Generic Paddy Plant Disease Detector (GP2D2): An Application of the Deep-CNN Model

Original Scientific Paper

Samuda Prathima
Department of ECE,
R.M.K. College of Engineering and Technology,
Chennai, India.
samudaprathimaece@rmkcet.ac.in

Praveena N G
Department of ECE,
R.M.K. College of Engineering and Technology,
Chennai, India.
drpraveenang@gmail.com

Sivachandar K
Department of ECE,
R.M.K. College of Engineering and Technology,
Chennai, India.
siva5446@gmail.com

Srigitha S Nath
Department of ECE,
Saveetha Engineering College, Chennai, India.
srigithapramodh@gmail.com

Sarala B
Department of ECE,
R.M.K Engineering College, Chennai, India.
bsa.ece@rmkec.ac.in

Abstract *– Rice is the primary food for almost half of the world's population, especially for the people of Asian countries. There is a demand to improve the quality and increase the quantity of rice production to meet the food requirements of the increasing population. Bulk cultivation and quality production of crops need appropriate technology assistance over manual traditional methods. In this work, six popular Deep-CNN architectures, namely AlexNet, VGG-19, VGG-16, InceptionV3, MobileNet, and ResNet-50, are exploited to identify the diseases in paddy plants since they outperform most of the image classification applications. These CNN models are trained and tested with Plant Village dataset for classifying the paddy plant images into one of the four classes namely, Healthy, Brown Spot, Hispa, or Leaf Blast, based on the disease condition. The performance of the chosen architectures is compared with different hyper parameter settings. AlexNet outperformed other convolutional neural networks (CNNs) in this multiclass classification task, achieving an accuracy of 89.4% at the expense of a substantial number of network parameters, indicating the large model size of AlexNet. For developing mobile applications, the ResNet-50 architecture was adopted over other CNNs, since it has a comparatively smaller number of network parameters and a comparable accuracy of 86.1%. A fine-tuned ResNet-50 architecture supported mobile app, "Generic Paddy Plant Disease Detector (GP2D2)" has been developed for the identification of most commonly occurring diseases in paddy plants. This tool will be more helpful for the new generation of farmers in bulk cultivation and increasing the productivity of paddy. This work will give insight into the performance of CNN architectures in rice plant disease detection task and can be extended to other plants too.*

Keywords*: disease identification, Convolutional Neural Networks, mobile application, paddy disease*

1. INTRODUCTION

Paddy is cultivated globally in more than hundred countries of a cultivation area of 158 million hectares [1] with an annual production of 700 million tons. Rice is one of the main staplefoods in Asian countries like India, Pakistan, China, Indonesia, Bangladesh and so on and these countries alone contribute an annual production of 640 million tons, which is almost 90% of global production. In India it is harvested in three seasons with the names: early Kharif from March to May, medium Kharif from June to October and Rabi from November to February.The possible reasons for reduction in rice yields areabnormal weather conditions, unseasonal rains and plant diseases caused by fungus, bacteria and viruses. Since we have less things to do with environmental conditions, to increase the productivity of paddy to meet the food requirements of increasing population, there is a need for a systematic identification of the disease in paddy plants at least.

Identification of diseases in paddy plants manually may result in errors and become difficult to do with large areas. For accurate detection of disease in large areas, earlier research involved various technologies such as image processing, IoT and Artificial Intelligence (AI) to address this problem. The research into applying AI for detection of diseases in paddy plants started a decade ago. From past research, it has been identified that Deep Convolutional Neural Networks (CNN) outperform other machine learning techniques for the identification of paddy plant disease. This study aims to identify the appropriate CNN architecture for Paddy plant disease identification. Hence, the performance of popular Deep-CNN architectures, including AlexNet, VGG-19, VGG-16, InceptionV3, MobileNet, and ResNet-50 for this multiclass problem was measured. These CNN architecture parameters are fine-tuned to get the maximum accuracy in classifying the plant image in one of the four classes such as healthy, Brown Spot, Hispa, or Leaf Blast. The performance of the above listed models for various hyperparameter configurations and their class predicting ability are discussed briefly in the later part of this study.

The organization of this paper as follows: Existing work is outlined in section II, pre-processing steps are presented section III, and result analysis is presented in the next section, followed by performance evaluation and the conclusions. The main contribution to this work are listed as follows:The performance of popular Deep-CNN models on Paddy plant disease identification is investigated for the Plant Village dataset. AlexNet, VGG-16, VGG-19, Inception V3, MobileNet, and ResNet-50 architectures are experimented with various batch sizes and learning rates.The high-performance CNN architecture suitable for mobile based application is developed and verified with real time paddy images.

2. RELATED WORKS

Earlier research carried out in similar applications helped to find the way for developing the web or mobile application for identifying the most common rice plant diseases. Ruoling Deng et al. [2] used an ensemble model by integrating three best performing models for this application, namely DenseNet-121, SE-ResNet-50, and ResNet-50 to develop a mobile app for rice plant disease detection. The base models are chosen based on their classification performance on a dataset of 33,026 images consisting of rice plant images affected by six types of rice diseases. In [3], Gugan Kathiresan et al. exploited transfer learning and General Adverse Networks (GAN) to develop a mobile app for the detection of disease in rice plants. Upon training and validation with three different dataset namely the rice leaf disease dataset (120 images), the rice disease image dataset (2092 images), and the leaf disease dataset (8293 images), the model achieved a cross validation accuracy of 98.38% without GAN and 98.7% with GAN. Krishnamoorthy et al. [4] explored transfer learning techniques on popular CNNs, namely VGG-16, ResNet-50, and InceptionV3,for classifying the disease-affected rice plant images into four classes. Upon hyperparameter tuning, the chosen architectures achieved the following accuracies for this multiclass classification problem: 87% for VGG-16, 93% for ResNet-50, and 95% for Inception V3. The work carried by Teja et al. [5] exploited different approaches to transfer learning techniques for feature extraction and reused the popularly trained models to minimize the prediction time. These approaches helped to achieve 99.33% testing accuracy for the InceptionV3 model.

In the study conducted by Md. Ashiqul Islam et al. [6], the performance of rice plant disease detection using the following CNN models was analysed: VGG-19, Inception ResNet-V2, ResNet-101 and Xception. The models are expected to classify the given image in any one of the following classes: Brown Spot, Leaf Blast, Bacterial Leaf Blight and Leaf Smut, which are frequent rice plant diseases in Bangladesh. Upon training Inception ResNetV2 with transfer learning, a maximum accuracy of 92.68% was achieved for five class classification, including healthy class. Swathika et al. [7] achieved an accuracy of 70% for two class classification using the ReLU activation function for Deep-CNN. Shivam et al. [8] have compared the LeNet5 model with the VGG-19 and MobileNetV2 models and concluded that MobileNetV2 is a light- weight model with a minimum number of parameters and computations as it uses the depth-wise separable Deep-CNN. They also identified that LeNet will work better for a smaller number of epochs. Shrivastava et al.[9] showed that the VGG-16 with a batch size of 8 and learning rate of 0.01works best for seven class classification tasks, including healthy class, with an accuracy of 93.11% among the compared architectures on Real Image Dataset. Divvela Srinivasa Rao et al. [10] identified the AlexNet model that gave better results for their own dataset.

Herlambang Dwi Prasetyo et al. [11] developed the web-based application using GoogLeNet architecture and Inception modules for rice plant disease detection to be used by end users. Chen et al. [12] used the transfer learning concept by combining the DenseNet pre-trained on ImageNet Dataset with the Inception module and showed good accuracy for the public dataset. Dengshan Li et al. [13] have developed avideo detection architecture to detect disease in rice plants in real time using Deep-CNN for three class classification. Kalai Priya et al. [14] performed the fine tuning of the hybrid algorithms to increase the accuracy of rice plant disease detection using deep learning networks. Rishabh Sharma et al. [15] made an attempt at real-time identification of hispa rice plant disease and its severity from images taken from paddy fields in the state of Punjab, India. Shreya Ghosal et al. [16] improved the testing accuracy of pre-trained VGG-16 model using ImageNet dataset with the transfer learning technique. Hossain et al. [17] proposed a model with minimized network parameters and applied it on a novel dataset consisting of 4199 rice plant images affected by the most common rice plant diseases of Bangladesh namely: Blast, Brown Spot, Bacterial Leaf Blight, Tungro and Sheath Blight, and achieved good accuracy.

S Ramesh et al. [18] have proposed the Jaya algorithm to detect the disease in rice plants using deep neural networks. They have generated a feedback loop as a post-processing step, to achieve the stability of their proposed algorithm. Liang et al. [19] proved that the Deep-CNN outperformed the traditional handcrafted methods with the Rice Blast Disease dataset. They also stated that the performance of CNN techniques with SoftMax and CNN with Support Vector Machine (SVM) is similar. Cham et al. [20] developed a CNN model on TensorFlow for the detection of disease in rice plants. They have also used the ML kit service to store it and implement it as a mobile application. Anandhan K et al. [21] created their own dataset of 1500 images and applied R-CNN and mask R-CNN to detect the diseases in rice plants, proving mask R-CNN performs better than the other one. Zaki et al. [22] have obtained 90% accuracy for disease detection in Tomato plants using leaf images with a fine-tuned MobileNetV2 and stated that MobileNetV2 is suitable for developing a mobile application since the number of registers available on mobile are less compared to the desktop. Ahmed et al. [23] have developed a mobile application for the detection of the 38 most common diseases in 14 plant varieties. This work used an imagery dataset consisting of 96,206 images of healthy and disease affected plants. With the developed real-time application, they have obtained an overall classification accuracy of 94%. Aishwarya et al. [24] proposed a custom CNN model for the detection and identification of Tomato Plant Diseases. Kohli et al. [25] utilized Fuzzy C-means clustering for segmenting the leaf images, followed by feature extraction (color and texture) for detecting the type of disease in the leaf. The CNN also provides good support in the segmentation process, which can be used for leaf disease detection as well [26]. Verma et al. [27] used MobileNetV2 with lightweight depthwise convolutions to improve computation time and accuracy. Juyal et al. [28] adopted the R-CNN mask for correct identification and accurately mask the disease-affected region for fast identification. This work aims to utilize the power of CNN in a rice plant disease detection task.

3. PERFORMANCE OF CNN ARCHITECTURES IN RICE PLANT DETECTION

The proposed work to develop an application for identification of disease in paddy plants, namely Generic Paddy Plant Disease Detector (GP2D2), using CNN architectures involves the list of processes such as loading images, image pre-processing, dataset splitting, applying deep learning algorithms, and classification. These processes are illustrated in the work flow diagram shown in Fig. 1.

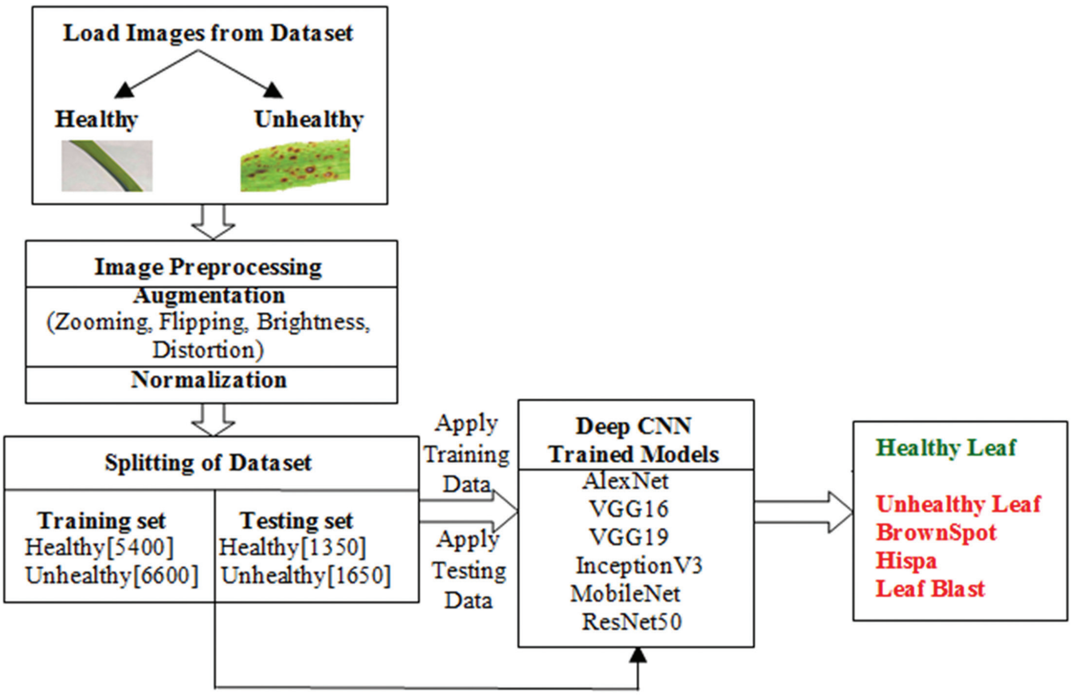

Fig. 1. Workflow of Proposed Rice Plant Disease Detection

a. Load Images from Dataset: The Plant Village dataset is chosen to train and test the CNN architecture since it consists of 3350 images belonging to three generic paddy plant disease classes and a healthy class. The sample images for each class, namely, Healthy, Brown Spot, Hispa and Leaf Blast, are shown in Fig. 2. The details about the number of images in each disease class are represented in Table 1.

b. Image Pre-processing: Augmentation helps to increase the number of images if a smaller dataset is available in order to make the model more generalized and also to avoid the over fitting problem. Hence, the augmentation techniques such as zooming, flipping, brightness, and distortion are used in this work. The images were resized to 224x224x3. Min-max normalization was adopted since this technique preserves the relationship that exists between the original data. This technique scales the data in the range (0, 1).

c. Splitting of Dataset: After applying augmentation the dataset size increased to 15000, and the train-test split technique is used for training and testing the CNN architectures. 80% of the data were used for training the CNN model, and 20% of the data were used for testing the classification performance of the same. Table 1 gives the number of images used for training and testing in each class.

d. Training of Pre-trained CNN Models: The six pre-trained models exploited in this work are: (i) AlexNet, (ii) VGG-19, (iii) VGG-16, (iv) InceptionV3, (v) MobileNet, and (vi) ResNet-50. The above models were trained and tested on the images of the Plant Village dataset and its augmented versions. The performance of the CNN architectures is observed with different combinations of hyper parameters: (i) Learning rate = 0.01 and Batch size = 32; (ii) Learning rate = 0.01 and Batch size = 64; (iii) Learning rate = 0.001 and Batch size = 32; and (iv) Learning rate = 0.001 and Batch size = 64. The complete details of the hyper parameters used in this study are given in Table 2.

Table 1. Dataset description of Augmented Images

Class	No. of Original Images (O)	No. of Images in each Augmentation Method				Total no. of Augmentation Images (A)	Total Images (O+A)	Training Images	Testing Images
		Zooming	Flipping	Brightness	Distortion				
Healthy	1488	2160	1080	1600	422	5262	6750	5400	1350
Brown Spot	523	695	338	492	171	1696	2219	1775	444
Hispa	565	784	402	603	206	1995	2560	2048	512
LeafBlast	779	1026	544	821	301	2692	3471	2777	694

The sample images of each class, before and after augmentation techniques are shown in Fig. 2.

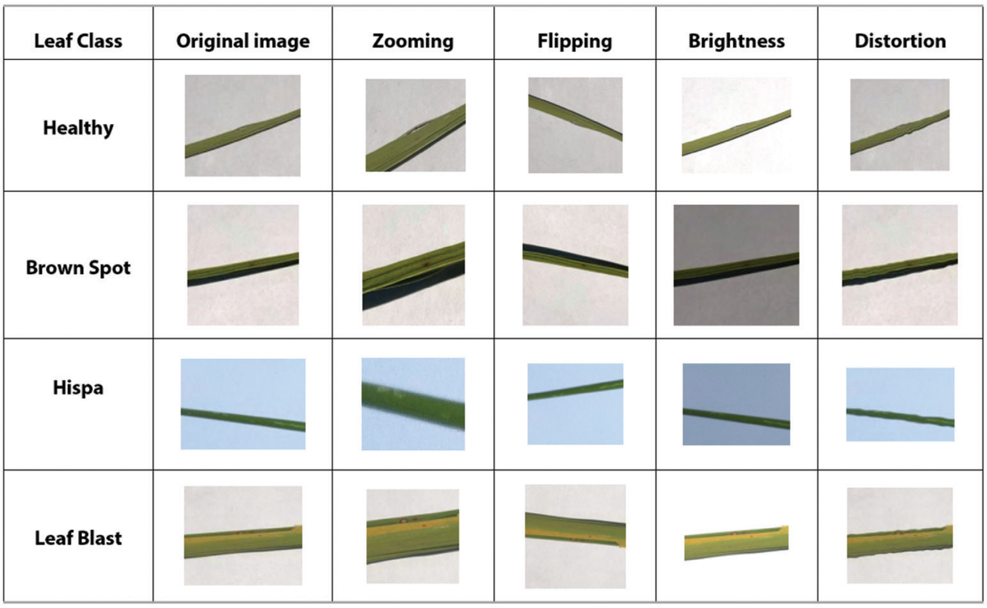

Fig. 2. Augmented images of paddy leaf diseases

Table 2. Hyper Parameters of the CNN Architecture

Hyper Parameters	Description
Batch size	32, 64
Learning rate	0.01, 0.001
Number of epochs	100
Type of optimizer	Adam
Activation function	ReLU
Loss	Categorical Cross Entropy

e. Performance Analysis

To analyse the performance of the above mentioned pre-trained architectures, the following metrics are used: Accuracy, Precision, Sensitivity, Specificity and F1 Score, which are shown in eqns.(1)-(5). The metric values are estimated as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Sensitivityor Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (4)$$

$$\text{F1 Score} = \frac{2TP}{2TP+FP+FN} \quad (5)$$

Where True Positive (TP) is the number of disease- affected images correctly predicted in its actual plant disease class, True Negative (TN) is the number of healthy plant images correctly predicted as healthy class, False Positive (FP) is the number of images not predicted correctly in its actual disease class or healthy class and False Negative (FN) is the number of disease-affected plant images wrongly predicted as healthy.

From Table 3, it is observed that AlexNet performed best with an average testing accuracy of 89.4% of batch size of 32 and learning rate of 0.01. Next, VGG-16 with 87.1% for 32, 0.01, followed by VGG-19 with 86.7% for 64, 0.001, ResNet-50 with 86.1% for 32, 0.001, MobileNet with 77.3% for 64, 0.01, and InceptionV3 with 42.4% for 32 batch size and 0.01 learning rate. From Table 4, it is observed that for batch size of 64, the VGG-19, MobileNet architectures performed well, and the other four architectures were performed well for batch size of 32. With a learning rate of 0.01, AlexNet, InceptionV3 performed well, while others performed well for 0.001. In Fig. 3, the computation time for the batch size of 32 and 64 with the learning rate of 0.01 and 0.001 for the pre-trained CNN architectures is shown for the different hyperparameters. The performance metrics obtained using pre-trained CNN models for four different combinations of Batch Size (BS) and Learning Rate (LR) were compared.

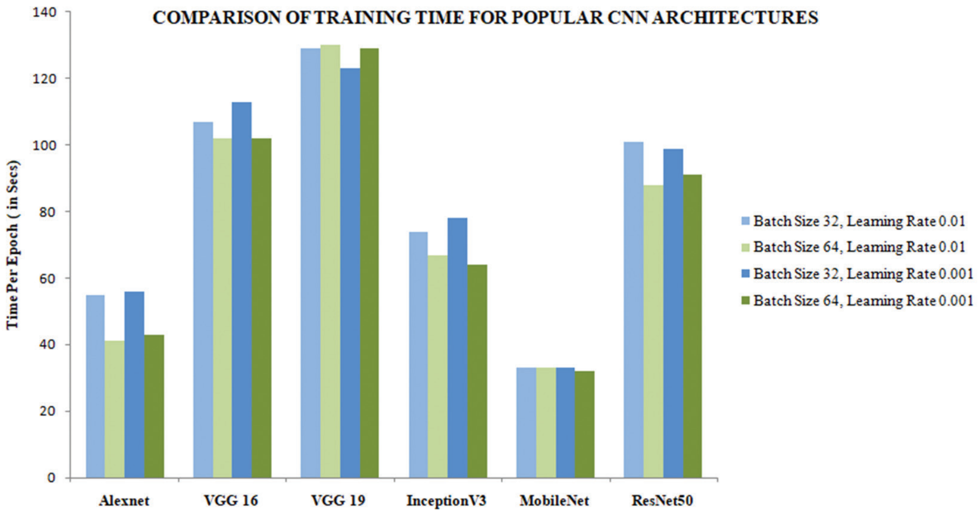

Fig. 3. Computation time of pre-trained CNN architectures for different hyperparameters

Table 3. Performance Comparison of Pre-trained CNN Architectures on Generic Paddy Plant Disease Detection

Model	Batch size	Learning Rate	Testing Accuracy	F1score	Precision	Recall	Specificity	Loss
AlexNet	32	0.01	0.8940	0.8939	0.8956	0.8923	0.8777	0.62
	64	0.01	0.8553	0.8552	0.8560	0.8544	0.8336	0.75
	32	0.001	0.8793	0.8789	0.8823	0.8757	0.8711	0.59
	64	0.001	0.8553	0.8549	0.8568	0.8531	0.8326	0.69
VGG-16	32	0.01	0.7080	0.6935	0.7396	0.6538	0.6635	1.06
	64	0.01	0.7433	0.7412	0.7615	0.7224	0.6837	1.15
	32	0.001	0.8713	0.8721	0.8731	0.8711	0.8520	1.10
	64	0.001	0.8647	0.8648	0.8651	0.8645	0.8520	1.38
VGG-19	32	0.01	0.7207	0.7128	0.7593	0.6728	0.8457	1.06
	64	0.01	0.7447	0.7427	0.7623	0.7243	0.8458	1.09
	32	0.001	0.8513	0.8510	0.8517	0.8503	0.8368	1.13
	64	0.001	0.8670	0.8675	0.8686	0.8664	0.8479	1.31
InceptionV3	32	0.01	0.4253	0.4248	0.4249	0.4246	0.3711	21.19
	64	0.01	0.4207	0.4207	0.4209	0.4205	0.3436	12.94
	32	0.001	0.2557	0.2558	0.2579	0.2538	0.2968	4.72
	64	0.001	0.2553	0.2535	0.2556	0.2515	0.2715	3.24
MobileNet	32	0.01	0.5410	0.1391	0.7498	0.0783	0.4102	1.10
	64	0.01	0.6053	0.5900	0.6205	0.5627	0.4806	1.01
	32	0.001	0.7700	0.7704	0.7753	0.7657	0.7462	1.08
	64	0.001	0.7733	0.7735	0.7670	0.7670	0.7302	1.06

ResNet-50	32	0.01	0.7620	0.7620	0.7763	0.7487	0.7011	0.97
	64	0.01	0.7907	0.7916	0.7955	0.7878	0.7468	1.21
	32	0.001	0.8610	0.8614	0.8620	0.8608	0.8402	1.58
	64	0.001	0.8583	0.8587	0.8590	0.8584	0.8406	1.26

4. RESULTS AND DISCUSSION

The summary of the performance comparison of the popular pre-trained models is presented in Table 5. AlexNet achieves highest accuracy among others, but it requires around 92 million parameters.VGG-16, VGG-19, and ResNet-50 require approximately 15 million to 24 million parameters to achieve accuracy closer to AlexNet. InceptionV3 shows very poor performance; hence, it may not be considered for this classification task. MobileNet requires a minimum number of parameters of around 4 million in order to achive accuracy. During the process of saving the adopted CNN on disk, the value of each and every single weight in the network layers is recorded. A network consisting of hundreds of millions of parameters, such as AlexNet, might occupy hundreds of megabytes, which is unacceptable on a mobile device or phone.

One solution for the mentioned problem is to reduce the resolution of the weights through quantization.

This process enables to store only the quantized values once and is used for assigning the network weights. But this process will reduce the architecture is the more preferable one for developing web app and will be a suggested one rather than implementing it as amobile application. MobileNet is an in-built mobile application that does not require accuracy, increases processing time, and also requires high computational power. Implementing the classification model in the backend of a cloud-based internet connection for rice plant disease detection has lesser classification accuracy than AlexNet and others. If accuracy is given more importance, then the targeted user needs to have good internet facilities to access the cloud-based web app through a smartphone, which has an AlexNet model as a backend model for prediction. VGG-16,VGG-19, and ResNet-50 also can be used as backend models.The Confusion matrix and the Area Under the Curve (AUC) of the best performing architectures with those particular batch sizes and learning rates are depicted in Fig. 4.

Table 4. Class Wise Performance of CNN Architecture on Generic Paddy Plant Disease Detection

Model	Batch size	Learning rate	Healthy	Brown Spot	Hispa	Leaf Blast
AlexNet	32	0.01	92.33	96.97	94.70	94.90
	64	0.01	90.14	96.57	93.13	92.73
	32	0.001	91.10	96.70	93.87	94.20
	64	0.001	88.66	96.80	93.07	92.53
VGG-16	32	0.01	76.27	93.37	85.27	86.70
	64	0.01	83.40	93.93	87.03	88.37
	32	0.001	90.13	96.93	93.27	93.93
	64	0.001	91.00	96.67	93.07	93.43
VGG-19	32	0.01	90.13	93.47	85.70	85.90
	64	0.01	89.77	94.27	93.07	93.43
	32	0.001	89.33	96.13	92.33	92.47
	64	0.001	90.57	96.33	93.10	93.40
InceptionV3	32	0.01	56.70	82.00	68.50	77.87
	64	0.01	54.47	85.37	66.57	77.73
	32	0.001	54.73	85.73	80.47	30.20
	64	0.001	55.37	85.83	83.07	26.80
MobileNet	32	0.01	59.00	91.13	83.03	75.03
	64	0.01	64.23	91.70	83.03	82.10
	32	0.001	81.70	95.00	90.33	88.23
	64	0.001	81.67	94.67	88.97	89.37
ResNet-50	32	0.01	81.37	94.37	87.13	89.63
	64	0.01	83.83	95.83	88.13	90.33
	32	0.001	89.17	96.53	93.40	94.00
	64	0.001	88.63	96.77	92.23	94.03

Table 5. Summary of Pre-trained CNN Architectures

Architecture	Accuracy	Total No of Parameters	More Compatible for Mobileapp/Webapp
AlexNet	89.40	91,784,764	Webapp
VGG-16	87.13	14,879,428	Webapp
VGG-19	86.70	20,189,124	Webapp
InceptionV3	42.53	21,810,980	Webapp
MobileNet	77.33	4,282,564	Mobileapp/Webapp
ResNet-50	86.10	24,145,668	Webapp

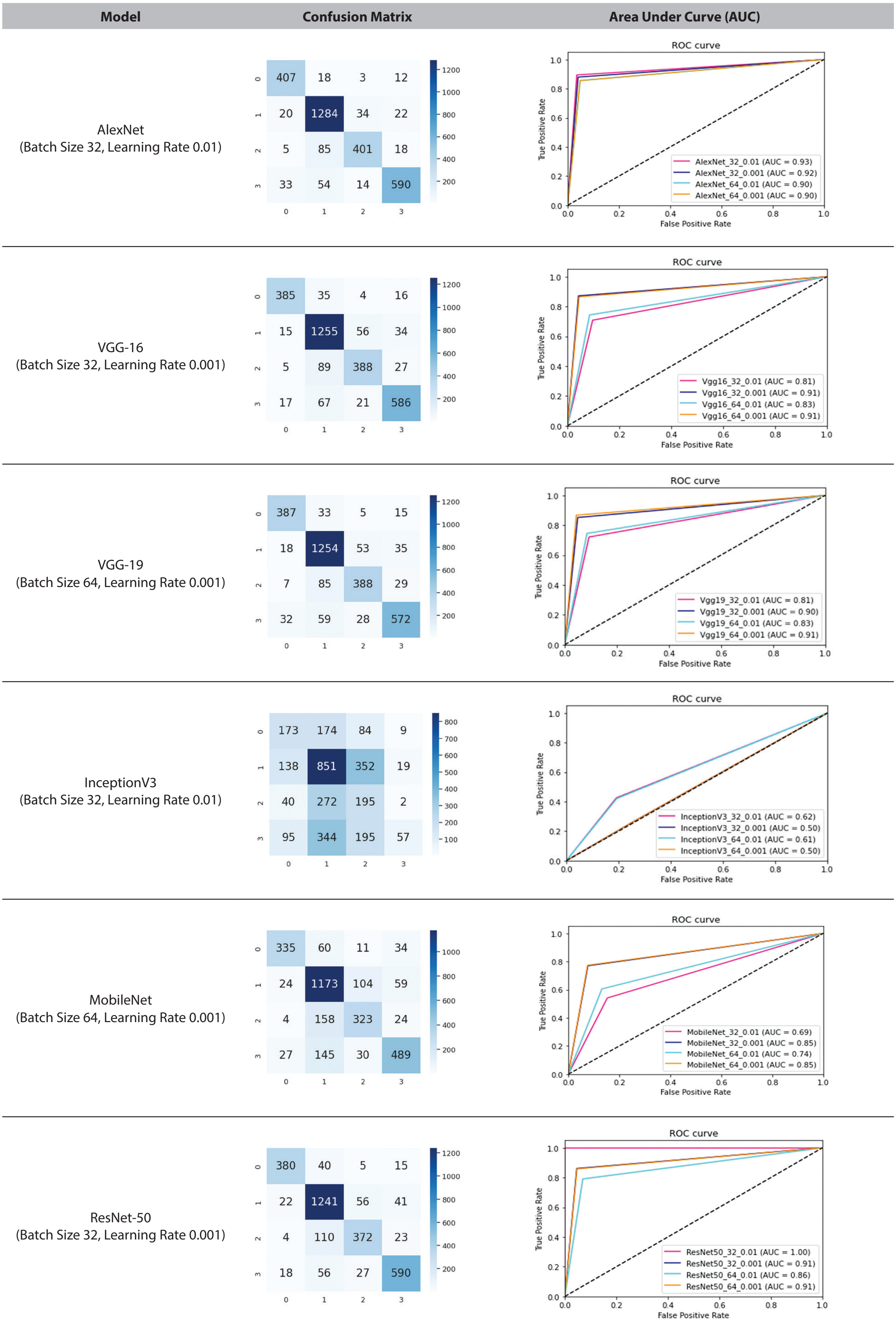

Fig. 4. Classification performance of various pertained CNN architecures

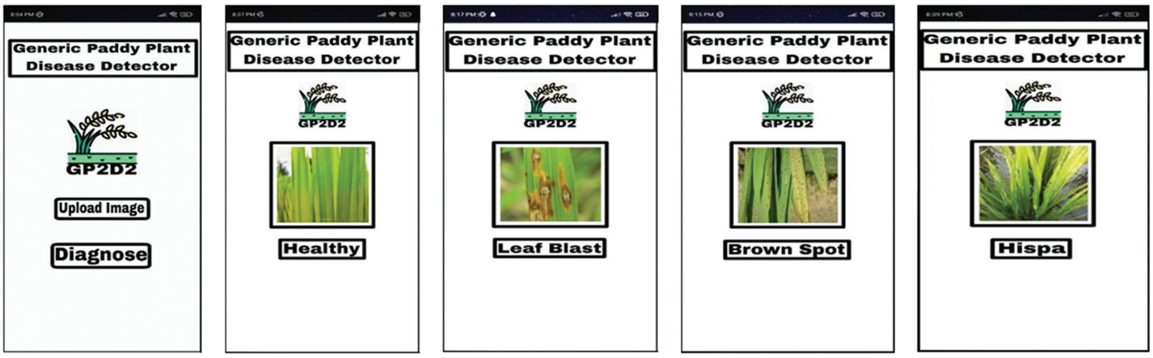

Fig. 5. Screenshots for mobile app home screen, healthy and diseased paddy leaves

5. DEVELOPMENT OF MOBILE APP: GENERIC PADDY PLANT DISEASE DETECTOR (GP2D2)

To make use of this study results a cloud based web application compatible with mobile was developed using AlexNet CNN architecture namely Generic Paddy Plant Disease Detector (GP2D2). It is an Android Mobile Application developed using Flutter with Android Studio as an Integrated Development Environment (IDE). The mobile application will help users to determine their plant health with a hassle-free process. Flutter is a free and open-source mobile UI framework created by Google for crafting natively compiled applications for mobile, web, and desktop forGoogle's Android operating system, built on JetBrains from a single codebase. Android Studio is the official IDE IntelliJ IDEA software and designed specifically for Android development. The GP2D2 mobile application allows users to capture images of the paddy plant. The user can upload the captured images to Firestore. Firestore is a cloud-hosted NoSQL database from Firebase and Google Cloud, where Android apps can access directly via native SDKs. The screenshots for the mobile app have been presented in the Fig. 5 representing home screen of the app, healthy leaves, Brown Spot, Leaf Blast and Hispa diseased leaves respectively.

6. CONCLUSION

This paper analyses the classification performance of popular CNN architectures in rice plant disease identification tasks. The results observed, AlexNet with highest accuracy of 89.4%, VGG-16, VGG-19 and ResNet-50 showed comparable accuracy. MobileNet can be considered for developing in-built mobile applications. The developed Generic Paddy Plant Disease Detector (GP2D2) application helps new-age farmers acquire the expert traditional farmer's experience in detecting the type of paddy disease digitally. Conventional methods for identification of disease in paddy plants are less effective when considered in a large area, and the developed mobile application will be more helpful in this case. Drones with cameras can be used for capturing the sample paddy images, which can be fed to the app for disease identification. This study will provide a handful of information on identifying suitable architecture for design and developing a real-time application for the identification of common diseases in paddy plants. This mobile application framework can be easily modified just by updating or replacing the existing model based on the requirements. This work will help the farmers easily identify the spread of paddy plant diseases in their early stages and take the necessary steps to avoid their spread. Successful implementation of this work will increase the Yield Per Hectare (YPH) of paddy plants and may improve the financial status. This work can be extended further to determine severity of disease conditions using semantic segmentation techniques through CNN [25].

7. REFERENCES:

[1] A Project of CGIAR, "Research Program on Rice", https://ricepedia.org/rice-as-a-crop/rice-productivity (accessed: 2023)

[2] R. Deng, M. Tao, H. Xing, C. Liu, K. Liao, L. Qi, "Automatic Diagnosis of Rice Diseases using Deep Learning", Frontiers in Plant Ccience, Vol. 12, 2021, pp. 1-15.

[3] G. Kathiresan, M. Anirudh, M. Nagharjun, R. Karthik, "Disease detection in rice leaves using transfer learning techniques", Journal of Physics: Conference Series, Vol. 1911, 2021, pp. 1-10.

[4] K. U. V. R. Teja, B. P. V. Reddy, L. R. Kesara, K. D. P. Kowshik, L. A. Panchaparvala, "Transfer Learning based Rice Leaf Disease Classification with Inception-V3," Proceedings of the International Conference on Smart Generation Computing, Communication and Networking, Pune, India, 29-30 October 2021, pp. 1-6.

[5] N. Krishnamoorthy, V. Parameswari, "Rice Leaf Disease Detection Via Deep Neural Networks with Transfer Learning for Early Identification", Turkish Journal of Physiotherapy and Rehabilitation, Vol. 32, No. 32, 2021, pp. 1087-1097.

[6] Md. A. Islam, Md. NymurRahmanShuvo, M. Shamsojjaman, ShazidHasan, Md. S. Hossain, T. Khatun, "An Automated Convolutional Neural Network based Approach for Paddy Leaf Disease Detection", International Journal of Advanced Computer Science and Applications, Vol. 12, No. 1, 2021, pp. 280-288.

[7] R. Swathika, S. Srinidhi, N. Radha, K. Sowmya, "Disease Identification in Paddy Leaves using CNN based Deep Learning", Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, Tirunelveli, India, 4-6 February 2021, pp. 1004-1008.

[8] S. P. S. Shivam, I. Kumar, "Rice Plant Infection Recognition using Deep Neural Network Systems", Proceedings of the 3rd International Semantic Intelligence Conference, New Delhi, India, February 25-27 2021, pp. 384-393.

[9] V. K. Shrivastava, M. K. Pradhan, M. P. Thakur, "Application of Pre-Trained Deep Convolutional Neural Networks for Rice Plant Disease Classification", Proceedings of the International Conference on Artificial Intelligence and Smart Systems, Coimbatore, India, 25-27 March 2021, pp. 1023-1030.

[10] D. S. Rao, N. Kavya, S. N. Kumar, L. Y. Venkat, N. P. Kumar, "Detection and Classification of Rice Leaf Diseases using Deep Learning", International Journal of Advanced Science and Technology, Vol. 29, No. 3, 2020, pp. 5868- 5874.

[11] H. D. Prasetyo, H. Triatmoko, Nurdiansyah, I. N. Isnainiyah, "The Implementation of CNN on Website-based Rice Plant Disease Detection", Proceedings of the International Conference on Informatics, Multimedia, Cyber and Information System, Jakarta, Indonesia, 19-20 November 2020, pp. 75-80.

[12] J. Chen, J. Chen, D. Zhang, Y. Sun, Y. Nanehkaran, "using Deep Transfer Learning for Image-based Plant Disease Identification", Computers and Electronics in Agriculture, Vol. 173, No. 105393, 2020.

[13] D. Li, R. Wang, C. Xie, L. Liu, J. Zhang, R. Li, F. Wang, M. Zhou, W. Liu, "A Recognition Method for Rice Plant Diseases and Pests Video Detection Based on Deep Convolutional Neural Network", Sensors, Vol. 20, No. 3, 2020, pp. 1-21.

[14] M. KalaiPriya, S. Dhanabal, "Analyses of Nine Different Types of Diseases in Paddy with Hybrid Algorithms using Deep Learning", International Journal of Engineering Research & Technology, Vol. 8, No. 8, 2020, pp. 1-7.

[15] R. Sharma, V. Kukreja, V. Kadyan, "Hispa Rice Disease Classification using Convolutional Neural Network", Proceedings of the 3rd International Conference on Signal Processing and Communication, Coimbatore, India, 13-14 May 2021, pp. 377-381.

[16] S. Ghosal, K. Sarkar, "Rice Leaf Diseases Classification using CNN with Transfer Learning", Proceedings of the IEEE Calcutta Conference, Calcutta, India, 28-29 February 2020, pp. 230-236.

[17] S. M. M. Hossain, Md. M. M. Tanjil, M. A. B. Ali, M. Z. Islam, Md. S. Islam, S. Mobassirin, I. H. Sarker, S. M. R. Islam, "Rice Leaf Diseases Recognition using Convolutional Neural Networks", Advanced Data Mining and Applications, Springer International Publishing, 2020, pp. 299-314.

[18] S. Ramesh, D. Vydeki, "Recognition and Classification of Paddy leaf Diseases using Optimized Deep Neural network with Jaya algorithm", Information Processing in Agriculture, Vol. 7, No. 2, 2020, pp. 249-260.

[19] Wj. Liang, H. Zhang, Gf. Zhang, C. Hong-xin, "Rice Blast Disease Recognition using a Deep Convolutional Neural Network", Scientific Reports, Vol. 9, No. 2869, 2019, pp. 1-10.

[20] M. F. X. Cham, R. Tanone, H. A. T. Riadi, "Identification of Rice Leaf Disease using Convolutional Neural Network Based on Android Mobile Platform", Proceedings of the 2nd International Conference on Innovative and Creative Information Technology, Salatiga, Indonesia, 23-25 September 2021, pp. 140-144.

[21] K. Anandhan, A. S. Singh, "Detection of Paddy Crops Diseases and Early Diagnosis using Faster Regional Convolutional Neural Networks", Proceedings of the International Conference on Advance Computing and Innovative Technologies in Engineering, Greater Noida, India, 4-5 March 2021, pp. 898-902.

[22] S. Z. M. Zaki, Md. A. Zulkifley, M. Md. Stofa, N. A. M. Kamari, N. A. Mohamed, "Classification of Tomato

Leaf Diseases using Mobile Net v2", IAES International Journal of Artificial Intelligence, Vol. 9, 2020, pp. 290-296.

[23] A. A. Ahmed, G. H. Reddy, "A Mobile-Based System for Detecting Plant Leaf Diseases using Deep Learning", Agri Engineering, Vol. 3, 2021, pp. 478-493.

[24] N. Aishwarya, N. G. Praveena, S. Priyanka, J. Pramod, "Smart Farming for Detection and Identification of Tomato Plant Diseases using Light Weight Deep Neural Network", Multimedia Tools and Applications, Vol. 82, 2023, pp. 18799-18810.

[25] P. Kohli, I. Kumar, V. Vimal, "Plant Leaf Disease Identification Using Unsupervised Fuzzy C-Means Clustering and Supervised Classifiers", Cyber Security in Intelligent Computing and Communications.Studies in Computational Intelligence, Springer, Singapore, Vol 1007, 2022.

[26] K. Sivachandar, V. Thulasi Bai, L. Simone, G. Dimauro, "Semantic Segmentation of Conjunctiva Region for Non-Invasive Anemia Detection Applications", Electronics, Vol. 9, No. 8, 2020, pp. 1309-1322.

[27]D. Verma, D. Bordoloi, V. Tripathi, "Plant Leaf Disease Detection Using Mobilenetv2", Webology, Vol. 18, No. 5, 2021.

[28] P. Juyal and S. Sharma, "Detecting the Infectious Area Along with Disease Using Deep Learning in Tomato Plant Leaves", Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, Thoothukudi, India, 3-5 December 2020, pp. 328-332.