2021 8th International Conference on Signal Processing and Integrated Networks (SPIN) | 978-1-6654-3564-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/SPIN52536.2021.9565945

Plant Disease detection of Bell Pepper Plant Using Transfer Learning over different Models

Prateek Thakur
University School of Information, Communication and Technology
Guru Gobind Singh Indraprastha University
New Delhi-110078, India
Email: prateek6397@gmail.com

Anuradha Chug
University School of Information, Communication and Technology
Guru Gobind Singh Indraprastha University
New Delhi-110078, India
Email: anuradha@ipu.ac.in

Amit Prakash Singh
University School of Information, Communication and Technology
Guru Gobind Singh Indraprastha University
New Delhi-110078, India
Email: amit@ipu.ac.in

***Abstract* — Bell Pepper, also called Capsicum, is a widely grown crop that suffers primarily from bacterial spot disease causing severe damage to the plant by dropping its leaves and fruits. The current methods to detect the disease are either through lab diagnosis, which takes quite an amount of time and may lead to further spreading of the disease throughout the crop, or an experienced eye. Focusing on this issue, this paper implements 3 Convolution Neural Network (CNN) architectures, namely, InceptionV3, ResNet50 and VGG16 via transfer learning technique to detect bacterial leaf spot disease in Bell Pepper Plant with the help of images from Plant Village dataset. These models were trained and tested over 2442 images of diseased and healthy leaves. The dataset images have also been augmented in various degrees to get better results. Further, fine tuning of the models by varying learning rates, batch size, epochs etc. has resulted in very high accuracies. VGG16 gives the best results with 99.72% accuracy and 0.998 AUC. ResNet50 showed 99.31% accuracy and 0.994 AUC whereas InceptionV3 resulted in the least of three with 95.77% accuracy and 0.953 AUC. These models are chosen due to their variable architectural differences and their performances are later compared using Accuracy, Loss, F1-score and AUC-ROC curve as performance metrics. The results in this paper show considerable improvement as compared to other research papers on detection of diseases in Bell Pepper plant. They can be helpful in the agriculture sector in timely detection of bacterial spot disease.**

Keywords— Deep Learning, Transfer Learning, ResNet50, VGG16, InceptionV3, Convolutional Neural Networks. Bacterial Leaf Spot, Bell Pepper Plant

I. INTRODUCTION

A total of 327 thousand tons of Capsicum is produced in India every year, taking up 46 thousand hectares of land with a productivity rate of 7180.70kg/hectare. It is extensively produced in Tamil Nadu, Karnataka, Himachal Pradesh, Andhra Pradesh, Maharashtra and hilly regions of Uttar Pradesh. According to S. Senthilkumar et al. an investment of 55.5 thousand USD/4000 m^2 is required to set up a polyhouse which majorly consists of problems like high maintenance cost, non-availability of skilled labor, lack of technical guidance, pest and disease problems etc.[1]

Bacterial Leaf Spot is a disease caused by *Xanthomonas campestris pv. Vesicatoria* [2] which causes severe damage to the plant by dropping the leaves and fruit from the bell pepper plant. It is a rod-shaped gram-negative bacterium that can even survive in plant debris and seeds from season to season. Bacterial Leaf spot develops rapidly in warm and moist conditions. The early symptoms are watery lesions that changes into dark brown spots that are one fourth of an inch in diameter. With time the damaged tissues dry out and fall off, leaving a tethered appearance of the affected leaf as shown in Fig. 1.

Every year a huge amount of time and money is invested in disease control without guaranteeing to save the crop. In addition to this overuse of crop protection chemicals reduce the quality of the soil and have an adverse effect on the natural environment. This can be avoided if we detect the pathogen, here the bacteria, at its early stage which will enable us to use the required amount of chemical at the affected part of the crop.

To detect diseases in plants, several Deep Learning [3] techniques such as Convolutional Neural Networks (CNN) [4], Artificial Neural networks (ANN) and Recurrent Neural Networks (RNN) have been implemented on various crops. In this paper the authors have used three pre-trained state-of-the-art CNN architectures, namely, ResNet50, VGG16 and InceptionV3 and trained them to detect Bacterial Leaf Spot disease on Bell Pepper plant leaf images. The models were trained over 2442 images of both diseased and healthy Bell Pepper plant leaves. The performances of all three models were compared via performance metrics such as validation accuracy, validation loss, F1-score and AUC – ROC curve.

The current study focuses on training the CNN models using Transfer Learning technique and fine tuning it to get higher accuracy with minimum loss. Later, a comparative study is done among the three models via performance metrics such as accuracy, loss, F1-score and AUC-ROC curve. The next section emphasizes on the related researches previously done followed by section III presenting the material and methods used in this study. Section IV is about the methodology involved to implement the network models. Further, section V outlines the results followed by a discussion about them. Lastly, section VI concludes the study and shows the direction for future work.

Fig. 1. Bell Pepper leaves infected by Bacterial Leaf Spot [5]

II. Related work

Wang et al. used VGG16, VGG19, InceptionV3 and ResNet-50 on Apple black rot disease and was able to achieve 80.0 – 90.4% classification accuracy[6]. Mohanty et al. trained GoogleNet and AlexNet using both with and without Transfer learning over 14 crops in Plant Village Dataset. By varying training and test ratios and using colored, grayscale and segmented versions of the images he achieved an accuracy of 99.35 % [7]. Zangh et al. achieved 97.28% as highest accuracy after training AlexNet, GoogleNet and ResNet with various hyperparameters in order to detect tomato leaf disease [8]. Bhagat et al. were able to achieve 96.78% test accuracy on their CNN model for Bell Pepper leaf disease classification [9]. Verma et al. implemented capsule network to detect potato leaf disease and achieved an accuracy of 91.83% [10]. Bhatia et al. proposed a hybrid model of Support Vector Machine (SVM) and Logistic Regression (LR) to predict powdery mildew disease in tomato plant and achieved 92.37% accuracy [11].

A number of research models have been presented by various researchers, but Bell Pepper leaf disease detection is done by very few. Our models output highly accurate results with very low loss function value and we have compared them via various performance metrics presenting a clearer understanding of the models.

III. Materials and methods

This section discusses the preliminaries of this study and has been divided into different subsections. First, subsection discusses about the dataset used to train the models. The next three subsections present an overview of the 3 CNN models used in this i.e., ResNet50, VGG16 and InceptionV3. Lastly, it discusses about the performance metrics used to compare the performances of the three models.

A. PlantVillage dataset

The PlantVillage dataset is a wide storage of 50,000 leaf images of fourteen crops categorized into thirty-eight plant-disease pairs available in grayscale, color, and segmented images. In this paper we have considered 2442 images of Bell Pepper plant leaves, out of which 1462 are of healthy leaves and 980 are of diseased leaves. Both the datasets are further divided into a 70:30 train-test ratio as shown in Table 1.

TABLE I. Distribution Of Bell Pepper Leaf Dataset

Bell Pepper Plant Dataset		
Class	***Train (70%)***	***Test (30%)***
Healthy	1023	439
Diseased	686	294

B. VGG16

K. Simonyan and A. Zisserman proposed a convolutional neural network in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition"[12]. This model has been trained on to 14 million images over 1000 classes and achieved 92.7% test accuracy in ImageNet [13]. It takes 224 x 224 RGB Images as input which are then passed through a stack of convolutional layers. The filters used in the convolutional layer have 3x3 kernel size and the convolutional stride is set to 1 pixel. Various configurations follow the same design, architecture and differ only in depth. The number of channels in the convolutional layers starts from 64 and are increased by a factor of 2 till it reaches 512. Five MaxPool layers carry out spatial pooling. All hidden layers are equipped with Rectified Linear Unit (ReLU) non- linearity. The network has 3 fully connected layers at the end of convolutional layers with first 2 layers having 4096 channels and last layers has 1000 channels (for classification of 1000 classes). With the help of transfer learning, we can modify the layers up to our own requirements and use the pre-trained weights of VGG 16 to get better results.

C. ResNet50

Residual Network or ResNet [14] is a neural network, which can be 8 times deeper than VGG nets with 152 layers and still has much lower complexity. Since deep neural networks are difficult to train due to vanishing gradient problem, the authors of ResNets came with the concept of *skip connections* or shortcuts to jump off some layers. Skip connections use fewer layers in the initial learning stages. This reduces the effect of vanishing gradient. It then gradually restores the skipped layers as it learns the feature space. This way it remains closer to the manifold[15]. Otherwise, the network would learn unnecessary details of the feature space, causing it to overfit on the data. Using Transfer Learning we have used the pre-trained weights of ResNet50 and modified it to meet our needs.

D. InceptionV3

Inception Network is fitting for images where the object to be detected have large variations in sizes e.g. A leaf image captured can be taken from far away or it can be taken in close up shot. Other CNNs such as VGG and ResNets use a single kernel size in the convolution step. Due to large variations in the location of information, finding the right kernel size is difficult. Hence, Inception incorporates convolution with different kernel sizes in the same layer. This makes the network wider instead of deeper. InceptionV2 and InceptionV3 are an enhancement to Inception Network. InceptionV2 reduces representation bottlenecks [16] and use smart factorization methods[17] during convolution to improve the computational efficiency. InceptionV3 includes all the upgrades in InceptionV2 and in addition uses RMSProp Optimizer, factorised 7x7 convolutions, BatchNormalization [18] in Auxillary Classifiers and Label Smoothing.

E. Performance Metrics

Evaluation and comparison cannot be done without quantification. Evaluation of a deep learning algorithm is an essential part of the project. We usually judge the performance of our model with classification accuracy, but it might give poor results in error loss or any other such metrics. Hence, different metrics should be considered while evaluating the performance of a deep learning model.

Below are few performance metrics that are used to evaluate the models in this paper: -

Confusion Matrix: Confusion Matrix is, as the name suggests, a matrix that evaluates the overall performance of the deep learning model. The matrix values show a comparison between predicted data and actual data. Confusion matrix is also the basis of all other metrics in performance evaluation. The confusion matrix constitutes of 4 important terms discussed below: -

- True Positives (TP): TPs are the number of cases that were predicted to be classified into, say, class 'Healthy' and the actual classification was also a class 'Healthy'.
- True Negatives (TN): TNs are the number of cases that were predicted to be classified in, say, class 'Diseased' and the actual classification was also a class 'Diseased'.
- False Positives (FP): FPs are the number of cases that were predicted to be classified as 'Healthy' but actually they were in class 'Diseased'.
- False Negatives (FN): FNs are the number of cases that were predicted to be classified as 'Diseased' but actually they were in class 'Healthy'

These 4 terms together create a confusion matrix. For a binary classification it will a 2 x 2 matrix. Table II shows a sample confusion matrix w.r.t to the dataset used in this paper.

TABLE II. CONFUSION MATRIX

Actual	**Predicted**	
	Diseased	***Healthy***
Diseased	TN	FP
Healthy	FN	TP

Accuracy: It is a measure of how close our model is to the actual data. It is calculated by taking the average of the values in the principal diagonal of the confusion matrix i.e.

$$Accuracy = \frac{True\ Postive + True\ Negatives}{Total\ Sample} \quad (3)$$

F1-score: F1-score measures a test's accuracy. It is the harmonic mean between precision and recall. It shows how many cases are classified correctly and how robust is the classifier. F1- score considers both predicted positive results and all predictions that should have been positive.

$$F1 - Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (4)$$

Precision and Recall can be defined as: -

Precision: It is the ratio of the number of correct positive predicted cases and total number of cases that are predicted positive. It can be written as: -

$$Precision = \frac{True\ Positives}{True\ Positives + FalsePositives} \quad (5)$$

Recall: It is the ratio of number of correct positive predicted cases and total number of actual positive cases. It can be written as: -

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (6)$$

AUC– ROC (Area under Curve - Receiver Operating Characteristics) curve: It is the most widely used metrics to evaluate performance of a classification problem. ROC curve is a graph between True Positive Rate (sensitivity) and False Positive Rate (1-specificity). ROC is a probability curve and AUC represent the degree of separability. Therefore, this metric tells us how much model is capable in classifying the data. The value of AUC ranges from 0 to 1, 0 being the lowest and 1 being the highest. The higher the AUC value the better is the capability of the model in distinguishing between the diseased leaves and healthy leaves. Fig 2. is a sample graph showing the AUC – ROC curve.

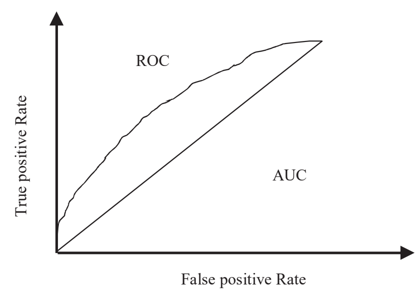

Fig. 2. AUC-ROC Curve

IV. PROPOSED METHOD

Using transfer learning technique, the authors of this paper have trained three state-of-the-art CNNs, namely, VGG16, ResNet50 and InceptionV3 over a Dataset of 2442 leaf images of Bell Pepper plant both diseased and healthy. A common research methodology has been adopted for all the three networks shown in Fig. 3. Following are the descriptions of the methods proposed for each network: -

A. VGG16

The authors have preprocessed both train and test images by resizing the image to 224 x 224 and then changing the color space of the images from RGB (red green blue) model to BGR (blue green red) model as suggested in VGG16 input specifications. Pre-trained weights obtained from ImageNet have been used during transfer learning. At the end of the convolution layers two 256 channel dense layers with activation function as "rectified linear unit" (Relu) is added. Lastly, an output dense layer of 2 units is added with "softmax" activation function. Stochastic Gradient Decent (SGD) optimizers have been used with learning rate = 0.001 and momentum = 0.9. Binary Crossentropy is the loss function used to train the model.

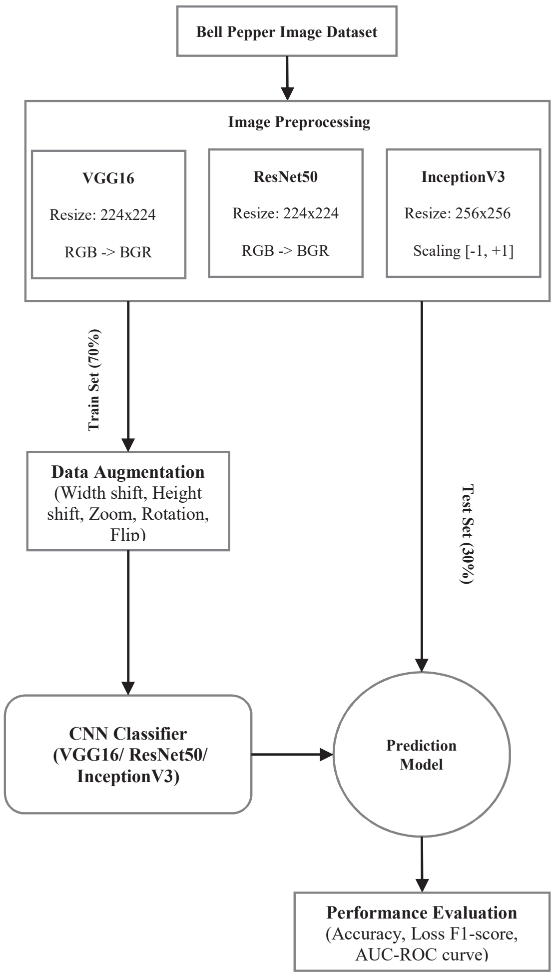

Fig. 3. Research Methodology of the proposed work

B. ResNet50

The images are preprocessed and resized to 224 x 224 pixels to train ResNet50 model. The training data are augmented using various methods such as rotation, width shift, height shift, magnification and flip in order to obtain higher accuracy. The model is trained in batches of 32 over 20 epochs. At the end of the ResNet, two dense layers with 256 and 128 channels respectively, are used by the activation function as rectified Linear Unit (ReLu). Glorot Kernel Initializer [19] has been used in the Dense Layers. Lastly, Adam optimizer has been used with learning rate of 0.0001.

C. InceptionV3

The images are preprocessed and resized to 256 x 256 pixels to train InceptionV3. The training data are augmented using various augmentation methods like rotation, width shift, height shift, magnification etc. to obtain better accuracy. At the end of the convolutions, a global average pooling layer followed by a 1024 channel dense layer with a 20 % drop out is added. Lastly, Adam optimizer is used with learning rate is set to 0.0001.

Table III shows the architectural comparison among the three CNNs. It can be observed that although VGG16 has smaller number of layers, but the size is very high as compared to the other models.

TABLE III. CNN ARCHITECTURE IMPLEMENTED

Model	Size (in MB)	Parameters	Depths
VGG16	528	21,203,778	23
ResNet50	98	48,311,105	179
InceptionV3	92	23,901,985	48

V. RESULTS AND DISCUSSIONS

This section shows the results obtained by training the network models and compares them in both tabular and graphical manner.

A. VGG16

This model showed the highest validation accuracy of 99.85% with a validation loss of 0.0133. The confusion matrix of the prediction is shown in Table IV. Metric calculation is as follows: -

Accuracy = (294 + 437) / (294 + 2 + 0 + 437) * 100
= **99.72%**

Precision = 437 / (437 + 0) = 1

Recall = 437 / (437 + 2) = 0.9954

F1 scores = 2 / ((1/1) + (1/0.9954)) = **0.9977**

TABLE IV. CONFUSION MATRIX FOR VGG16 CLASSIFICATION

Actual	Predicted	
	Diseased	***Healthy***
Diseased	294	0
Healthy	2	437

Fig 4. shows the AUC-ROC curve of VGG16 model with AUC value as **0.998.**

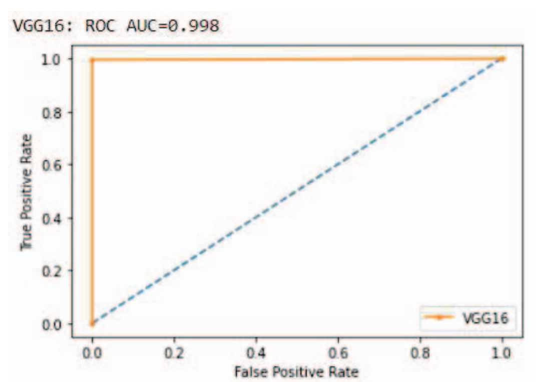

Fig. 4. AUC – ROC curve of VGG16 model

B. ResNet50

ResNet50 training resulted in 98.59% accuracy with validation loss of 0.0408. The confusion matrix of model prediction is shown in Table V. Metric calculation is as follows: -

Accuracy = (294 + 434) / (294 + 0 + 5 + 434) * 100
= **99.31 %**
Precision = 434 / (434 + 0) = 1
Recall = 434 / (434 + 5) = 0.9886
F1 scores = 2 * (1*0.9813) / (1 + 0.9813) = **0.9942**

TABLE V. CONFUSION MATRIX FOR RESNET50 CLASSIFICATION

Actual	**Predicted**	
	Diseased	***Healthy***
Diseased	294	0
Healthy	5	434

Fig 5. Shows the AUC-ROC curve of ResNet50 model with AUC value as **0.994.**

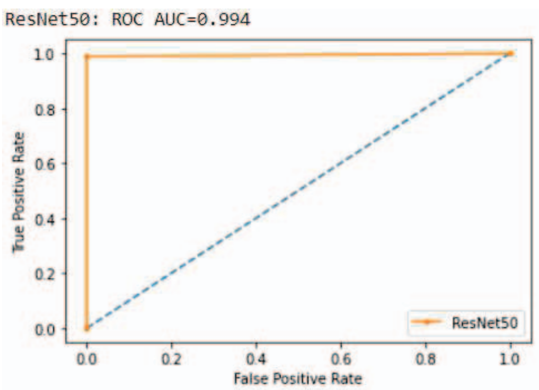

Fig. 5. AUC – ROC curve of ResNet50 model

C. *InceptionV3*

InceptionV3 presented the lowest accuracy among all three networks. The confusion matrix of InceptionV3 model over Bell Pepper plant dataset is given in Table VI. Metric calculation is as follows: -
Accuracy = (264 + 416) / (264 + 416 + 20 + 10) * 100
= **95.77 %**
Precision = 416 / (416 + 20) = 0.9541
Recall = 416 / (413 + 10) = 0.9834
F1 scores = 2* (0.9541 * 0.9834) / (0.9541 + 0.9834)
= **0.9685**

TABLE VI. CONFUSION MATRIX FOR INCEPTIONV3 CLASSIFICATION

Actual	**Predicted**	
	Diseased	***Healthy***
Diseased	264	20
Healthy	10	416

Fig 5. Shows the AUC-ROC curve of ResNet50 model with AUC value as **0.953.**

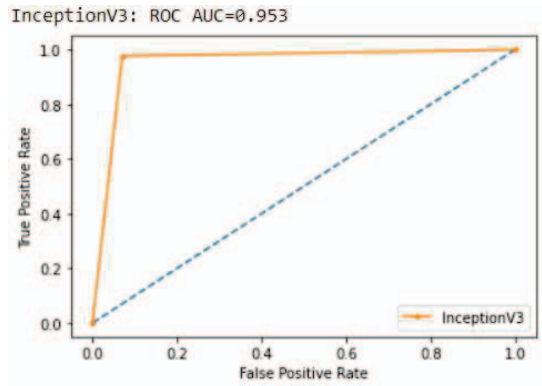

Fig. 6. AUC – ROC curve of InceptionV3 model

Table VII shows the performance comparison between the three CNNs whereas Fig. 7 shows the graphical performance comparison of the network models.

TABLE VII. PERFORMANCE COMPARISON OF RESNET50, VGG16 AND INCEPTIONV3 CLASSIFIERS

Classifier	**Performance Metrics**			
	Accuracy	***Loss***	***F1-score***	***AUC***
Vgg16	**99.72%**	**0.0116**	**0.9977**	**0.998**
ResNet50	99.31%	0.0165	0.9942	0.994
InceptionV3	95.77%	0.1240	0.9685	0.953

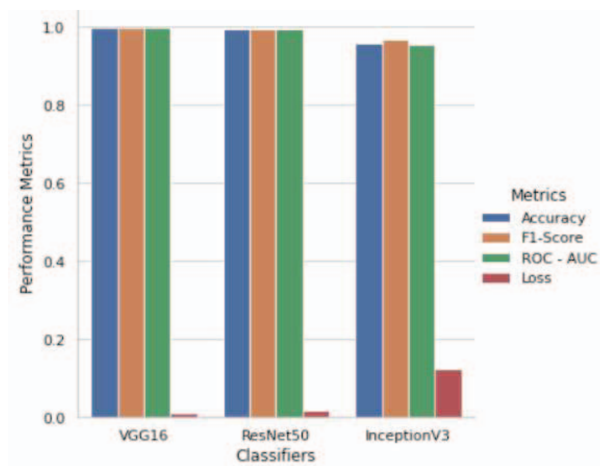

Fig. 7. Performance graph for VGG16, ResNet50 and InceptionV3 classifier

This study is helpful as even though there is so much investment involved in the cultivation of bell pepper plants, there are very few researches that focus on minimizing the efforts required in pest and disease control domain. The models in this paper have shown considerable improvement in detecting the bacterial spot disease as compared to the present research work. Fine tuning of hyperparameters using random search and training dataset augmentation lead to results with very high accuracy and low error loss. Of the three VGG16 model showed the highest precision, however, its architecture occupied the largest space between the three and the design cost was also higher. ResNet50, though being the deepest neural network, showed considerable performance, but took less space and less computing time due to skip connections. InceptionV3 showed the lowest accuracy, but occupied the least space and was originally made to give low computation cost as compared to ResNet50. Accuracy is an important metric, but so is the space occupied and computational cost.

VI. Conclusion and Future Scope

In this paper the authors implemented three pre-trained state-of-the-art CNN architectures, VGG16, ResNet50 and InceptionV3 for classification of healthy and diseased leaves of Bacterial Leaf Spot disease in Bell Pepper Plant, via transfer learning technique. Although VGG16 showed the highest accuracy but due to ResNet50 considerable high accuracy, less size and low computation cost, it turns out to be a better candidate to be practically used in real world conditions.

However, the dataset used in this paper included images of individual distinctive leaves, which is not the case in real time. So, for further research, we plan to incorporate real - time plant leaf images with background noise and overlapping leaves to train our developed models which will give us a clear understanding of the applicability of the modes. Real-time better classification can tell us where to spread the pesticide in the crops, preventing overuse of pesticides over the crops that don't require it. With the help of NIR imaging, we can detect the stress in the plant even before the visible symptoms of the disease appear in the plant. Hence, a large dataset with varying real time data is required to create a model that can be applied practically.

Acknowledgment

We would like to thank the Department of Science and Technology (DST) for their constant support in the completion of this research under the project titled "Application of Internet of Things (IoT) in Agriculture Sector", DST/Reference.No. T-319/2018-19.

References

[1] S. Senthilkumar, K. R. Ashok, M. Chinnadurai, and S. P. Ramanathan, "An Economic Analysis of Capsicum Production under Protected Cultivation in North West Region of Tamil Nadu, India," *Int.J.Curr.Microbiol.App.Sci*, vol. 7, no. 6, pp. 2276–2283, 2018, doi: 10.20546/ijcmas.2018.706.272.

[2] F. Thieme *et al.*, "Insights into genome plasticity and pathogenicity of the plant pathogenic bacterium Xanthomonas campestris pv. vesicatoria revealed by the complete genome sequence," *J. Bacteriol.*, vol. 187, no. 21, pp. 7254–7266, Nov. 2005, doi: 10.1128/JB.187.21.7254-7266.2005.

[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015, doi: 10.1038/nature14539.

[4] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[5] "Bacterial leaf spot in pepper | Vegetable Pathology – Long Island Horticultural Research & Extension Center." https://blogs.cornell.edu/livegpath/gallery/peppers/bacterial-leaf-spot-in-pepper/ (accessed May 21, 2021).

[6] G. Wang, Y. Sun, and J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning," *Comput. Intell. Neurosci.*, vol. 2017, 2017, doi: 10.1155/2017/2917536.

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, no. September, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[8] K. Zhang, Q. Wu, A. Liu, and X. Meng, "Can deep learning identify tomato leaf disease?," *Adv. Multimed.*, vol. 2018, 2018, doi: 10.1155/2018/6710865.

[9] M. Bhagat, D. Kumar, R. Mahmood, B. Pati, and M. Kumar, "Bell pepper leaf disease classification using CNN," Feb. 2020, doi: 10.1109/IDEA49133.2020.9170728.

[10] S. Verma, A. Chug, and A. P. Singh, "Exploring capsule networks for disease classification in plants," *J. Stat. Manag. Syst.*, vol. 23, no. 2, pp. 307–315, Feb. 2020, doi: 10.1080/09720510.2020.1724628.

[11] A. Bhatia, A. Chug, and A. P. Singh, "Hybrid SVM-LR classifier for powdery mildew disease prediction in tomato plant," in *2020 7th International Conference on Signal Processing and Integrated Networks, SPIN 2020*, Feb. 2020, pp. 218–223, doi: 10.1109/SPIN48934.2020.9071202.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2015, Accessed: May 29, 2021. [Online]. Available: http://www.robots.ox.ac.uk/.

[13] "ImageNet." https://www.image-net.org/update-mar-11-2021.php (accessed May 29, 2021).

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[15] P. P. Brahma, D. Wu, and Y. She, "Why Deep Learning Works: A Manifold Disentanglement Perspective," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, no. 10, pp. 1997–2008, Oct. 2016, doi: 10.1109/TNNLS.2015.2496947.

[16] N. Tishby and N. Zaslavsky, "Deep Learning and the Information Bottleneck Principle," *2015 IEEE Inf. Theory Work. ITW 2015*, Mar. 2015, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1503.02406.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2016, vol. 2016-December, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning, ICML 2015*, Feb. 2015, vol. 1, pp. 448–456, Accessed: May 29, 2021. [Online]. Available: https://arxiv.org/abs/1502.03167v3.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." Accessed: May 26, 2021. [Online]. Available: http://www.iro.umontreal.