, Wang XD (2013) Recognition of greenhouse tomato disease based on image processing technology. *Tech Autom Appl* 9:83–89
- Zhang LN, Yang B (2014) Research on recognition of maize disease based on mobile internet and support vector machine technique. *Trans Tech Publ* 108(13):659–662
- Al Hiary H, Bani Ahmad S, Reyalat M, Braik M, ALRahamneh Z (2011) Fast and accurate detection and classification of plant diseases. *Int J Comput Appl* 17:31–38. <https://doi.org/10.5120/ijca>
- Mokhtar U, Hassenian AE, Emary E, Mahmoud MA (2015) SVM-Based detection of tomato leaves diseases. In: *Advances in intelligent system and computing*. Springer, pp 641–652
- Le Cun Y, Bottou L, Bengio Y, et al. (1998) Gradient-based learning applied to document recognition. *Proc. IEEE* 86(11): 2278–2324

14. Yanming G, et. al. (2016) Deep learning for visual understanding: a review. *Neurocomputing* 187: 27–48
15. Practical Deep Learning Examples with MATLAB (2018) Math Works, Inc., pp 1–33
16. Yang L et al (2018) Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267:378–384
17. Brahimi M, Boukhalifa K, Moussaoui A (2017) Deep learning for tomato diseases: classification and symptoms visualization. *Appl Artif Intell* 31:299–315
18. Hinton GE (2012) A practical guide to training restricted Boltzmann machines. *Mo-Mentum* 9(1):599–619
19. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, pp 1–783
20. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci.* <https://doi.org/10.3389/fpls.2016.01419>
21. Konstantinos PF (2018) Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric* 145:311–318
22. Rangarajan AK, Purushothaman R, Ramesh A (2018) Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Comput Sci* 133:1040–1047