Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Biosystems ENGINEERING

Check for updates
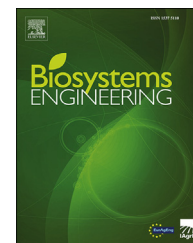
Research Paper

Identification and recognition of rice diseases and pests using convolutional neural networks

Chowdhury R. Rahman [c,], Preetom S. Arko [a], Mohammed E. Ali [a],*
Mohammad A. Iqbal Khan [b], Sajid H. Apon [a], Farzana Nowrin [b],
Abu Wasif [a]

[a] *Bangladesh University of Engineering and Technology, Dhaka, Bangladesh*
[b] *Bangladesh Rice Research Institute, Gazipur, Bangladesh*
[c] *United International University, Dhaka, Bangladesh*

ARTICLE INFO

ABSTRACT

Accurate and timely detection of diseases and pests in rice plants can help farmers in applying timely treatment on the plants and thereby can reduce the economic losses substantially. Recent developments in deep learning-based convolutional neural networks (CNN) have greatly improved image classification accuracy. Being motivated by the success of CNNs in image classification, deep learning-based approaches have been developed in this paper for detecting diseases and pests from rice plant images. The contribution of this paper is two fold: (i) State-of-the-art large scale architectures such as VGG16 and InceptionV3 have been adopted and fine tuned for detecting and recognising rice diseases and pests. Experimental results show the effectiveness of these models with real datasets. (ii) Since large scale architectures are not suitable for mobile devices, a two-stage small CNN architecture has been proposed, and compared with the state-of-the-art memory efficient CNN architectures such as MobileNet, NasNet Mobile and SqueezeNet. Experimental results show that the proposed architecture can achieve the desired accuracy of 93.3% with a significantly reduced model size (e.g., 99% smaller than VGG16).

1. Introduction

Rice occupies about 70 percent of the gross crop area and accounts for 93 percent of total cereal production in Bangladesh (Coelli, Rahman, & Thirtle, 2002). Rice also ensures food security of over half the world population (Calpe, 2002). Researchers have observed 10—15% average yield loss because of 10 major diseases of rice in Bangladesh (Mahmud, Hossain, & Ahmad, 2016). Timely detection of rice plant diseases and pests is one of the major challenges in agriculture. Hence, there is a need for automatic rice disease detection using readily available mobile devices in rural areas.

Deep learning techniques have shown great promise in image classification. In recent years, these techniques have

been used to analyse diseases of tea (Karmokar, Ullah, Siddiquee, & Alam, 2015), apple (Wang, Sun, & Wang, 2017), tomato (Fuentes, Yoon, Kim, & Park, 2017), grapevine, peach, and pear (Sladojevic, Arsenovic, Anderla, Culibrk, & Ste-fanovic, 2016). Babu and Rao (2007) proposed a feed forward back propagation neural network from scratch in order to detect the species of plant from leaf images. Neural network ensemble (NNE) was used by Karmokar, Ullah, Siddiquee, and Alam (2015) to recognise five different diseases of tea plant from leaves. Bhagawati, Bhagawati, Singh, Nongthombam, Sarmah, and Bhagawati (2015) trained a neural network with weather parameters such as temperature, relative humidity, rainfall and wind speed to forecast rice blast disease. Mohanty, Hughes, and Salathé (2016) used deep CNN to detect disease from leaves using 54,306 images of 14 crop species representing 26 diseases, while Sladojevic, Arsenovic, Anderla, Culibrk, and Stefanovic (2016) used CaffeNet model to recognize 13 different types of plant diseases. Wang, Sun, and Wang (2017) worked on detecting four severity stages of apple black rot disease using *PlantVillage* dataset. They used CNN architectures with different depths and implemented two different training methods on each of them. A real time tomato plant disease detector was built using deep learning by Fuentes, Yoon, Kim, and Park (2017). Brahimi, Boukhalfa, and Moussaoui (2017) used fine-tuned AlexNet and GoogleNet to detect nine diseases of tomatoes. Cruz, Luvisi, De Bellis, and Ampatzidis (2017) injected some texture and shape features to the fully connected layers placed after the convolutional layers so that the model can detect Olive Quick Decline Syndrome effectively from the limited dataset. Instead of resizing images to a smaller size and training a model end-to-end, DeChant, Wiesner-Hanks, Chen, Stewart, Yosinski, and Gore (2017) used a three stage architecture (consisting of multiple CNNs) and trained the stage-one model on full scaled images by dividing a single image into many smaller images. Barbedo (2018) used transfer learning on GoogleNet to detect 56 diseases infecting 12 plant species. Using a dataset of 87,848 images of leaves captured both in laboratory and in the field, Ferentinos (2018) worked with 58 classes containing 25 different plants. Liu, Zhang, He, and Li (2018) built a CNN combining the ideas of AlexNet and GoogLeNet to detect four diseases of apple. Images of individual lesions and spots instead of image of whole leaf were used by Barbedo (2019) for identifying 79 diseases of 14 plant species. Few researches have also been conducted on rice disease classification (Atole & Park, 2018; Lu, Yi, Zeng, Liu, & Zhang, 2017). Lu et al. (2017) conducted a study on detecting 10 different rice plant diseases using a small handmade CNN architecture, inspired by older deep learning frameworks such as LeNet-5 and AlexNet, using 500 images. Atole and Park (2018) used AlexNet (large architecture) to distinguish among three classes - normal rice plant, diseased rice plant and snail infested rice plant using 227 images.

The research mentioned above mainly focused on accurate plant disease recognition and classification. For this purpose, they implemented various types of CNN architectures such as AlexNet, GoogLeNet, LeNet-5 and so on. In some studies, ensemble of multiple neural network architectures have been used. These studies played an important role for automatic and accurate recognition and classification of plant diseases. But their focus was not on modifying the training method for the models that they had constructed and used. Moreover, they did not consider the impact of the large number of parameters of these high performing CNN models in real life mobile application deployment.

In this research, two state-of-the-art CNN architectures, VGG16 and InceptionV3, have been tested in various settings. Fine tuning, transfer learning and training from scratch have been implemented to assess their performance. In both the architectures, fine tuning the model while training has shown the best performance. Though these deep learning-based architectures perform well in practice, a major limitation of these architectures is that they have a large number of parameters, a problem similar to previously conducted researches. For example, there are about 138 million parameters in VGG16 (Simonyan & Zisserman, 2014). In remote areas of developing countries, farmers do not have internet connectivity or have slow internet speed. So, a mobile application capable of running CNN-based model offline is needed for rice disease and pest detection. So, a memory efficient CNN model with reasonably good classification accuracy is required. Since the reduction of the number of parameters in a CNN model reduces its learning capability, one needs to make a trade-off between memory requirement and classification accuracy to build such a model.

To address the above issue, in this paper, a new training method called **two stage training** has been proposed. A CNN architecture, named **Simple CNN**, has been proposed which achieves high accuracy leveraging two stage training in spite of its small number of parameters. Experimental study shows that the proposed **Simple CNN** model outperforms state-of-the-art memory efficient CNN architectures such as MobileNet, NasNet Mobile and SqueezeNet in recognising rice plant diseases and pests.

All training and validation have been conducted on a rice dataset collected in real life scenario as part of this research. A rice disease may show different symptoms based on various weather and soil conditions. Similarly, pest attack can show different symptoms at different stages of an attack. Moreover, the diseases and pests can occur at any part of the plant which include leaf, stem and grain. Images can also be of heterogeneous background. This research addresses all these issues while collecting data. This paper focuses on recognizing eight different rice plant diseases and pests that occur at different times of the year at Bangladesh Rice Research Institute (BRRI). This work also includes a ninth class for non-diseased rice plant recognition.

In summary, this paper makes two important contributions in rice disease and pest detection. First, state-of-the-art large scale deep learning frameworks have been tested to investigate the effectiveness of these architectures in rice plant disease and pest identification from images collected from real-life environments. Second, a novel two-stage training based light-weight CNN has been proposed that is highly effective for mobile device based rice plant disease and pest detection. This can be an effective tool for farmers in remote environment.

2. Materials and methods

2.1. Data collection

Rice diseases and pests occur in different parts of the rice plant. Their occurrence depends on many factors such as temperature, humidity, rainfall, variety of rice plants, season, nutrition, etc. An extensive exercise was undertaken to collect total 1426 images of rice diseases and pests from paddy fields of Bangladesh Rice Research Institute (BRRI). Images have been collected in real life scenario with heterogeneous backgrounds from December, 2017 to June, 2018 for a total of seven months. The image collection has been performed in a range of weather conditions - in winter, in summer and in overcast condition in order to get as fully representative a set of images as possible. Four different types of camera have been used in capturing the images. These steps increase the robustness of our model. This work encompasses a total of five classes of diseases, three classes of pests and one class of healthy plant and others - a total of nine classes. The class names along with the number of images collected for each class are shown in Table 1. Note that Sheath Blight, Sheath Rot and their simultaneous occurrence have been considered in the same class, because their treatment method and place of occurrence are the same.

Symptoms of different diseases and pests are seen in different parts such as leaf, stem and grain of the rice plant. Bacterial Leaf Blight disease, Brown Spot disease, Brown Plant Hopper pest (late stage) and Hispa pest occur on the rice leaf. Sheath Blight disease, Sheath Rot disease and Brown Plant Hopper pest (early stage) occur on the rice stem. Neck Blast disease and False Smut disease occur on rice grain. Stemborer pest occurs on both rice stem and rice grain. All these aspects have been considered while capturing images. To prevent classification models from being confused between dead parts

Table 1 — Image collection of different classes.	
Class Name	No. of Collected Images
False Smut	93
Brown Plant Hopper (BPH)	71
Bacterial Leaf Blight (BLB)	138
Neck Blast	286
Stemborer	201
Hispa	73
Sheath Blight and/or Sheath Rot	219
Brown Spot	111
Others	234

(a) Bacterial Leaf Blight (Disease)　(b) Brown Plant Hopper (Pest)　(c) Brown Spot (Disease)

(d) False Smut (Disease)　(e) Stemborer (Pest)　(f) Hispa (Pest)

(g) Neck Blast (Disease)　(h) Sheath Blight (Disease)　(i) Healthy Leaf (Others)

Fig. 1 — A sample image of each detected class.

and diseased parts of rice plant, images of dead leaf, dead stem and dead grain of rice plants have been incorporated into the dataset. For example, diseases like BLB, Neck Blast and Sheath Blight have similarity with dead leaf, dead grain and dead stem of rice plant respectively. Thus images of dead leaf, dead stem and dead grain along with images of healthy rice plant have been considered in a class that has been named **others**. Sample images of each class have been depicted in Fig. 1.

False Smut, Stemborer, Healthy Plant class, Sheath Blight and/or Sheath Rot class show multiple types of symptoms. Early stage symptoms of Hispa and Brown Plant Hopper are different from their later stage symptoms. All symptom variations of these classes found in the paddy fields of BRRI have been covered in this work. These intra-class variations have been described in Table 2. BLB, Brown Spot and Neck Blast disease show no considerable intra-class variation around BRRI area. An illustrative example for Hispa pest has been given in Fig. 2.

2.2. Experimental setup

Keras framework with tensorflow back-end has been used to train the models. Experiments have been conducted with two state-of-the-art CNN architectures containing large numbers of parameters such as **VGG16** and **InceptionV3**. Later the proposed light-weight two-stage **Simple CNN** has been tested and compared with three state-of-the-art memory efficient CNN architectures such as MobileNetv2, NasNet Mobile and SqueezeNet. VGG16 (Simonyan & Zisserman, 2014) is a sequential CNN architecture using 3×3 convolution filters. After each maxpooling layer, the number of convolution filters gets doubled in VGG16. InceptionV3 (Szegedy et al., 2015) is a non-sequential CNN architecture consisted of inception blocks. In each inception block, convolution filters of various dimensions and pooling are used on the input in parallel. The number of parameters of these five architectures along with **simple CNN** architecture have been given in Table 3. Three

Table 3 — State-of-the-art CNN architectures and their parameter no.

CNN Architecture	No. of Parameters
VGG16	138 million
InceptionV3	23.8 million
MobileNetv2	2.3 million
NasNet Mobile	4.3 million
SqueezeNet	0.7 million
Simple CNN	0.8 million

Table 2 — Intra-class variation in some diseases and pests.

Class Name	Symptom Variation	Sample No.
BPH	Early stage of Brown Plant Hopper attack	50
	Late stage of Brown Plant Hopper attack	21
False Smut	Brown symptom	66
	Black symptom	27
Others	Healthy green leaf and stem	96
	Healthy yellow grain	71
	Dead leaf and stem	67
Hispa	Visible black pests and white spot on plant leaf	53
	No visible pest, intense spot on leaves	20
Stemborer	Symptom on grain	180
	Symptom on stem	21
Sheath Blight and/or Sheath Rot	Black Stem	70
	White spots	77
	black and white symptom mixed	72

Fig. 2 — Hispa Variations: Image on the left has visible black pests and white spots on the plant leaf which occur during early stage of Hispa attack. Image on the right has intense spots on leaves with no visible pest occurring during later stage of Hispa attack.

different types of training methods have been implemented on each of these five architectures.

Baseline training: All randomly initialised architecture layers are trained from scratch. This method of training takes time to converge.

Fine Tuning: The convolution layers of the CNN architectures are trained from their pre-trained ImageNet weights, while the dense layers are trained from randomly initialised weights.

Transfer Learning: In this method, the convolution layers of the CNN architectures are not trained at all. Rather pre-trained ImageNet weights are kept intact. Only the dense layers are trained from their randomly initialised weights.
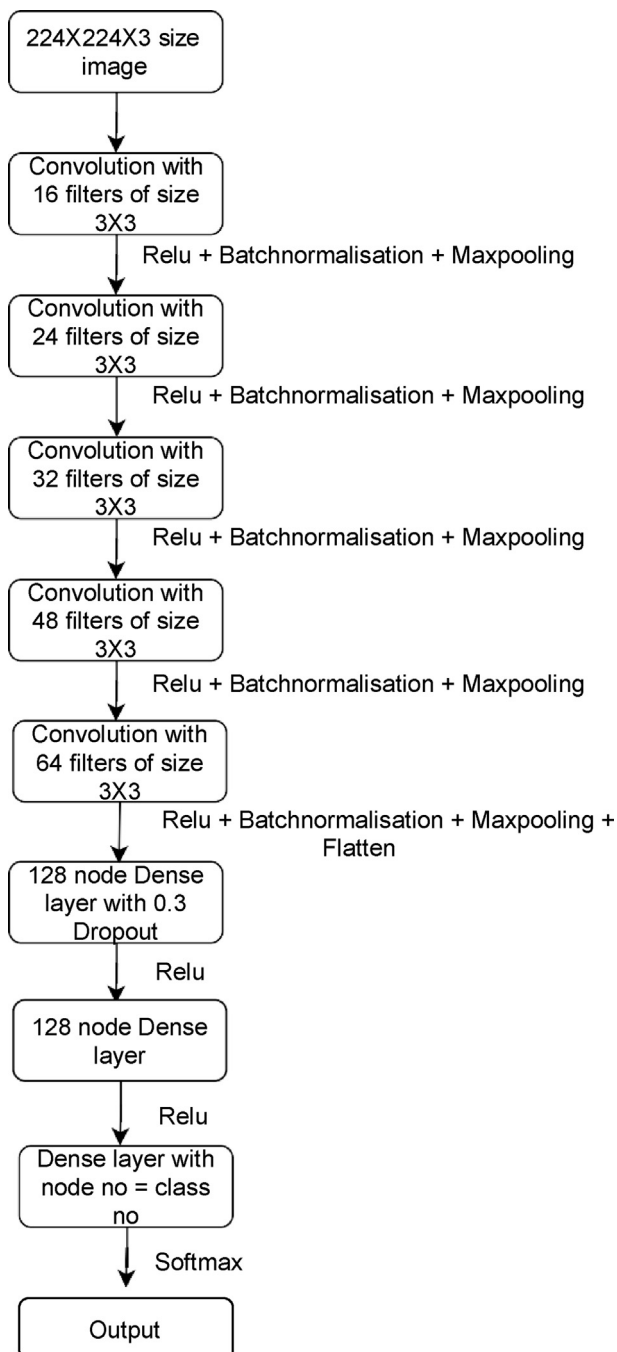

Fig. 3 — Simple CNN architecture.

10-fold cross-validation accuracy along with **standard deviation** have been used as model performance metric since the dataset used in this work does not have any major imbalance. **Categorical Crossentropy** has been used as loss function for all CNN architectures since this work deals with multi-class classification. All intermediate layers of the CNN architectures used in this work have **relu** as activation function while the activation function used in the last layer is **softmax**. The hyperparameters used are as follows: dropout rate of 0.3, learning rate of 0.0001, mini batch size of 64 and number of epochs 100. These values have been obtained through hyperparameter tuning using 10-fold cross-validation. Adaptive Moment Estimation (Adam) optimiser has been used for updating the model weights.

All the images have been resized to the default image size of each architecture before working with that architecture. For example, InceptionV3 requires $299 \times 299 \times 3$ pixel size image while VGG16 requires image of pixel size $224 \times 224 \times 3$. Random rotation from $-15°$ to $15°$, rotations of multiple of $90°$ at random, random distortion, shear transformation, vertical flip, horizontal flip, skewing and intensity transformation have been used as part of the data augmentation process. Every augmented image is the result of a particular subset of all these transformations, where rotation type transformations have been assigned high probability. It is because CNN models in general are not rotation invariant. In this way, 10 augmented images from every original image have been created. Random choice of the subset of the transformations helps augment an original image in a heterogeneous way.

A remote Red Hat Enterprise Linux server of RMIT University has been used for carrying out the experiments. The configuration of the server includes 56 CPUs, 503 GB RAM, 1 petabyte of user specific storage and two NVIDIA Tesla P100-PCIE GPUs each of 16 GB.

Table 4 — Quantitative performance of different state-of-the-art CNN architectures obtained from 10 fold cross validation (best accuracy of each architecture has been mentioned in bold character).

CNN Architecture	Training Method Used	Mean Validation Accuracy	Standard Deviation
VGG16	Baseline training	89.19%	10.28
	Transfer Learning	86.52%	5.37
	Fine Tuning	**97.12%**	2.23
InceptionV3	Baseline training	91.17%	3.96
	Transfer Learning	72.09%	7.96
	Fine Tuning	**96.37%**	3.9
MobileNetv2	Baseline training	78.84%	7.38
	Transfer Learning	77.52%	8.56
	Fine Tuning	**96.12%**	3.08
NasNet Mobile	Baseline training	79.98%	6.96
	Transfer Learning	78.21%	8.09
	Fine Tuning	**96.95%**	3.35
SqueezeNet v1.1	Baseline training	74.88%	8.18
	Transfer Learning	42.76%	9.12
	Fine Tuning	**92.5%**	3.75
Simple CNN	Two Stage Training	**94.33%**	0.96

2.3. *Proposed simple CNN model*

Apart from adapting state-of-the-art CNN models, a memory efficient two-stage small CNN architecture, namely **Simple CNN** shown in Fig. 3 has been constructed from scratch inspired by the sequential nature of VGG16. Fine tuned VGG16 provides excellent result on rice dataset. This **Simple CNN** architecture has only 0.8 million parameters compared to 138 million parameters of VGG16. All five of the state-of-the-art CNN architectures trained and tested in this work have shown the best result when fine tuning has been used (see Section 3). Two stage training is inspired from fine tuning. In **stage one**, the entire image dataset of nine classes are divided

into 17 classes by keeping all intra-class variations in separate classes. These variations have been shown in detail in Table 2. For example, **others** class is divided into three separate classes. Thus, the model is trained with this 17 class dataset. As a result, the final dense layer of the model has 17 nodes with softmax activation function. In **stage two**, the original dataset of nine classes is used. All layer weights of simple CNN architecture obtained from stage one are kept intact except for the topmost layer. This dense layer consisting of 17 nodes is replaced with a dense layer consisting of nine nodes with softmax activation function. Such measures are taken, because stage two training data are divided into the nine original classes. Now all the layers of the **Simple CNN** architecture are trained using this

BLB: Bacterial leaf blight
BPH: Brown plant hopper
BS: Brown spot
FS: False smut
Oth: Others
His: Hispa
NB: Neck blast
SBR: Sheath blight and/ or sheath rot
Stm: Stemborer

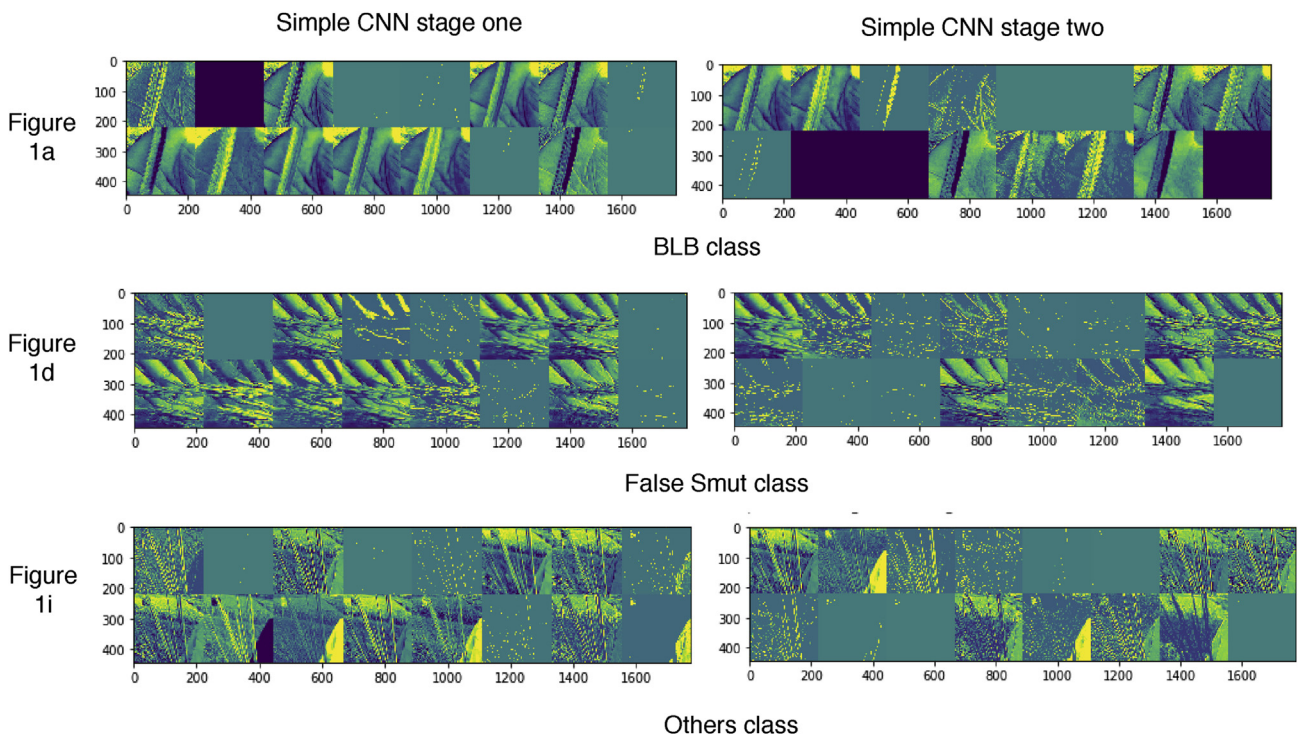**Fig. 4 — Confusion matrix generated using simple CNN on entire dataset.**

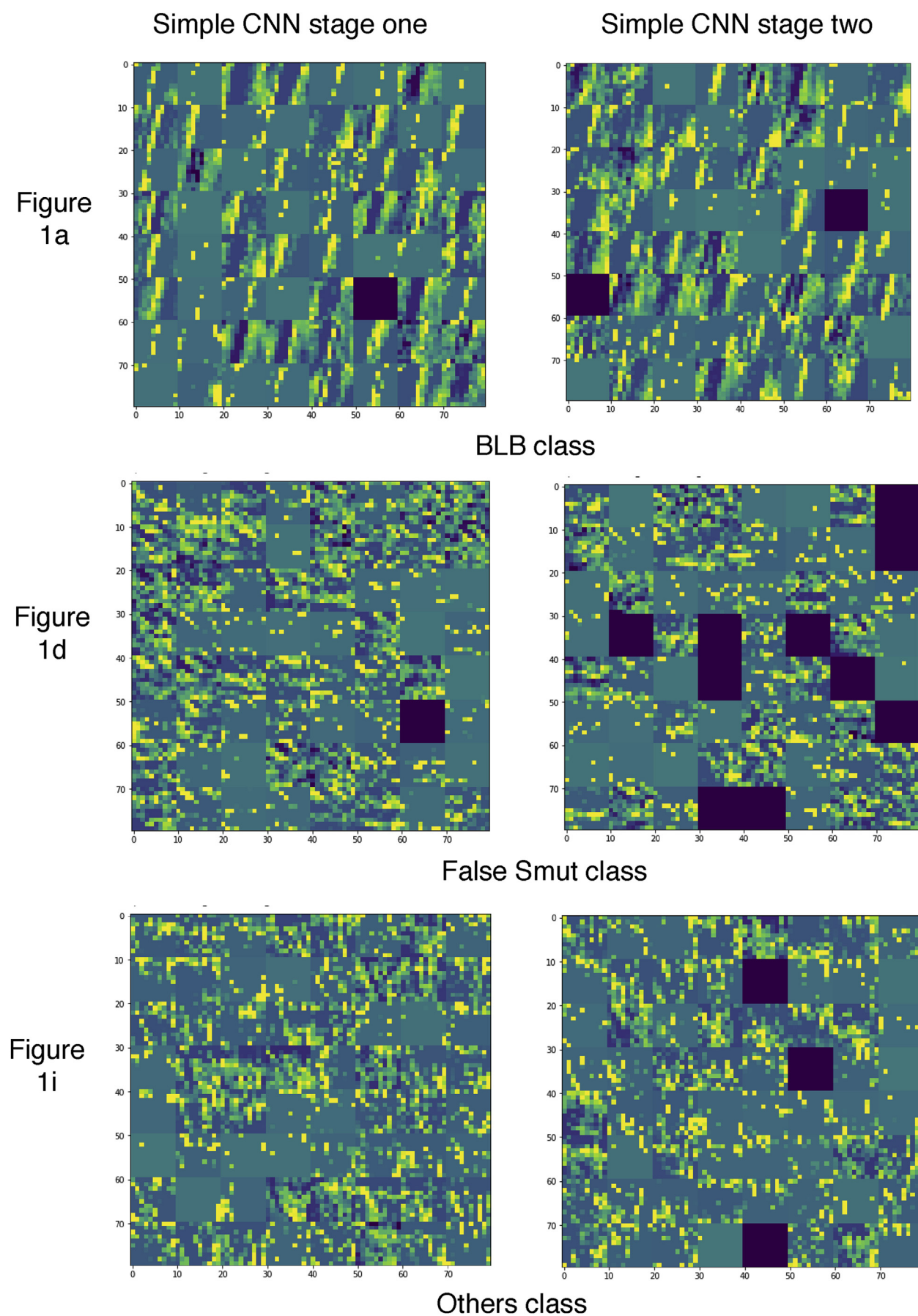**Fig. 5 — First convolution layer output of simple CNN.**

Fig. 6 — Last convolution layer output of simple CNN.

nine class dataset which are initialised with the pre-trained weights obtained from stage one training. Experiments show the effectiveness of applying this method.

3. Results and discussion

Experimental results obtained from 10-fold cross-validation for the five state-of-the-art CNN architectures along with Simple CNN have been shown in Table 4. Transfer learning gives the worst result in all five of the models. For the smallest architecture SqueezeNet, it is below 50%. Rice disease and pest images are different from images of ImageNet dataset. Hence, the freezing of convolution layer weights disrupts learning of the CNN architectures. Although baseline training, also known as training from scratch, does better than transfer learning, the results are still not satisfactory. For the three small models, the accuracy is less than 80%. The standard deviation of validation accuracy is also large which denotes low precision. This shows that the models are not able to learn the distinguishing features of the classes when trained from randomly initialised weights. More training data may solve this problem. Fine tuning gives the best result in all cases. It also ensures high precision (lowest standard deviation). It means that for the state-of-the-art CNN architectures to achieve good accuracy on rice dataset, training on the large ImageNet dataset is necessary prior to training on the rice dataset. Fine-tuned VGG16 achieves the best accuracy of 97.12%. The **Simple CNN** architecture utilising two stage training achieves comparable accuracy and the highest precision without any prior training on ImageNet dataset. Rather, this model is trained from scratch.

From Table 3, it is evident that the Simple CNN model has a small number of parameters even when compared to small state-of-the-art CNN architectures such as MobileNet and NasNet Mobile. The number of parameters of SqueezeNet (the smallest of the five state-of-the-art CNN architectures in terms of parameter number) is comparable to the parameter number of the Simple CNN. Sequential models like VGG16 need depth in order to achieve good performance, hence they have large numbers of parameters. Although Simple CNN is a sequential model with a low number of parameters, its high accuracy (comparable to the other state-of-the-art CNN architectures) proves the effectiveness of two stage training. Future research should aim at building miniature version of memory efficient non-sequential state-of-the-art CNN architectures such as InceptionV3, DenseNet and Xception. These architectures should be able to achieve similar excellent results with even smaller number of parameters.

A major limitation of two stage training is that the entire dataset has to be divided manually into symptom classes. In a large dataset, detecting all the major intra-class variations is a labour intensive process. There is a great chance of missing some symptom variations. Minor variety within a particular class maybe misinterpreted as separate symptoms. One possible solution could be to use high dimensional clustering algorithms on each class-specific image set separately in order to automate this process of identifying intra-class variations.

The confusion matrix generated from the application of Simple CNN on the entire dataset (training and validation set combined) has been shown in Fig. 4. 4.3% of the False Smut images existing in the dataset have been misclassified, which is the highest among all present classes of this work. False Smut symptoms cover a small portion of the entire image (captured with heterogeneous backgrounds) compared to other existing pest and disease images.

The first convolution layer outputs of Simple CNN have been shown in Fig. 5. The three rows from top to bottom represent output for Fig. 1a, d and i respectively, while the left and right columns represent output for stage one and stage two of the Simple CNN model respectively. Each of the six images contains 16 two dimensional mini images of size 222×222 (first convolution layer outputs a matrix of size $222 \times 222 \times 16$). The last convolution layer outputs of Simple CNN have been shown in Fig. 6 in a similar setting. Each of the six images of Fig. 6 contains 64 two dimensional mini images of size 10×10 (last convolution layer outputs a matrix of size $10 \times 10 \times 64$). The first layer maintains the regional features of the input image, although some of the filters are blank (not activated). The activations retain almost all of the information present in the input image. The last convolution layer outputs are visually less understandable. This representation depicts less information about the visual contents of the input image. Rather this layer attempts to present information related to the class of the image. The intermediate outputs for different classes are visually different for different classes. An interesting aspect can be observed in Fig. 6. The last convolution layer output for stage one model carries considerably fewer blank two dimensional mini images than does the stage two model. This shows the capability of the stage two model in terms of learning with less features. This helps Simple CNN achieve good accuracy and high precision after stage two training.

4. Conclusion

This work has the following contributions:

- A dataset of rice diseases and pests consisting of 1426 images has been collected in real life scenario which cover eight classes of rice disease and pest. This dataset is expected to facilitate further research on rice diseases and pests. The dataset is available here (https://drive.google.com/open?id=1ewBesJcguriVTX8sRJseCDbXAF_T4akK). The details of the dataset have been described in Subsection 2.1.
- Three different training methods have been implemented on two state-of-the-art large CNN architectures and three state-of-the-art small CNN architectures (targeted towards mobile applications) on the rice dataset. Fine tuning from pre-trained ImageNet weight always provided the best result for all five architectures.
- A new concept of two stage training derived from the concept of fine tuning has been introduced which enables proposed **Simple CNN** architecture of this work to perform well in real life scenario.

In future, incorporating location, weather and soil data along with the image of the diseased part of the plant can be investigated for a comprehensive and automated plant disease detection mechanism. Segmentation or object detection-based algorithm can be implemented with a view to detecting and classifying rice diseases and pests more effectively in the presence of heterogeneous background.

Declaration of Competing Interest

None declared.

Acknowledgments

REFERENCES

Atole, R. R., & Park, D. (2018). A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. *International Journal of Advanced Computer Science and Applications, 9*(1), 6770.

Babu, M. P., & Rao, B. S. (2007). Leaves recognition using back propagation neural network- advice for pest and disease control on crops. *IndiaKisan. Net: Expert Advisory System*.

Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture, 153*, 46—53.

Barbedo, J. G. A. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering, 180*, 96—107.

Bhagawati, R., Bhagawati, K., Singh, A., Nongthombam, R., Sarmah, R., & Bhagawati, G. (2015). Artificial neural network assisted weather based plant disease forecasting system. *International Journal on Recent and Innovation Trends in Computing and Communication, 3*(6), 4168—4173.

Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence, 31*(4), 299—315.

Calpe, C. (2002). Rice in world trade, part ii. status of the world rice market. In *Proceedings of the 20 th session of the international rice commission*.

Coelli, T., Rahman, S., & Thirtle, C. (2002). Technical, allocative, cost and scale efficiencies in Bangladesh rice cultivation: A non-parametric approach. *Journal of Agricultural Economics, 53*(3), 607—626.

Cruz, A. C., Luvisi, A., De Bellis, L., & Ampatzidis, Y. (2017). X-fido: An effective application for detecting olive quick decline syndrome with deep learning and data fusion. *Frontiers of Plant Science, 8*, 1741.

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., et al. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology, 107*(11), 1426—1432.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture, 145*, 311—318.

Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors, 17*(9), 2022.

Karmokar, B. C., Ullah, M. S., Siddiquee, M. K., & Alam, K. M. R. (2015). Tea leaf diseases recognition using neural network ensemble. *International Journal of Computer Application, 114*(17).

Liu, B., Zhang, Y., He, D., & Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry, 10*(1), 11.

Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing, 267*, 378—384.

Mahmud, H., Hossain, I., & Ahmad, M. (2016). *Efficacy of bau-biofungicide on brown spot and bacterial leaf blight disease and vigour index of rice*.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers of Plant Science, 7*.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (p. 19).

Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Computational Intelligence and Neuroscience*.