Article:

Bijoy, M.H., Hasan, N., Biswas, M. et al. (5 more authors) (2024) Towards sustainable agriculture: a novel approach for rice leaf disease detection using dCNN and enhanced dataset. IEEE Access, 12. pp. 34174-34191. ISSN 2169-3536

https://doi.org/10.1109/access.2024.3371511

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3371511

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Towards Sustainable Agriculture: A Novel Approach for Rice Leaf Disease Detection using dCNN and Enhanced Dataset

MEHEDI HASAN BIJOY[1], NIROB HASAN[2], MITHUN BISWAS[3], SUVODEEP MAZUMDAR[4], ANDREA JIMENEZ[4], FAISAL AHMED[5], MIRZA RASHEDUZZAMAN[6], AND SIFAT MOMEN[2]
[1]Department of Information and Communications Engineering, Aalto University, Otakaari 24, 02150 Espoo, Finland (e-mail: mehedi.bijoy@aalto.fi)
[2]Department of Electrical and Computer Engineering, North South University, Plot 15, Block B, Bashundhara, Dhaka-1229, Bangladesh (e-mail: thenirobhasan@gmail.com, sifat.momen@northsouth.edu)
[3]FS Solution Co. Ltd., South Korea, (e-mail: mranjon@gmail.com)
[4]Information School, University of Sheffield, The Wave, 2 Whitham Road, Sheffield, S10 2AH, UK (e-mail: s.mazumdar@sheffield.ac.uk, a.jimenez@sheffield.ac.uk)
[5]IQVIA, Dhaka, Bangladesh (e-mail: faisal.aop@gmail.com)
[6]Department of Electrical and Electronic Engineering, University of Liberal Arts Bangladesh (e-mail: mirza.rasheduzzaman@ulab.edu.bd)

Corresponding authors: Suvodeep Mazumdar (s.mazumdar@sheffield.ac.uk), Sifat Momen (sifat.momen@northsouth.edu).

“This work was supported by X/161099 GCRF QR Project.”

ABSTRACT Rice is one of the foremost food grains that dispenses sustenance to about half of the world’s population. It is cultivated all over the world. The leaf disease detection of this crop is one of the chronic agricultural obstacles that farmers and planting experts have been struggling with for a long time. As a result of the leaf diseases, producing the amount of rice required to feed the world’s rising population has become very challenging. Hence, automatically detecting rice leaf diseases is an inevitable task to increase productivity. Numerous deep learning based methods have been proposed for rice leaf disease detection, which we found rather inefficient considering the size of the models. In this article, we introduce a lightweight deep Convolutional Neural Network (dCNN) based method for rice leaf disease detection, that outperforms contemporary state-of-the-art methods and showcases competitive performance against 21 established benchmark architectures, including AlexNet, MobileNet, ResNet50, DenseNet121, ResNeXt50, ShuffleNet, ConvNext, EfficientNet, GogoleNet, SwinTransformer, VisionTransformer, and MaxVit, to name a few, with significantly lower trainable parameters. Notably, our method achieves an accuracy score of 99.81%, a precision score of 0.99828, a recall score of 0.99826, and an f1-score of 0.99827. Moreover, we enhance the rice leaf disease dataset by merging two existing datasets and supplemented them with an additional 95 manually annotated images gathered from publicly available sources on the internet. We also develop a comprehensive crop health monitoring system for farmers, and develop an open API for the automatic annotation of new instances, benefiting the research community at large.

INDEX TERMS deep learning; deep convolutional neural network; rice leaf disease detection; leaf disease classification

I. INTRODUCTION

Rice serves as a vital staple, providing sustenance to billions of individuals across the globe. Remarkably, it is cultivated in more than three-fifths, precisely 61.54%, of countries worldwide [1]. However, the confluence of a burgeoning population and dwindling arable land has led to an escalating reliance on rice, exacerbating the issue of gradual food scarcity. Moreover, rice production faces persistent challenges, including the annual onslaught of various maladies, particularly the detrimental rice leaf diseases. These afflictions disrupt the natural growth, structure, and coloration of rice leaves, arising from internal abnormalities often caused by fungi or viruses. Bangladesh, for instance, has a predominantly agrarian economy where rice is the staple food for the majority of the population [2]. Positioned as the largest delta globally, Bangladesh is bestowed with rivers cascading from the Himalayas. Despite having fertile soil and advantageous seasons, the country has been unable to fully harness its agricultural potential due to limited access to cutting-edge technology [3] as well as the hesitancy to adopt new technologies due to wide number of factors including misconception, confusion and uncertainty [4]. Currently, it produces approximately 35 million metric tons of rice annually, falling short of meeting the demands of its growing population [5]. The underlying cause of this shortfall can be attributed to rice leaf diseases, unequivocally responsible for diminished crop yield [6]. To enhance productivity, regular crop health monitoring and proper care are essential. But manual detection of rice leaf diseases is a time-consuming, labor-intensive, and costly process, rendering it impractical [7].

Earlier efforts in rice leaf disease detection focused on machine learning algorithms-based, neural network-based, and hybrid methods. The conventional machine learning algorithms, while effective in some cases, often demonstrated limited performance as they heavily relied on manual, hand-crafted feature engineering. This reliance on human-crafted features not only posed challenges in capturing complex patterns but also led to increased development time and effort [8], [9]. The recent emergence of neural networks has introduced CNN-based approaches, which incorporate transfer learning with pre-trained models and customized architectures. Unfortunately, these approaches often entail high asymptotic complexity due to their extensive trainable parameter size, making them challenging to deploy on resource-constrained devices and limiting their practicality in real-time applications. Our extensive study found that the development of an effective rice leaf disease detection method is hindered by two fundamental barriers: a large trainable parameter size of the model and a small dataset size. To address these limitations, we propose an advanced end-to-end method which aims to automatically and reliably detect rice leaf diseases, providing valuable assistance to farmers and contributing to the agricultural development of the country. Specifically, we present a lightweight deep Convolutional Neural Network (dCNN) architecture and an enhanced dataset.

The objective of this study is to develop and deploy deep learning models capable of accurately predicting rice leaf diseases. In particular, this article delves into the detection of the five frequently occurring rice leaf diseases, including bacterial leaf blight, blast, brown spot, sheath blight, and tungro, using a lightweight dCNN model. To aid farmers in effortlessly monitoring crop health, we have developed a comprehensive crop health monitoring system comprising a user-friendly website and an Android application. Moreover, we introduce an open API and enhance the rice leaf disease dataset, making it a valuable resource for the research community. The dataset is enhanced by collecting data from the internet and manually annotating them with the assistance of domain experts, resulting in a broader range of rice leaf disease variations. As for the API, which accepts input images and returns disease labels along with insights into the disease's etiology and suggested subsequent actions, it facilitates automatic annotation of new instances, benefiting the research community at large.

The key contributions of this article are summarized below:

- We propose a lightweight dCNN architecture for rice leaf disease detection that outperforms several contemporary state-of-the-art methods. For instance, it surpasses [10], [11], and [12] with 16, 811, and 152 times fewer parameters. It also demonstrates superior performance compared to [13] and [14].
- We compare the performance of our proposed method with 21 benchmark architectures, comprising 16 convolution-based and five transformer-based methods. It outperforms the majority of these methods, achieving competitive performance with the remaining ones, where the differences in performance are negligible, with a much lower trainable parameter size.
- We conducted extensive experiments including a wide range of scenarios and varying environmental circumstances. These scenarios included images with natural backgrounds, diverse camera angles generated through random rotations, varying distances captured using zoom-out procedures, and alterations in image quality using both downsampling and upsampling approaches. Furthermore, the model was evaluated using datasets obtained from different geographical locations including Indonesia, China, and Taiwan.
- We enhance the rice leaf disease datasets in [15] and [10] by collecting data, at least 95 unique RGB images, from the internet and manually annotating them by domain experts to ensure accurate and high-quality labeling.
- We develop a comprehensive crop health monitoring system for farmers, encompassing a user-friendly website, an intuitive Android app, and an accessible open API, with the aim of assisting both farmers and the research community.

The remainder of this article is structured as follows. Section II presents a comprehensive literature review on the detection of rice leaf diseases. Section III of the paper examines the limitations of the existing dataset found in the literature and proposes data preparation procedures to address these shortcomings. In Section IV, we delve into the challenges encountered, elaborate on our network architectures, and outline the proposed methodology. The experimental results are expounded upon in Section V, encompassing comparisons with benchmark architectures and state-of-the-art methods. Further details regarding model deployment can be found in Sub-section V-H. Next, the advantages and drawbacks of the proposed method is discussed in Section VI. Finally, Section VII concludes our study and outlines future possibilities for research in this domain.

II. RELATED WORKS

A wide range of approaches has been proposed for detecting rice leaf diseases. These methodologies primarily fall into three categories: machine learning algorithms-based, neural network-based, and a fusion of both. The performance of the methods deliberately relies on the dataset and discerning strategy employed for feature extraction.

Conventional machine learning algorithms based methods have been proposed in [5], [13], [14], [16], [17]. For instance, [14] and [13] use XGBoost and Support Vector Machine (SVM), respectively, whereas [16] and [17] utilize random forest classifier to detect rice leaf diseases. The performance of these methods is not up to the mark and solely relies on feature engineering. Due to the advent of deep learning, which tends to outperform machine learning algorithms, most of the studies propose a Convolutional Neural Network (CNN) based approach to address the problem. These studies can further be classified into transfer learning based [18]–[21] and custom model based [22]–[26] approaches. Among transfer learning based methods, [12], [11], [18], [20], and [19], [21] employ DenseNet, VGG, and Insception-ResNet pre-trained on ImageNet. Recently, [18] presents a two-stage CNN architecture by adopting and fine-tuning VGG16 and InceptionV3. Likewise, [20] and [27] modify VGG16 and ResNet18, respectively, to reduce the model parameter size. Among custom model based approaches, [23] proposes a MobileNet like architecture by incorporating attention mechanism and name it ADSNN-BO. A custom CNN architecture has also been proposed in [24], [26], [27]. Lately, [27] further utilizes Generative Adversarial Network (GAN) to generate synthetic data. However, [28] tackled the same problem by utilizing edge computing concepts. A few studies [29], [30] propose such hybrid approaches which are an amalgamation of CNN and machine learning algorithms. In [29], they perform two approaches including CNN with fully connected layers and CNN with SVM. They use a CNN architecture as a baseline which is identical to LeNet. The first approach includes the base CNN followed by a SVM classifier. The second approach consists of the base CNN and two additional fully connected layers. In contrast, [5], [14] propose such approaches that amalgamate both CNN and machine learning algorithms as well. Among these methods, [14] removes the background of an image based on the saturation threshold. Afterward, disease-affected regions are segmented using the threshold mask on the hue plane of the HSV images. Finally, an extreme gradient boosting decision tree ensemble(XGBoost) is used for classifying the diseases with the Logistic loss function. In another work [5], the background is removed using a segmentation technique done by Otsu's threshold method that determines the optimal value for the global threshold. Then, they use the extracted features from CNN for classification using SVM. They consider three kernel functions namely linear, polynomial and radial basis function (RBF), and report the highest performance using SVM with polynomial Kernel and HOG.

Neural network based approaches tend to outperform statistical and typical machine learning algorithm based approaches [5], [13], [14]. Azim and colleagues [14] remove the leaf background depending on the color of the leaf, e.g., they remove the green part of the leaves and take only the affected portion of the image. In real life scenarios, the color of the leaves does not always remain the same. In that case, they may not be able to extract the features properly in the case of dark or light green. "Histogram of Oriented Gradients (HOG)" is used to describe features in [5]. They use SVM with a polynomial kernel function as the dataset is tiny, but they do not consider all the features of the images. There might be some noisy leaf images in the dataset where SVM can not perform well. Another study [30] uses ResNet50 and SVM in a dataset that consists of 5932 images of four rice leaf diseases. The SVM classifier is not suitable for large datasets. At the same time, ResNet50 requires extensive data which is creating an anomaly in the performance. Recently, [11] proposes a CNN architecture that uses the pretrained VGG16 backbone and transfer learning. VGG is a huge model and takes more time to process an image than other models like ResNet. They do not consider other pretrained models. [13] uses color features to explore 14 distinct color spaces and extract four from each color channel. Although the color of the leaves is not always the same, it varies depending on the lighting. Additionally, they have used an SVM classifier, which does not perform well with a large amount of data. Our proposed method resolves the limitations found in the literatures by considering all the local features of an image with a much smaller model parameter size than existing methods.

III. DATASET PREPARATION

A. OVERVIEW OF EXISTING DATASETS

We collected several existing rice leaf datasets from various online sources [10], [15], [31]–[33]. However, after a thorough scrutinization, we found that the field of rice leaf disease detection faces a significant challenge due to the lack of a publicly available large enough dataset. It was also observed that many publicly accessible datasets (e.g. [31]) lack reliability due to the inclusion of identical or augmented versions of images from the train set to the test set. This phenomenon results in an artificial inflation of performance metrics when evaluating the model on the test set. Such models are unlikely to meet the expected performance when applied to real-world data. The lack of extensive publicly accessible datasets presents a major challenge in this field of study, which is worsened by the arduous task of collecting leaf data with subtle disease variations, diverse environmental conditions, and the tedious task of accurately annotating samples.

To tackle this issue, we merged two datasets obtained from sources [10] and [15], and combined them with 95 quality images that we collected from various sources of internet. The datasets [10] and [15] were chosen based on their superior quality. There were a total of 3876 (augmented) and 120 raw images in [10] and [15] respectively. We selected 80 of 120 images from [15] based on the image classes that we are interested in. The 95 quality images that we collected from the internet which were annotated manually. The images from the internet and the 80 images from [15] were combined and then augmented to form a total of 1409 augmented images. This along with the 3876 images from [10] form the image dataset of 5285 images. This curated dataset of 5285 images were used in this work. Considering the crucial role a large dataset plays in achieving notable performance from neural networks,

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3371511

our enhanced version holds immense potential in advancing the field. Hence, our curated set of 95 distinct RGB images, meticulously collected and annotated, serves as a substantial enhancement to the existing dataset.

B. DATA ACCUMULATION

As our extensive study found that the paucity of a publicly available large enough dataset is the main barrier towards the development of efficient rice leaf disease detection, we amalgamate two datasets [10], [15] and further enhance them with images procured from the internet. The manual annotation technique employed in this study encompasses the following components: The symptoms of diseases, along with their corresponding visual representations, were acquired from BARI (Bangladesh Agricultural Research Institute), which is the largest agricultural research institute in Bangladesh. Subsequently, the collected samples were meticulously examined. Additionally, internet data that exhibited a clear resemblance to the visual representation of specific diseases were incorporated and annotated. Annotation was carried out by three members of the group who studied the visual representation and independently classified the diseases. Only the images that received unanimous agreement from all three members were included. Inclusion of the additional images from the internet resulted in the increase in the number of diseased classes.

Firstly, we collected data from the UCI Machine Learning Repository [15], which contains 120 images distributed among three classes: bacterial leaf blight, brown spot, and leaf smut, each containing 40 images. Next, we acquired an additional 95 images from the internet, denoted as $I = \{I_1, I_2, ..., I_{95}\}$, introducing three new disease classes: blast, sheath blight, and tungro, which are very relevant to the rice leaf diseases encountered in Bangladesh. We combined these collected images with the manually annotated images from the UCI dataset, resulting in a total of five classes. The leaf smut class was omitted from the dataset [15] due to its significantly lower number of images compared to the other classes. Then, we performed data augmentation (which is detailed in the sub-section **??**) on the combined images, generating a set of 1409 images. Subsequently, we merged this augmented dataset with the one used in [10], which already contained 3876 augmented images. The resulted dataset comprises 5285 images, representing five disease classes: sheath blight, tungro, brown spot, blast, and bacterial leaf blight.

C. DATA AUGMENTATION

Convolutional Neural Network (CNN) models demand a substantial volume of training data to effectively discern underlying patterns and attain optimal performance during inference. In this context, image augmentation emerges as a pragmatic and widely adopted approach [34]–[37] to construct a resilient image classifier with limited training data. By augmenting the dataset through various transformations, it substantially increases the number of images, thereby bolstering the capability of deep learning models to achieve better performance.

A significant amount of synthetic data was therefore generated using conventional data augmentation techniques, encompassing eight distinct transformations, namely cropping, horizontal and vertical shifting, horizontal and vertical flipping, zooming in and out, and rotation. Each of these transformations play a crucial role in creating a unique representation of the original image. We ensure that augmented instances do not overlap across multiple sets, thereby eliminating any potential data fabrication issues. To achieve this, the augmentation process commences with cropping each instance of our combined dataset, preserving their spatial dimensions while resizing them to 240×240 pixels, ensuring uniformity in image size. Subsequently, horizontal and vertical shifts are applied, with a height and width shift range of 0.2, respectively. This leads to the random truncation of the image within the selected negative or positive range, effectively creating shifts both horizontally and vertically. Furthermore, the original instances are flipped horizontally and vertically with a probability score of 0.5. These horizontal and vertical flips produce unique images by transforming rows into columns and vice versa, thereby expanding the dataset's diversity. In addition, the rotation transformation is employed, randomly rotating the images clockwise within the range of 1 to 45 degrees randomly, which introduces further variability to the dataset. Lastly, we adopt the zoom in and out transformation with a range of 0.3, allowing us to alter the aspect ratio of the resultant instances, further enriching the dataset with varied representations.

By employing these data augmentation techniques, our dataset is substantially augmented, providing an extensive and diverse collection of instances for robust model training.

D. ENHANCED RICE LEAF DISEASE DATASET

The enhanced dataset contains 5593 images from 5 disease classes including sheath blight, tungro, brown spot, leaf smut, and bacterial leaf blight. A few sample instances of the dataset are shown in Figure 1. We split the dataset into train, validation, and test sets containing 3158, 1277, and 850 images respectively. The statistic of the dataset can be found in Table 1.

Class Name	**# Images**		
	Training	***Validation***	***Test***
Sheath Blight	371	221	149
Tungro	410	246	163
Brown Spot	936	282	187
Bacterial Blast	410	246	163
Bacterial Leaf Blight	1031	282	188
	Total = 3158	**Total = 1277**	**Total = 850**

TABLE 1. The statistics of our enhanced rice leaf disease dataset.

The validation and test sets are balanced, whereas the training set exhibits some degree of imbalance. Notably, the training set contains the highest number of images for

Bacterial Leaf Blight Bacterial Blast Brown Spot Sheath Blight Tungro

FIGURE 1. Sample Images from Rice Leaf Disease Dataset.

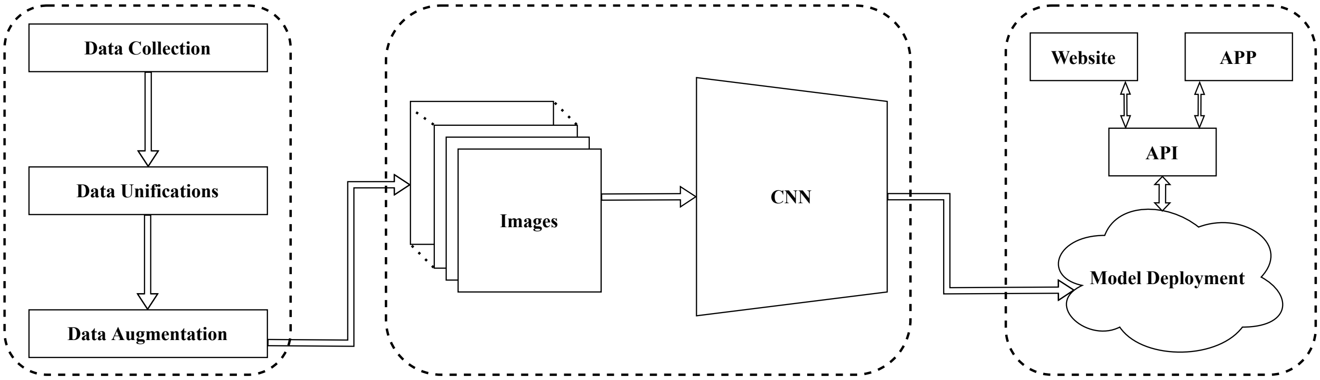


FIGURE 2. (Left) Image preprocessing begins by gathering image data for our project. It includes data unification and augmentation where we amalgamate multiple datasets and generate synthetic data. (Middle) Our proposed deep learning model which takes an image as input and classifies it into one of the disease categories based on the features of the image. (Right) The deployment of our model which includes an API, an android app, and a website.

bacterial leaf blight, whereas sheath blight has the fewest instances. Specifically, bacterial leaf blight and sheath blight represent approximately 33% and 11% of the total images in the training set, respectively. Similarly, tungro, brown spot, and bacterial blast make up roughly 13%, 30%, and 13% of the training set images, respectively.

IV. METHODOLOGY

A. OVERVIEW

The proposed method takes an image as input and classifies it into one of the disease categories based on the local features of the image. It begins by taking an image as input and resizing it according to the input image size of the model. Then it extracts the features of the image with the help of convolution and pooling layers. Finally, it uses the extracted features to classify the image. Figure 2 depicts the approach for rice leaf disease detection.

B. APPROACH

Training the model involves two major steps: forward propagation and backward propagation. Firstly, we initialize the weights of the model randomly. In the case of forward propagation (equation 1), we pass a batch of images through the model. The input data moves forward and generates predictions. The calculations in the neurons of the hidden layers and output layer are as follows.

$$a_n^{[l]} = f((W_{n1}^{[l]} \times a_1^{[l-1]}) + ... + (W_{nj}^{[l]} \times a_j^{[l-1]}) + b_i) \quad (1)$$

Where, l refers to the hidden layer, n refers to a neuron of a hidden layer, j refers to a neuron of previous hidden layer, W is the weight matrix, a is the output of a neuron, b is the bias and f is the activation function.

To update the weights in backward propagation, the loss is calculated using the prediction from forward propagation and the actual label. The cross-entropy loss function is employed because our extensive scrutiny revealed that the Adam optimizer and categorical cross-entropy loss function together yield the best performance. The loss is calculated as the negative logarithm of the softmax output for a specific class, along with the true label. The equation for calculating softmax probability is as follows:

$$\sigma = \frac{e^{z_i}}{\sum_{n=1}^{k} e^{z_k}}, k = 1, 2, ..., n \quad (2)$$

Softmax returns the probability of the input belonging to each of the classes. The probability value ranges between 0 to 1, and the sum of all probabilities equals 1. We then compute the cross-entropy loss, which is given in equation 3.

$$CE = -\sum_{c=1}^{n} t_i log(\sigma) \quad (3)$$

The cross-entropy is the product of the true label of a certain class and the negative logarithm of the softmax of its prediction. Hence, the t_i of the equation will be zero for all other classes, but it will be 1 for the true label or class.

We calculated the goodness of fit using prediction and actual label by employing equation 3. Based on the value of the loss, we updated the weights in backward propagation. Since we use mini batches, the weights of the model will be updated at each mini batch. The Adam optimizer is used to minimize the cost function which is an amalgamation of RMSprop and momentum. The final weight (equation 4) and bias (equation 5) updation formula is as follows.

$$W_t = W_{t-1} - \eta \frac{\bar{m}_t}{\sqrt{\bar{v}_t} + \epsilon} \tag{4}$$

$$B_t = B_{t-1} - \eta \frac{\bar{m}_t}{\sqrt{\bar{v}_t} + \epsilon} \tag{5}$$

where, W is model weights, B stands for bias, t is the current state, η is the step size, and $\bar{m}_t$ and $\bar{v}_t$ are bias corrected estimators for the first and second momentum estimator.

1) Shortcomings of Existing Architectures

We commence the experiment by utilizing established benchmark architectures, ensuring a robust foundation for our study. Our selection includes seven widely recognized architectures: AlexNet [38], MobileNetV2 [39], MobileNetV3 [40], ResNet50 [41], DenseNet121 [42], ResNeXt50 [43], and ShuffleNetV2 [44]. Each of these models is proficient at taking an image as input and accurately classifying it into its respective disease category. Their individual performances showcase high efficacy. The empirical outcomes of these models can be found in Table 3.

The vast majority of these well-known architectures perform admirably. However, the total number of trainable parameters in these models is massive. For example, ResNet50 contains 25.5M trainable parameters. As a result, deploying such models is costlier and impractical for oprational settings that introduce constraints such as low-resource devices or reduced bandwith as is typical in remote regions in the Global South. Therefore, we develop a tiny dCNN model for rice leaf disease detection that is efficient in terms of performance and practical for deployment. For instance, our rice leaf disease detection (RLDD) model is 150 times smaller than ResNet50, yet it performs similarly. The extensive experiments support the validity of the proposed method, which is successful and efficient in classifying rice leaf diseases.

2) Proposed Architecture

Our proposed model consists of mainly two parts – one part of the model is used for feature extraction while the other part is used for classification. The feature extraction component involves convolution and pooling layers to effectively capture relevant patterns in the data. On the other hand, the classification component consists of dense layers, also known as fully connected layers, which aid in making accurate predictions based on the extracted features. Moreover, Dropout layer is used to avoid overfitting of the model. The proposed model is illustrated in figure 3.

A non-linear mapping $f(x, \theta)$ is exploited by the model. In all of the hidden layers of the model, we used Rectified Linear Unit (ReLU), $R(z) = max(0, z)$. But in the output layer we used Softmax activation function which is $\sigma = e^{z_i} / \sum_{n=1}^{k} e^{z_k}$, for $j = 1, ..., k$, to produce probabilistic predictions. To mitigate the risk of overfitting, a dropout ratio of 10% is consistently applied across the model. The details of all layers of the model are available in the table 2.

Layer Name	Parameters	Output Size
Conv2D 1	F: 24, KS: 3x3, S: 1x1	BSx222x222x24
MaxPooling 1	Pool Size: 2x2, S: 2x2	BSx111x111x24
Conv2D 2	F: 32, KS: 3x3, S: 1x1	BSx109x109x32
MaxPooling 2	Pool Size: 2x2, S: 2x2	BSx54x54x32
Conv2D 3	F: 40, KS: 4x4, S: 1x1	BSx51x51x 40
MaxPooling 3	Pool Size: 2x2, S: 2x2	BSx25x25x40
Conv2D 4	F: 48, KS: 4x4, S: 1x1	BSx22x22x48
MaxPooling 4	Pool Size: 2x2, S: 2x2	BSx11x11x48
Conv2D 5	F: 56, KS: 4x4, S: 1x1	BSx8x8x56
MaxPooling 5	Pool Size: 2x2, S: 2x2	BSx4x4x56
Conv2D 6	F: 64, KS: 4x4, S: 1x1	BSx1x1x64
Fully Connected 1	Units: 72	BSx72
Fully Connected 2	Units: 64	BSx64
Output	Units: No. of Classes	BSxNo. Of Classes

TABLE 2. The details of our proposed Rice Leaf Disease Detection architecture where F, KS, S, and BS denote filters, kernel size, strides, and batch size, respectively.

Our proposed deep learning model is super lightweight, comprising only 0.17M parameters. This is remarkably small compared to other models such as AlexNet, ResNet50, DenseNet121, etc. Despite its significantly smaller size compared to the models mentioned in sub-section IV-B1, our model outperforms most of them. For instance, it is 150 and 364 times smaller than ResNet50 and AlexNet, respectively, while still delivering competitive performance with ResNet50 and surpassing AlexNet. Furthermore, it outperforms models [39] and [43], even though it has 20.5 and 147 times fewer parameters, respectively. This remarkable performance makes our lightweight model an excellent choice for efficient and effective rice leaf disease detection.

V. EXPERIMENTAL ANALYSIS

A. HYPERPARAMETER TUNING

We fine-tune the model's hyperparameters, resulting in further performance improvement. We conduct empirical experiments with different dropout ratios, activation functions, optimizers, batch sizes, hidden layers, and loss functions. Through this experimentation, we determine that a specific combination, as shown in Figure 4, yields the best result.

Regarding the size of the model, we explore the impact of adding more convolution layers. We observe that increasing the convolution layers does not improve the model's performance but does increase the total number of trainable parameters. Conversely, reducing the number of convolution layers dramatically decreases the model's performance. Additionally, we experiment with dropout ratios of 10%, 20%, and 40% and discover that as we increase the dropout ratio,

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3371511

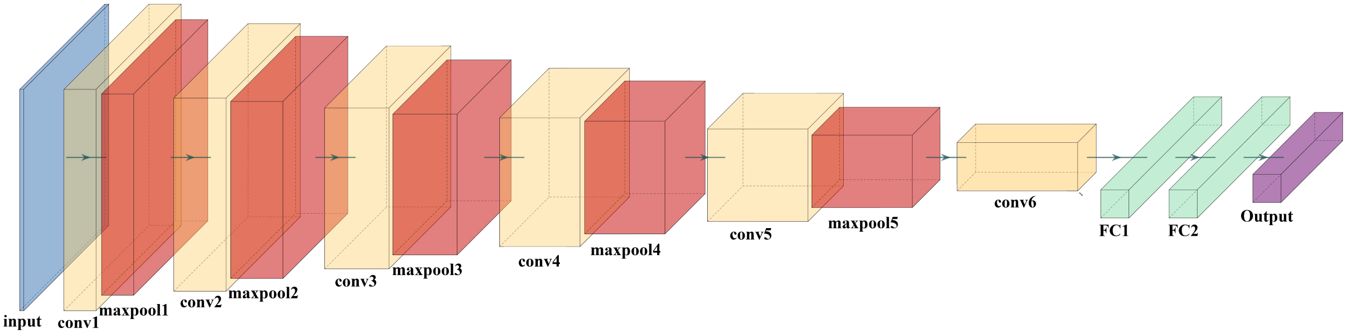
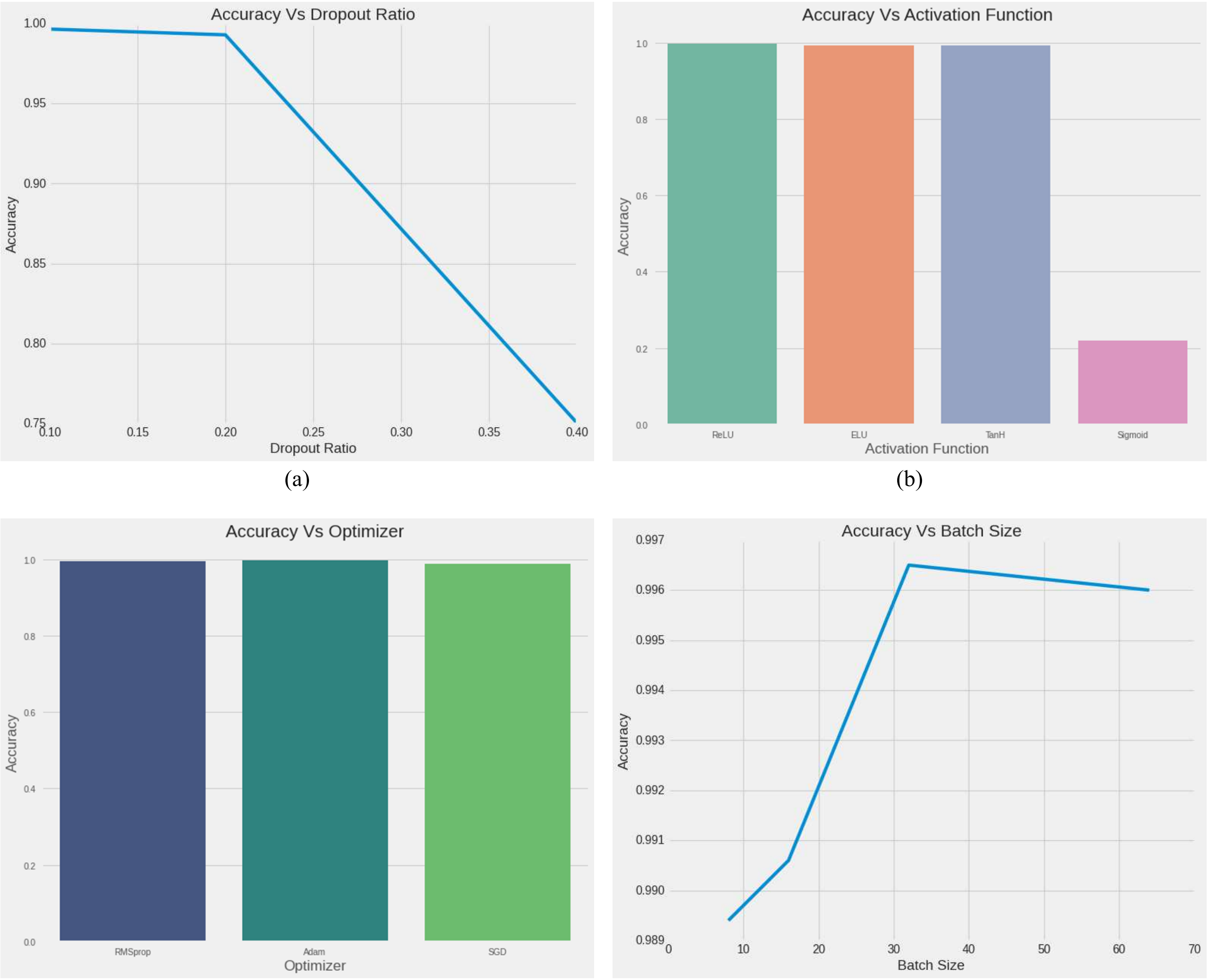


FIGURE 3. **Our proposed dCNN backbone for rice leaf disease detection. It consists of six convolution layers, five maxpooling layers, and two fully connected layers.**



(a) (b)

(c) (d)

FIGURE 4. **The experimental outcomes of our proposed rice leaf disease detection model for using different dropout ratios, activation functions, optimizers, and batch sizes. Sub-figures (a), (b), (c), and (d) illustrates the empirical outcomes of different dropout ratios, activation functions, optimizers, and batch sizes.**

the model's accuracy gradually decreases. Figure 4(a) depicts the accuracy vs. dropout ratio graph, illustrating the changes in accuracy for different dropout ratios. In terms of activation functions, we utilize ReLU, ELU, TanH, and Sigmoid and find that ReLU delivers the best performance, while Sigmoid yields the lowest results. Figure 4(b) presents the empirical outcomes of different activation functions. For optimizers, we experiment with RMSprop, Adam, and SGD. Although all three optimizers perform comparably (Figure 4(c)), Adam exhibits slightly better and faster performance than RMSprop and SGD. Figure 4(d) displays the accuracy vs. batch size graph, highlighting the relationship between accuracy and batch size. It shows that the model's accuracy increases until the batch size reaches 32 and decreases afterward. Consequently, in the final training, we used ReLU activation function, Adam optimizer, a dropout ratio of 10%, a batch size of 32, and a learning rate of 0.001. Regarding the learning rate, we experimented with adding a decay rate, particularly cosine decay, which ultimately reduced the overall performance.

B. EVALUATION METRICS

We employ performance measures including Accuracy, Precision, Recall and F1 Score to measure the performance of our method. The confusion matrix is utilized to determine the accuracy, precision, recall and f1-score.

- **Confusion Matrix:** It is an $N \times N$ matrix which is used for evaluating the performance of a classification model where N is the number of classes. It compares the actual labels to the model's predictions and gives us a holistic view regarding the performance of our model along with the type of errors it is making. The $N \times N$ matrix consists of two kind of values: positive and negative. The columns and rows of the matrix represent the actual and predicted values respectively. The four most important terminologies in a confusion matrix are True Positive (TP), True Negative (TN), False Positive (FP) or Type-I Error, and False Negative (FN) or Type-II Error.
  True Positive (TP): The model predicted positive and it's true, which means the predicted value matches the actual value.
  True Negative (TN): The model predicted negative and it's true, which means the predicted value matched the actual value.
  False Positive (FP) or Type-I Error: The model predicted positive but it's false, which means the predicted value was falsely predicted. The actual value was negative but the model predicted as positive.
  False Negative (FN) or Type-II Error: The model predicted negative but it's false, which means the predicted value was falsely predicted. The actual value was positive but the model predicted as negative.
- **Accuracy:** Accuracy means, from all the instances, how many of them our mode predicted correctly. It is calculated as the total number of correct predictions divided by the total number of instances in the dataset. The higher the accuracy, the better the model performance is.
- **Precision:** Precision indicates how many of the instances that were accurately predicted turned out to be positive. It decides whether a model is trustworthy or not. It's beneficial in situations when a False Positive (FP) is more of a concern than a False Negative (FN). The formula for calculating the precision is following:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

- **Recall:** Recall indicates how many of the actual positive cases our model was able to correctly anticipate. It is a useful metric in cases where False Negative (FN) trumps False Positive (FP). The formula for calculating the precision is following:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

- **F1 Score:** F1 Score also known as F-Score or F-Measure is a harmonic mean of Precision and Recall. F-Score comes handy where it is difficult to compare to models with low precision and high recall or vice-versa. The formula for calculating the precision is following:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

C. COMPARISON

We compare the performance of our proposed method with 21 well-known benchmark models and several recently published cutting-edge methods [10]–[14], [30], [45]–[52]. The extensive comparison validates the effectiveness of our method.

1) Comparison with Benchmark Architectures

We conducted a thorough comparison of our proposed method with 21 benchmark architectures to rigorously assess and validate the efficacy of our approach against a diverse set of established models. Specifically, our method was evaluated against 16 convolution-based and 5 transformer-based architectures. To do so, we train these models on our enhanced dataset and report their performance on Table-3, showcasing the effectiveness of our proposed dCNN. Our approach demonstrated superior performance in terms of accuracy, precision, recall, and F1 score when compared to 13 convolution and transformer-based methods, including well-established models such as SwinTransformer, SwinTransformerV2, ResNet50, and ConvNext, to name a few. Moreover, our method achieved competitive performance across the remaining benchmark architectures while maintaining a significantly reduced parameter size. For instance, it outperforms [39], [43], and [38] with 20.5, 147, and 364.7 times fewer parameters, respectively. In contrast, some architectures [40]–[42], [44], [53] gives slightly better performance than ours – however when considering the size of the model parameters, our model outperforms all of them by a great margin.

Type	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	#Param. (M)
Convolution	MNASNet	72.348 ±8.668	59.132 ±5.596	68.636 ±4.52	62.122 ±5.024	3.1
	SqueezeNet	92.912 ±28.038	91.354 ±34.995	92.648 ±29.214	91.584 ±33.912	0.73
	AlexNet	93.525 ±5.219	94.2867 ±4.664	93.5467 ±5.086	93.6617 ±5.022	57.02
	VGG	97.5267 ±1.18	97.6867 ±1.094	97.4767 ±1.133	97.5467 ±1.155	128.78
	ResNet50	99.29 ±1.361	99.3167 ±1.315	99.29 ±1.361	99.2967 ±1.361	23.51
	DenseNet121	99.5 ±0.296	99.54 ±0.293	99.498 ±0.277	99.516 ±0.284	6.95
	ConvNext	99.6017 ±0.445	99.6117 ±0.443	99.6067 ±0.428	99.6117 ±0.433	27.82
	InceptionV3	99.642 ±0.28	99.662 ±0.256	99.644 ±0.296	99.652 ±0.28	25.12
	ResNeXt50	99.664 ±0.257	99.68 ±0.267	99.672 ±0.232	99.676 ±0.248	22.99
	MobileNetV2	99.712 ±0.21	99.718 ±0.215	99.71 ±0.198	99.712 ±0.205	2.23
	EfficientNetV2	99.82 ±0.147	99.83 ±0.147	99.8125 ±0.143	99.82 ±0.148	21.46
	MobileNetV3	99.82 ±0.147	99.8175 ±0.143	99.82 ±0.149	99.815 ±0.147	4.2
	RegNet	99.832 ±0.21	99.84 ±0.203	99.832 ±0.216	99.836 ±0.21	3.9
	ShuffleNetV2	99.832 ±0.21	99.848 ±0.184	99.824 ±0.201	99.836 ±0.193	1.25
	EfficientNet	99.8467 ±0.175	99.81 ±0.18	99.81 ±0.18	99.81 ±0.18	7.8
	GoogleNet	99.902 ±0.192	99.906 ±0.179	99.9 ±0.2	99.904 ±0.185	5.6
Transformer	SwinTransformer_base	27.924 ±10.877	11.004 ±12.097	18.09 ±28.894	13.786 ±11.816	27.52
	SwinTransformerV2_base	70.9 ±37.35	77.34 ±29.314	70.06 ±38.712	69.51 ±42.195	27.58
	VisionTransformer_base	98.89 ±1.072	99.05 ±1.211	98.93 ±0.934	99.08 ±1.453	85.8
	VisionTransformer_large	99.94 ±0.0931	99.92 ±0.126	99.925 ±0.117	99.925 ±0.128	303.3
	MaxViT	99.95 ±0.125	99.954 ±0.115	99.946 ±0.136	99.95 ±0.125	30.41
	Ours	99.808 ±0.286	99.828 ±0.282	99.826 ±0.255	99.826 ±0.269	0.18

TABLE 3. Comparison of the performance of various models in terms of accuracy (Acc.), precision (PR), recall (RE), F1 score (F1), and the number of parameters (#Param.).

Paper	Method	Accuracy	Parameters (M)	Dataset		
				#Classes	#Instances	Source
Hossain et al., 2020 [10]	Custom CNN	97.82%	3.7	5	4199	BRRI [1]
Shrivastava et al., 2021 [13]	SVM	94.07%	-	4	619	Self-generated
Ghosal et al., 2020 [11]	VGG16	93.34%	138	4	1649	IRRI [2]
Azim et al., 2021 [14]	XGBoost	86.58%	-	3	120	UCI ML [15]
Chen et al., 2020 [12]	Custom CNN	92.46%	0.3	5	500	Self-generated
Sethy et al., 2020 [30]	ResNet50 + SVM	98.38%	23.5	5	5932	BCCH [3]
Bhattacharya et al., 2020 [45]	Custom CNN	78.44%	-	3	2000	-
Su et al., 2022 [46]	Custom CNN	81.25%	-	3	120	UCI ML [15]
Ahad et al., 2023 [47]	Densenet121	97.62%	6.95	9	42,876	Combined
Rawat et al., 2023 [48]	ResNet 50	99.50%	23.5	3	4000	Self-generated
Haridasan et al., 2023 [49]	Custom CNN + SVM	91.45%	-	6	-	-
Wang et al., 2023 [50]	ConvNext	94.82%	27.8	7	4523	Combined
Ritharson et al., 2024 [51]	VGG16	99.94%	129	5	5932	Sethy et al., 2020
Din et al., 2024 [52]	Custom CNN	97%	-	5	4748	Self-generated
Ours	Custom CNN	99.65%	0.18	5	5285	Combined + Collected [4]

TABLE 4. Comparison of the performance of our proposed method with other existing methods.

2) Comparison with State-of-the-Art (SOTA) Methods

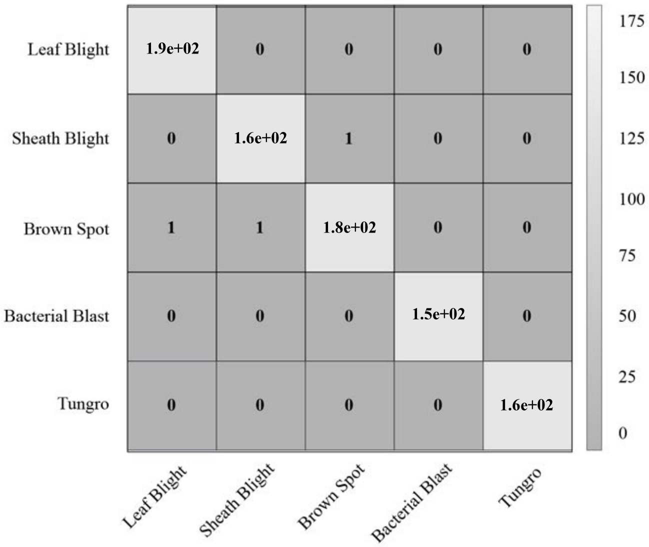
Table 4 contains the performance of our proposed method as well as other existing methods [10]–[14]. Our proposed method outperforms each of these existing methods. It improves the accuracy by 1.8%, precision by 4.87%, recall by 4.66%, and f1 score by 5.1% than [10], which is second best to our method, with 21.7 times fewer parameters. It outperforms [11] and [12] by 7.2% and 5.6% higher accuracy with 811 and 152 times fewer model parameter size. It also outperforms [13] and [14] by attaining 6.3% and 13.1% higher accuracy score respectively.

D. PERFORMANCE ANALYSIS

Our proposed method, RLDD, achieves an impressive 99.65% accuracy on the test set, with a precision of 0.99667, recall of 0.99657, and an f1 score of 0.99674. The performance of our method is explicitly illustrated through the confusion matrix in Figure 5, demonstrating its accuracy in

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3371511

classifying instances of leaf blight, bacterial blast, and tungro. However, it makes a few minor mistakes while classifying instances of sheath blight and brown spot.



	Leaf Blight	Sheath Blight	Brown Spot	Bacterial Blast	Tungro
Leaf Blight	1.9e+02	0	0	0	0
Sheath Blight	0	1.6e+02	1	0	0
Brown Spot	1	1	1.8e+02	0	0
Bacterial Blast	0	0	0	1.5e+02	0
Tungro	0	0	0	0	1.6e+02

FIGURE 5. **Confusion Matrix of Our RLDD Model.**

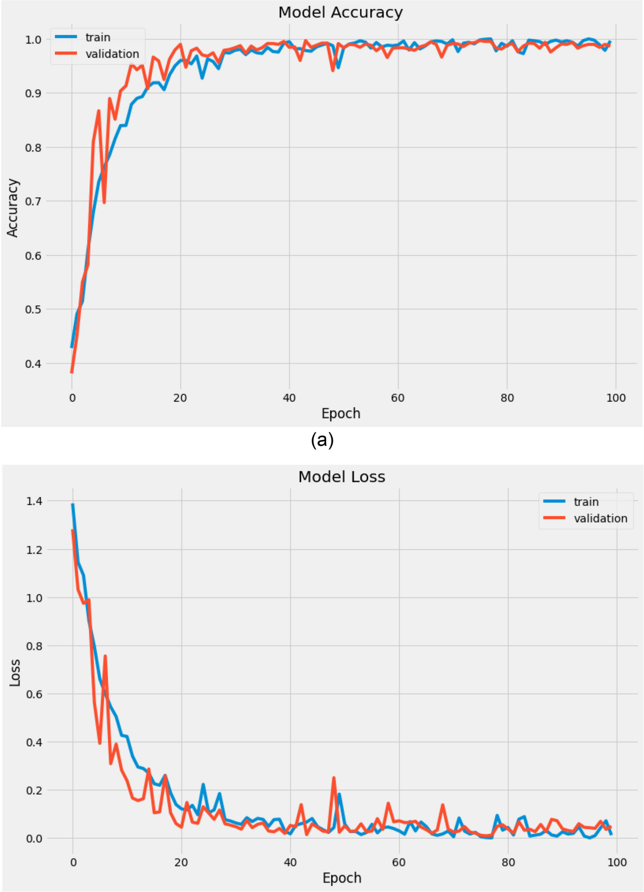
Figure 6(a) showcases the changes in accuracy and loss in our RLDD model concerning epochs. The accuracy vs. epoch graph in Figure 6(a) displays an upward trend, indicating that as the number of epochs increases, the model's accuracy also improves. In contrast, the loss vs. epoch graph in Figure 6(a) shows a downward slope, suggesting that as the epochs progress, the model's loss decreases. These trends indicate that the model is effectively learning to generalize new test cases and improve its performance over time.

E. INTERPRETABILITY OF THE MODEL

Incorporating interpretability into our model enhances its reliability and trustworthiness. The Gradcam technique, introduced by Selvaraju et al. [54], is employed to pinpoint the specific area of the input image that significantly influences the model's prediction. The depiction of the prominent regions, as seen in Figure 7, enhances clarity and reinforces the reliability of our model.

F. FURTHER VALIDATION

We further validate the performance of our proposed dCNN across various environmental conditions and geographical datasets to assess its generalizability.



(a)

(b)

FIGURE 6. **Changes of Accuracy and Loss of our proposed rice leaf disease detection model with respect to epochs. Sub-figures (a) and (b) are model accuracy vs epoch and loss vs epoch graphs respectively.**

1) Model Robustness in Different Scenarios

To bolster the resilience of the proposed model, a comprehensive series of experiments were conducted, encompassing various challenging scenarios encountered during testing. These situations included images placed against natural backgrounds, where the model demonstrated an impressive accuracy of $99.615\%\pm0.156\%$. Furthermore, the model's performance was evaluated under diverse camera angles, showcasing its ability to maintain accuracy at $86.5033\%\pm2.054\%$ even when subjected to random rotations. Additionally, the model's capability to handle varying distances was assessed, resulting in accuracies of $85.8667\%\pm1.646\%$ for zoomed-out images. Moreover, the model's capacity to endure changes in image quality was scrutinized through downsampling and subsequent upsampling, achieving notable accuracies of $99.525\%\pm0.0173\%$ and $99.5233\%\pm0.265\%$, respectively. These findings underscore the robustness and adaptability of the proposed model across diverse environmental conditions and scenarios.

2) Different Geographical Data

We assess the robustness of our model across diverse datasets obtained from various geographical locations. Our analysis involves three datasets sourced from distinct countries, namely China, Indonesia, and Taiwan. The performance metrics presented in Table-5 highlight our model's effectiveness on these geographically diverse datasets. Notably, we

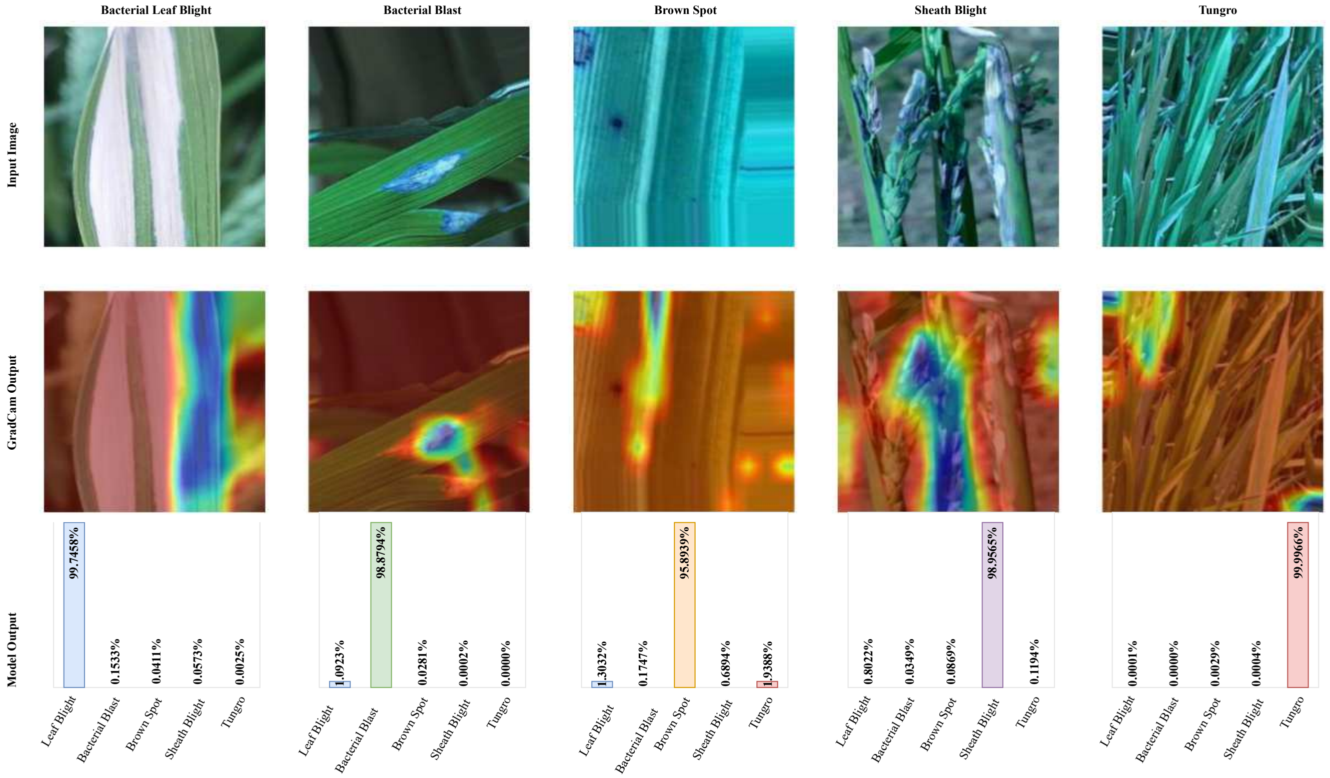


FIGURE 7. The outcomes of the integrated Gradcam for clear model interpretation, revealing influential areas to strengthen confidence in decision-making.

observe a considerable performance drop when training the model on our data and testing it on datasets from different geographical locations, emphasizing the nuanced differences in data distribution. However, through meticulous fine-tuning on diverse geographical data, our model successfully regains its performance, showcasing its adaptability and generalization capabilities.

3) Computational Cost Analysis

We conducted a comprehensive analysis, comparing the training, validation, and inference times of our model against 21 benchmark models. Table-6 presents a clear demonstration of the superior performance and efficiency of our model, surpassing all other models listed by achieving exceptional results with minimal training, validation, and inference times.

G. VARYING LIGHT INTENSITY

We compared the performance of our model with all 21 benchmark architectures under varying light intensities. To adjust image brightness, we increased and decreased the intensity by 20%. Figure-8 illustrates examples of normal, brighter, and darker versions of an instance. To empirically validate the performance, we pursued two approaches. (1) Firstly, we trained all the models on normal data and tested them on lighter or darker data, resulting in lower performance across the board. This outcome was expected, as the models were not trained on similar data and therefore struggled with recognition. (2) Secondly, fine-tuning the models on brighter and darker data enabled all models to resume effective operation.

FIGURE 8. Visualizing diverse light intensity transformations: (Left) darker with decreased intensity, (Right) brighter with increased intensity, (Middle) unaltered.

Table-7 elucidates the empirical results across diverse light intensities for all models. Our model demonstrates unwavering performance consistency across normal, brighter, and darker data, underscoring its efficacy. Particularly noteworthy is its outperformance of various convolutional-based and transformer-based methods, showcasing competitive performance among other benchmarks. Moreover, it significantly outshone all models in parameter size, marking a notable advancement in efficiency.

Location	Before Fine-tuning (%)				After Fine-tuning (%)			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
Indonesia	81.75	0.8244	0.8211	0.8116	97.91	0.9865	0.9791	0.9811
Taiwan	68.81	0.7135	0.6728	0.6864	96.19	0.9708	0.9399	0.9539
China	60.11	0.5224	0.5336	0.5169	91.05	0.9202	0.9043	0.9099

TABLE 5. **Comparison of model performance before and after fine-tuning on different geographical data.**

Type	Model	Parameters (M)	Time per Epoch (S)		
			Training	Validation	Test
Convolution	MNASNet	3.1	17.9333 ±1.311	4.3667 ±0.705	3 ±0
	SqueezeNet	0.73	15.7 ±1.663	4.7333 ±1.27	3 ±0
	AlexNet	57.02	14.3 ±1.83	6.4 ±2.568	2 ±0
	VGG	128.78	32.3333 ±1.138	8.4667 ±1.731	4 ±0
	ResNet50	23.51	50.1333 ±0.58	8.6 ±1.663	5 ±0
	DenseNet121	6.95	38.7 ±0.61	7.1667 ±0.932	5 ±0
	ConvNext	27.82	58.5 ±2.271	8.2333 ±1.09	5 ±0
	InceptionV3	25.12	57.9333 ±1.161	10.9667 ±0.932	7 ±0
	ResNeXt50	22.99	50.3667 ±0.705	8 ±0.692	5 ±0
	MobileNetV2	2.23	19.2 ±0.61	4.7333 ±1.5	3 ±0
	EfficientNetV2	21.46	41.4 ±1.153	7.5333 ±1.351	4 ±0
	MobileNetV3	4.2	16.6333 ±1.351	4.4667 ±1.27	4 ±0
	RegNet	3.9	17.8 ±0.461	4.5 ±1.153	4 ±0
	ShuffleNetV2	1.25	12.3667 ±1.27	4.2333 ±1.27	3 ±0
	EfficientNet	7.8	29.8667 ±0.705	5.9667 ±0.352	3 ±0
	GoogleNet	5.6	19.8 ±0.922	4.7667 ±0.96	3 ±0
Transformer	SwinTransformer_base	27.52	48.1667 ±0.58	8.4333 ±0.705	5 ±0
	SwinTransformerV2_base	27.58	56.4 ±2.224	9.7 ±1.057	6 ±0
	VisionTransformer_base	85.8	112.2 ±2.619	16.2333 ±1.311	10 ±0
	VisionTransformer_large	303.3	427.4667 ±1.62	60.5 ±5.222	16 ±0
	MaxViT	30.41	74.8333 ±0.48	11.9667 ±0.58	8 ±0
	Ours	0.18	7.5333 ±0.352	2.5667 ±0.352	1 ±0

TABLE 6. **Comparative analysis of training, validation, and inference times for our model and 21 benchmark models.**

H. EXPERT SYSTEM

We deploy our proposed model through a website and an android app. To do so, an open API has been developed which is essentially a gateway between server and web or app interface.

1) Application Programming Interface (API)

An API is a set of programming code that allows data to be exchanged between two or more software products, like websites and apps. It is quicker, flexible, secured and responsive. We develop an open API that is accessible for everyone. It takes an image as query and returns the crop name, disease name, and disease details. Figure 9 illustrates the endpoint of our API.

We deploy our proposed model and kept it on the server. The website and the android app interact with the pre-trained model via the API. As a result, the user's device uses very little memory which makes it very convenient from the user's perspective. For instance, when a user uploads an image to our website or app, the API takes the image as an input query. Then it sends the image to the server and makes a prediction about it using the model. Finally, it returns the prediction of the model from the server to web or app interface.

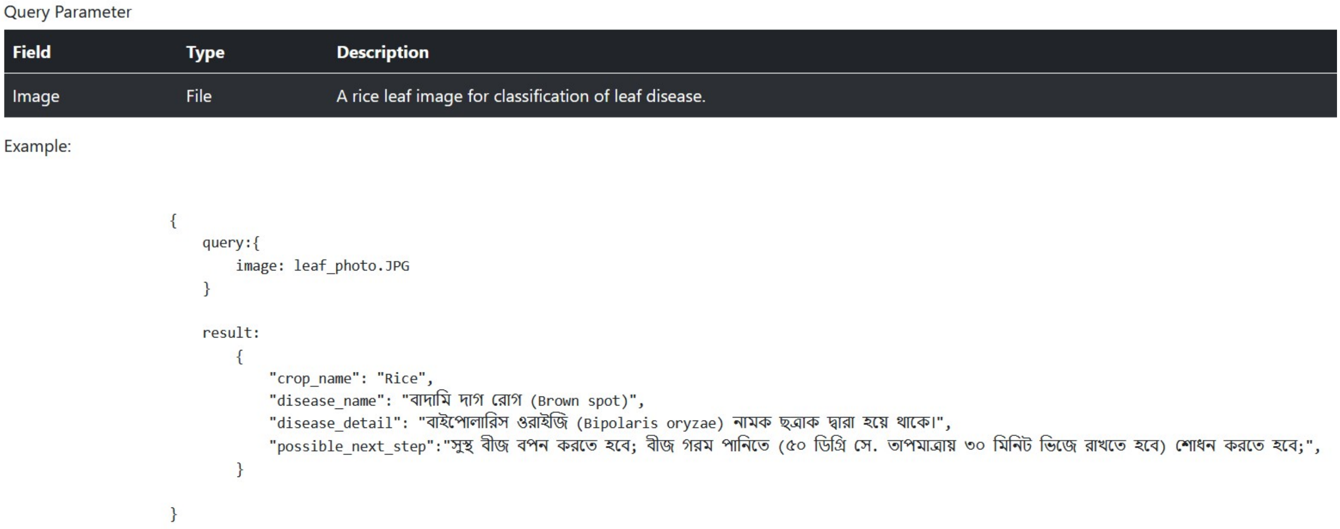
2) Web Interface

A web user interface, often known as a Web app, allows a user to use a web browser to interact with data or software on a web server. A web app is platform independent as well. As a consequence, we have seen a huge increase in popularity of web-based application in recent years. Websites are flexible as it can be accessed through browser application (web browser) from both computer and mobile devices. Figure 10 depicts the interface of our developed website.

As previously stated, the website communicates with the server via the API. The website's interface is simple enough for someone with only a rudimentary understanding of browsing websites to use. A user would simply upload an image from the device by clicking the **Choose file** button, navigating to the desired image location, and clicking the **Submit** button. The predictions will be displayed on the website, along with other pertinent information such as crop name, disease details, and possible next steps. Figure 11 depicts the disease

Type	Model	Parameters (M)	Accuracy (%)			
			Darker	Brighter	Normal	Average
Convolution	MNASNet	3.1	66.58 ±3.667	69.62 ±6.231	72.348 ±8.668	69.516 ±6.189
	SqueezeNet	0.73	88.115 ±1.228	92.2 ±2.134	92.912 ±2.038	91.076 ±1.8
	AlexNet	57.02	64.35 ±3.252	84.305 ±7.593	93.525 ±5.219	80.727 ±5.355
	VGG	128.78	92.785 ±1.505	95.834 ±1.672	97.5267 ±1.18	95.382 ±1.452
	ResNet50	23.51	92.405 ±4.272	98.41 ±0.208	99.29 ±1.361	96.702 ±1.947
	DenseNet121	6.95	94.82 ±4.082	98.76 ±1.003	99.5 ±0.296	97.693 ±1.794
	ConvNext	27.82	98.71 ±0.0346	99.465 ±0.605	99.6017 ±0.445	99.259 ±0.362
	InceptionV3	25.12	98.76 ±1.003	99.54 ±1.472	99.642 ±0.28	99.314 ±0.918
	ResNeXt50	22.99	97.41 ±2.041	99.565 ±0.0173	99.664 ±0.257	98.88 ±0.772
	MobileNetV2	2.23	98.995 ±1.816	99.585 ±0.605	99.712 ±0.21	99.431 ±0.877
	EfficientNetV2	21.46	99.11 ±0.208	99.41 ±0.421	99.82 ±0.147	99.447 ±0.259
	MobileNetV3	4.2	98.935 ±0.813	99.465 ±0.19	99.82 ±0.147	99.407 ±0.383
	RegNet	3.9	99.23 ±1.418	99.82 ±0.208	99.832 ±0.21	99.627 ±0.612
	ShuffleNetV2	1.25	99.055 ±0.813	99.273 ±0.17	99.832 ±0.21	99.387 ±0.398
	EfficientNet	7.8	99.525 ±0.398	99.766 ±0.423	99.8467 ±0.175	99.713 ±0.332
	GoogleNet	5.6	98.405 ±1.02	99.633 ±0.31	99.902 ±0.192	99.313 ±0.507
Transformer	SwinTransformer_base	27.52	22.0733 ±10.146	25.11 ±10.377	27.924 ±10.877	25.036 ±10.467
	SwinTransformerV2_base	27.58	62.705 ±0.0173	66.195 ±1.747	70.9 ±1.35	66.6 ±1.038
	VisionTransformer_base	85.8	89.645 ±4.731	96.41 ±0.208	98.89 ±1.072	94.982 ±2.004
	VisionTransformer_large	303.3	93.724 ±2.21	97.7 ±1.418	99.94 ±0.0931	97.121 ±1.24
	MaxViT	30.41	98.5067 ±0.489	99.87 ±0.0346	99.95 ±0.125	99.442 ±0.216
	Ours	0.18	99.2833 ±0.641	99.71 ±0.138	99.808 ±0.286	99.6 ±0.355

TABLE 7. Comparison of model performance in terms of accuracy under different light intensity transformations.



Query Parameter

Field	Type	Description
Image	File	A rice leaf image for classification of leaf disease.

Example:

```
{
    query:{
        image: leaf_photo.JPG
    }

    result:
        {
            "crop_name": "Rice",
            "disease_name": "বাদামি দাগ রোগ (Brown spot)",
            "disease_detail": "বাইপোলারিস ওরাইজি (Bipolaris oryzae) নামক ছত্রাক দ্বারা হয়ে থাকে।",
            "possible_next_step":"সুস্থ বীজ বপন করতে হবে; বীজ গরম পানিতে (৫০ ডিগ্রি সে. তাপমাত্রায় ৩০ মিনিট ভিজে রাখতে হবে) শোধন করতে হবে;",
        }

}
```

FIGURE 9. API endpoint

detection process for rice leaf disease.

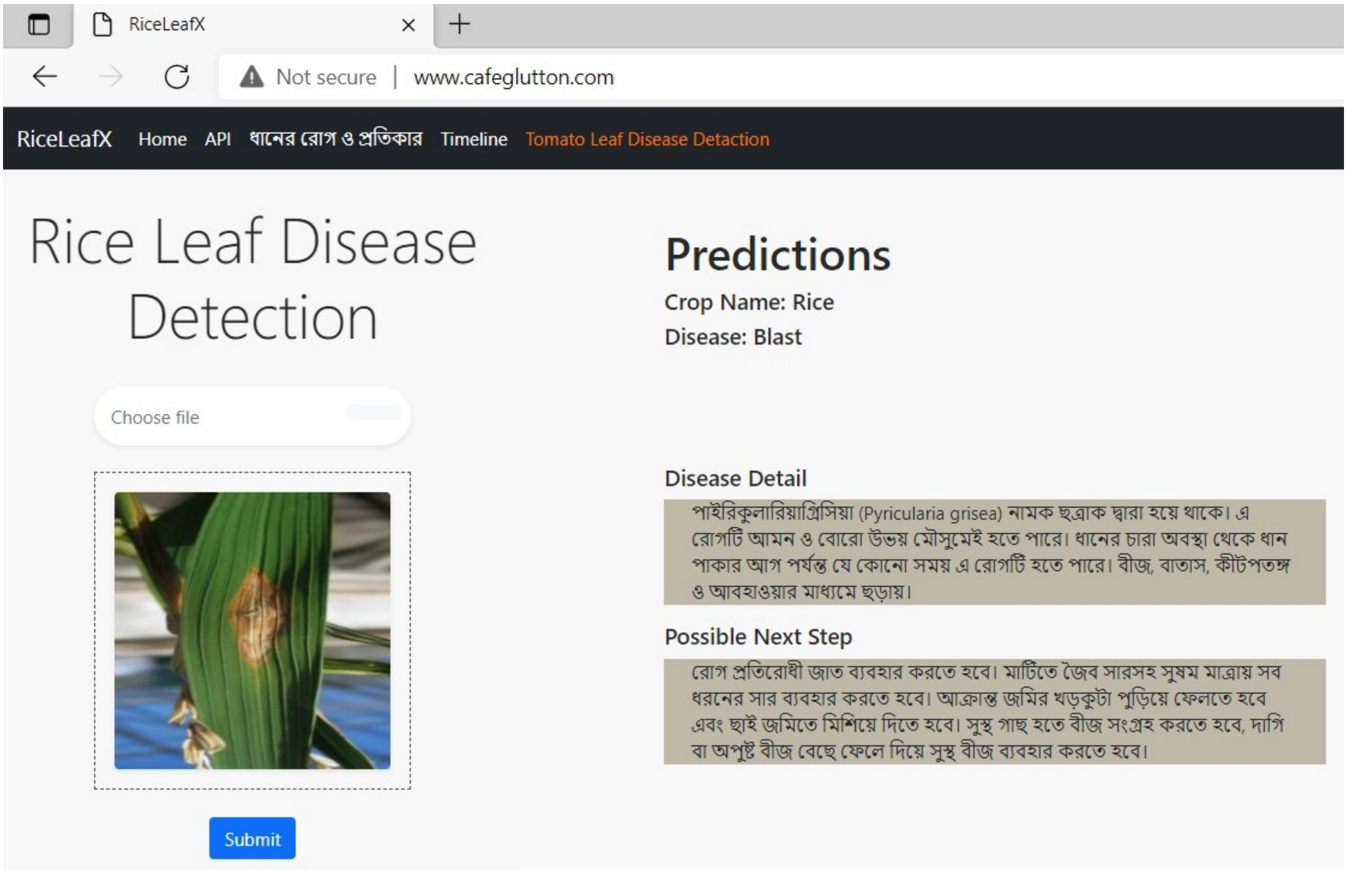
Website also provides extra information pertaining to rice leaf diseases. For instance, it has a section which contains the reasons of a disease and the remedies. A user can obtain these information from the website as well. In navigation bar, an option for API is included. Users can find useful information pertaining to the API that is working behind the website.

3) APP Interface

Due to the affodability and availabilty, smartphones have now become ubiquitous in developing countries. Consequently, android application can play a significant role in improving the model's usability. Figure 12 illustrates our developed android app interface and steps for predicting rice or leaf diseases using it.

We develop the app considering low resource devices and

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3371511



Rice Leaf Disease Detection

Predictions

Crop Name: Rice

Disease: Blast

Disease Detail

পাইরিকুলারিয়াগ্রিসিয়া (Pyricularia grisea) নামক ছত্রাক দ্বারা হয়ে থাকে। এ রোগটি আমন ও বোরো উভয় মৌসুমেই হতে পারে। ধানের চারা অবস্থা থেকে ধান পাকার আগ পর্যন্ত যে কোনো সময় এ রোগটি হতে পারে। বীজ, বাতাস, কীটপতঙ্গ ও আবহাওয়ার মাধ্যমে ছড়ায়।

Possible Next Step

রোগ প্রতিরোধী জাত ব্যবহার করতে হবে। মাটিতে জৈব সারসহ সুষম মাত্রায় সব ধরনের সার ব্যবহার করতে হবে। আক্রান্ত জমির খড়কুটা পুড়িয়ে ফেলতে হবে এবং ছাই জমিতে মিশিয়ে দিতে হবে। সুস্থ গাছ হতে বীজ সংগ্রহ করতে হবে, দাগি বা অপুষ্ট বীজ বেছে ফেলে দিয়ে সুস্থ বীজ ব্যবহার করতে হবে।

FIGURE 10. Web interface

Rice Leaf Disease Detection

Predictions

Crop Name:

Disease:

Rice Leaf Disease Detection

Predictions

Crop Name:

Disease:

Rice Leaf Disease Detection

Predictions

Crop Name: Rice

Disease: Bacterial leaf blight

Disease Detail

জ্যানথোমোনাস অরাইজি পিভি অরাইজি (Xanthomonas oryzae pv.oryzae) এক ধরনের ব্যাকটেরিয়ার আক্রমণে হয়ে থাকে।

Possible Next Step

রোগ প্রতিরোধী জাত চাষ করতে হবে। রোগাক্রান্ত জমির ধান কাটার পর নাড়া ও খড় পুড়িয়ে ফেলতে হবে। চারা উঠানোর সময় যেন শিকড় কম ছিঁড়ে সে ব্যাপারে সতর্ক থাকা। সুষম মাত্রায় সার ব্যবহার ও ইউরিয়া সার ৩/৪ কিস্তিতে প্রয়োগ করতে হবে।

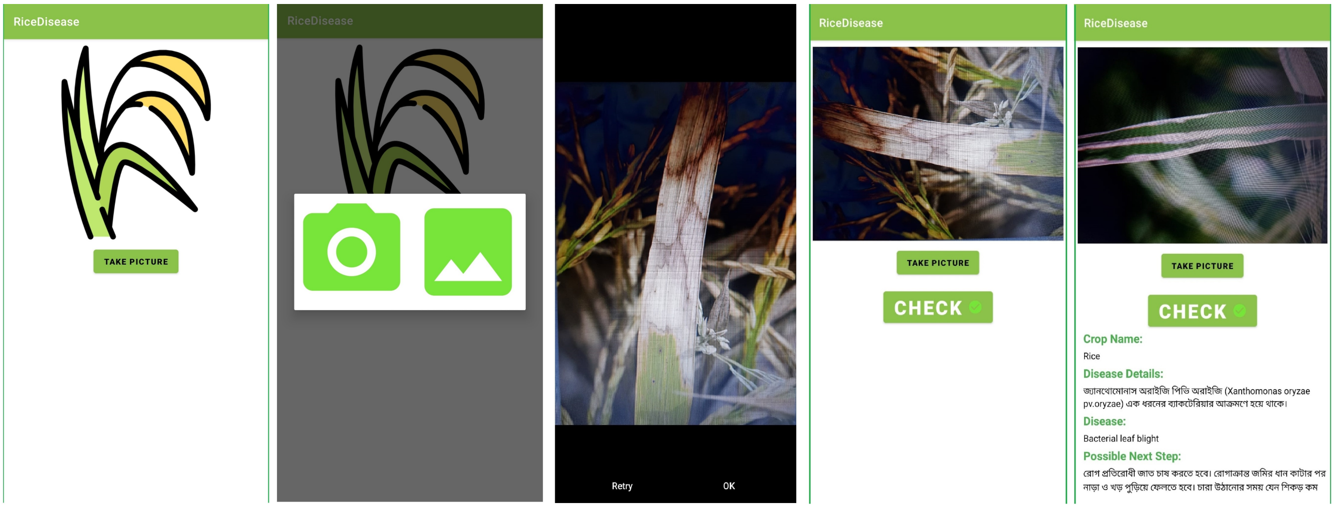
FIGURE 11. (Left) The web interface for rice leaf disease detection. (Middle) Uploaded an image from the device. (Right) Get predictions for the uploaded image.

its users. We kept the interface simple, so that, users can use it without any supervision. The interface directly redirects a user to the main page. The user gets an option for taking a picture or selecting an image from the gallery via the app. After capturing or selecting the appropriate image, the user can send it for prediction by clicking the **Check** button. Once the image is sent, the API receives the image and the results obtained will be returned to the app interface. The app requires very little space as the model does not run on the user's device. Hence, users with low configured devices will be able to use the app smoothly.

VI. DISCUSSION

We conducted a comprehensive evaluation of our model, comparing it with 21 established benchmark models utilizing a spectrum of convolution and transformer based architectures, each varying in trainable parameters from 73,000 to 303.3 million. The performance of our model was rigorously assessed against different environmental situations including varying light intensity, camera angle, varying distances, different image quality and images with natural background. We

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3371511



Crop Name:
Rice
Disease Details:
জ্যানথোমোনাস অরাইজি পিভি অরাইজি (Xanthomonas oryzae pv.oryzae) এক ধরনের ব্যাকটেরিয়ার আক্রমণে হয়ে থাকে।
Disease:
Bacterial leaf blight
Possible Next Step:
রোগ প্রতিরোধী জাত চাষ করতে হবে। রোগাক্রান্ত জমির ধান কাটার পর নাড়া ও খড় পুড়িয়ে ফেলতে হবে। চারা উঠানোর সময় যেন শিকড় কম

FIGURE 12. APP interface

also tested our model with images from different geographic locations. This analysis aimed to demonstrate the robustness and generalizability of the models across a broad range of scenarios. Our model demonstrates high performance under different situations. Moreover, our model shows a much superior performance in terms of the trainable parameters of the model. This makes our model useful to be operate on edge devices and in offline mode, particularly in remote areas where internet connectivity is limited or unreliable. Furthermore, our developed API can be leveraged for annotation purposes, effectively addressing the challenge of data scarcity.

One of the limitations of our model is that it can predict five rice leaf diseases. Other rice leaf diseases as well healthy rice leaves cannot be predicted by this model. In future, we plan to overcome these challenges by collecting more relevant data and training with dCNN.

VII. CONCLUSIONS AND FUTURE WORK

Rice leaf disease detection is an inevitable task to increase production. Early identification of rice leaf diseases will assist farmers in saving their harvest from getting affected by diseases. Existing methods for rice leaf disease detection are found to be rather ineffective for several reasons. From a pragmatic viewpoint, the solution needs to be operational within resource-constrained environments. This means the model would need to work with fewer parameters and be as light as possible. This paper presents a lightweight dCNN based model for detecting five of the most common rice leaf diseases including brown spot, tungro, bacterial blight, sheath blight, and bacterial blast. We compare the performance of our model with 21 benchmark architectures and 14 concurrent methods. The extensive experimental outcomes validate the effectiveness of our method and demonstrate efficiency in detecting diseases which in turn will help the farmers to save the production losses at an early stage. The nature of our proposed method is end-to-end. It achieves competitive performance with benchmark architectures with much lower asymptotic complexity and shows superior performance with existing methods. We enhance an existing dataset by manually collecting data and annotating them by experts. This study brings forward more varieties of rice leaf diseases and a fine-tuned dCNN model, following an end-to-end manner, which gives accurate performance with significantly lower asymptotic complexity. Additionally, the subsequent research develops an integrated application to fit in low-end devices which includes an API, an android APP and a website.

The activities in the project builds upon much of the findings generated from a field study conducted with agricultural communities in rural Bangladesh. A range of challenges were identified, and access to expertise was one of them. The activity described in this paper aims to address this need within the community, where farmers could benefit from the availability of expert guidance on disease and treatment. Therefore, future work in the project would involve conducting evaluations with farmer communities to understand how the app performs in real world contexts. This would also involve conducting user studies to understand usability of the applications and scalability of the API.

ACKNOWLEDGEMENT

This research was funded by X/161099 GCRF QR Project. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

REFERENCES

[1] S. M. Shaheen, V. Antoniadis, M. Shahid, Y. Yang, H. Abdelrahman, T. Zhang, N. E. Hassan, I. Bibi, N. K. Niazi, S. A. Younis *et al.*, "Sustainable applications of rice feedstock in agro-environmental and construction sectors: a global perspective," *Renewable and Sustainable Energy Reviews*, vol. 153, p. 111791, 2022.

[2] I. J. Shelley, M. Takahashi-Nosaka, M. Kano-Nakata, M. S. Haque, and Y. Inukai, "Rice cultivation in bangladesh: present scenario, problems,

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3371511

and prospects," *Journal of International Cooperation for Agricultural Development*, vol. 14, no. 4, pp. 20–29, 2016.

[3] A. Haque, N. Islam, N. H. Samrat, S. Dey, and B. Ray, "Smart farming through responsible leadership in bangladesh: possibilities, opportunities, and beyond," *Sustainability*, vol. 13, no. 8, p. 4511, 2021.

[4] S. Mazumdar, S. Ehdeed, A. Jimenez, F. Ahmed, S. Momen, and M. Rasheduzzaman, "Understanding the information landscape in agricultural communities in rural bangladesh," *The Electronic Journal of Information Systems in Developing Countries*, vol. 89, no. 1, p. e12245, 2023.

[5] M. E. Pothen and M. L. Pai, "Detection of rice leaf diseases using image processing," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2020, pp. 424–430.

[6] N. Manohar and K. J. Gowda, "Image processing system based identification and classification of leaf disease: A case study on paddy leaf," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020, pp. 451–457.

[7] K. Ahmed, T. R. Shahidi, S. M. I. Alam, and S. Momen, "Rice leaf disease detection using machine learning techniques," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE, 2019, pp. 1–5.

[8] F. Zhao, Y. Chen, Y. Hou, and X. He, "Segmentation of blood vessels using rule-based and machine-learning-based methods: a review," *Multimedia Systems*, vol. 25, pp. 109–118, 2019.

[9] M. A. Khan, A. Alqahtani, A. Khan, S. Alsubai, A. Binbusayyis, M. M. I. Ch, H.-S. Yong, and J. Cha, "Cucumber leaf diseases recognition using multi level deep entropy-elm feature selection," *Applied Sciences*, vol. 12, no. 2, p. 593, 2022.

[10] S. M. M. Hossain, M. M. M. Tanjil, M. A. B. Ali, M. Z. Islam, M. S. Islam, S. Mobassirin, I. H. Sarker, and S. R. Islam, "Rice leaf diseases recognition using convolutional neural networks," in *International Conference on Advanced Data Mining and Applications*. Springer, 2020, pp. 299–314.

[11] S. Ghosal and K. Sarkar, "Rice leaf diseases classification using cnn with transfer learning," in *2020 IEEE Calcutta Conference (CALCON)*. IEEE, 2020, pp. 230–236.

[12] J. Chen, D. Zhang, Y. A. Nanehkaran, and D. Li, "Detection of rice plant diseases based on deep transfer learning," *Journal of the Science of Food and Agriculture*, vol. 100, no. 7, pp. 3246–3256, 2020.

[13] V. K. Shrivastava and M. K. Pradhan, "Rice plant disease classification using color features: a machine learning paradigm," *Journal of Plant Pathology*, vol. 103, no. 1, pp. 17–26, 2021.

[14] M. A. Azim, M. K. Islam, M. M. Rahman, F. Jahan *et al.*, "An effective feature extraction method for rice leaf disease classification," *Telkomnika*, vol. 19, no. 2, pp. 463–470, 2021.

[15] "UCI Machine Learning Repository: Rice Leaf Diseases Data Set," https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases, [Online; accessed: 04.08.2023].

[16] S. Saha and S. M. M. Ahsan, "Rice disease detection using intensity moments and random forest," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*. IEEE, 2021, pp. 166–170.

[17] P. Mekha and N. Teeyasuksaet, "Image classification of rice leaf diseases using random forest algorithm," in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*. IEEE, 2021, pp. 165–169.

[18] C. R. Rahman, P. S. Arko, M. E. Ali, M. A. I. Khan, S. H. Apon, F. Nowrin, and A. Wasif, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosystems Engineering*, vol. 194, pp. 112–120, 2020.

[19] M. A. Islam, M. N. R. Shuvo, M. Shamsojjaman, S. Hasan, M. S. Hossain, and 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.

[20] Z. Jiang, Z. Dong, W. Jiang, and Y. Yang, "Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning," *Computers and Electronics in Agriculture*, vol. 186, p. 106184, 2021.

[21] N. Krishnamoorthy, L. N. Prasad, C. P. Kumar, B. Subedi, H. B. Abraha, and V. Sathishkumar, "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environmental Research*, vol. 198, p. 111275, 2021.

[22] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.

[23] Y. Wang, H. Wang, and Z. Peng, "Rice diseases detection and classification using attention based neural network and bayesian optimization," *Expert Systems with Applications*, vol. 178, p. 114770, 2021.

[24] A. Rahman, S. N. Shoumik, M. Rahman, H. Hena *et al.*, "Rice disease detection based on image processing technique," in *Smart Trends in Computing and Communications: Proceedings of SmartCom 2020*. Springer, 2021, pp. 135–145.

[25] K. Kiratiratanapruk, P. Temniranrat, W. Sinthupinyo, S. Marukatat, and S. Patarapuwadol, "Automatic detection of rice disease in images of various leaf sizes," *arXiv preprint arXiv:2206.07344*, 2022.

[26] S. K. Upadhyay and A. Kumar, "A novel approach for rice plant diseases classification with deep convolutional neural network," *International Journal of Information Technology*, vol. 14, no. 1, pp. 185–199, 2022.

[27] G. Kathiresan, M. Anirudh, M. Nagharjun, and R. Karthik, "Disease detection in rice leaves using transfer learning techniques," in *Journal of Physics: Conference Series*, vol. 1911, no. 1. IOP Publishing, 2021, p. 012004.

[28] S. S. H. Rumy, M. I. A. Hossain, F. Jahan, and T. Tanvin, "An iot based system with edge intelligence for rice leaf disease detection using machine learning," in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE, 2021, pp. 1–6.

[29] W.-j. Liang, H. Zhang, G.-f. Zhang, and H.-x. Cao, "Rice blast disease recognition using a deep convolutional neural network," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.

[30] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, p. 105527, 2020.

[31] "Rice Diseases Image Dataset," https://www.kaggle.com/datasets/minhhuy2810/rice-diseases-image-dataset/versions/1, [Online; accessed: 04.08.2023].

[32] "Rice Disease Dataset," https://www.kaggle.com/datasets/nischallal/rice-disease-dataset, [Online; accessed: 04.08.2023].

[33] "Rice Leaf Diseases Dataset," https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases, [Online; accessed: 04.08.2023].

[34] G. Geetharamani and A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Computers & Electrical Engineering*, vol. 76, pp. 323–338, 2019.

[35] P. Paul, M. A.-U.-A. Bhuiya, M. A. Ullah, M. N. Saqib, N. Mohammed, and S. Momen, "A modern approach for sign language interpretation using convolutional neural network," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2019, pp. 431–444.

[36] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, no. 7, p. 939, 2019.

[37] S. Sharif, N. Mohammed, S. Momen, and N. Mansoor, "Classification of bangla compound characters using a hog-cnn hybrid model," in *Proceedings of the International Conference on Computing and Communication Systems: I3CS 2016, NEHU, Shillong, India*. Springer, 2018, pp. 403–411.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[40] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[43] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[44] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[45] S. Bhattacharya, A. Mukherjee, and S. Phadikar, "A deep learning approach for the classification of rice leaf diseases," *Intelligence Enabled Research: DoSIER 2019*, pp. 61–69, 2020.

[46] N. T. Su, P. D. Hung, B. T. Vinh, and V. T. Diep, "Rice leaf disease classification using deep learning and target for mobile devices," in *Proceedings of International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2021 (Volume 1)*. Springer, 2022, pp. 136–148.

[47] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of cnn-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, 2023.

[48] P. Rawat, A. Pandey *et al.*, "Rice leaf diseases classification using deep learning techniques," in *2023 International Conference on Networking and Communications (ICNWC)*. IEEE, 2023, pp. 1–8.

[49] A. Haridasan, J. Thomas, and E. D. Raj, "Deep learning system for paddy plant disease detection and classification," *Environmental Monitoring and Assessment*, vol. 195, no. 1, p. 120, 2023.

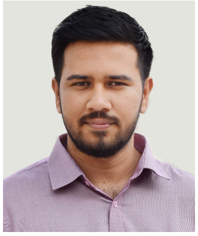
[50] X. Wang, Y. Wang, J. Zhao, and J. Niu, "Eca-convnext: A rice leaf disease identification model based on convnext," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6234–6242.

[51] P. I. Ritharson, K. Raimond, X. A. Mary, J. E. Robert, and J. Andrew, "Deeprice: A deep learning and deep feature based classification of rice leaf disease subtypes," *Artificial Intelligence in Agriculture*, vol. 11, pp. 34–49, 2024.

[52] N. M. U. Din, A. Assad, R. A. Dar, M. Rasool, S. U. Sabha, T. Majeed, Z. U. Islam, W. Gulzar, and A. Yaseen, "Ricenet: A deep convolutional neural network approach for classification of rice varieties," *Expert Systems with Applications*, vol. 235, p. 121214, 2024.

[53] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[54] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.



MEHEDI HASAN BIJOY is currently pursuing his MSc degree at Aalto University, specializing in Computer, Communication, and Information Sciences, with a primary focus on Speech and Language Technology. Prior to this, he earned his BSc degree in Computer Science and Engineering with Summa Cum Laude distinction from North South University in 2021. Along his academic journey, he has held various roles, including working as a lecturer at the Bangladesh University of Business and Technology, as a research assistant at United International University, and as a teaching assistant followed by a lab instructor at North South University. Furthermore, he served as a reviewer for the Bangla Language Processing workshop at EMNLP 2023.

NIROB HASAN received a Bachelor of Science degree in Computer Science and Engineering from North South University, Dhaka, Bangladesh in 2021. Commencing his career as a software engineer in 2022, he has demonstrated expertise in diverse domains of software development. His current responsibilities encompass the development of production-level APIs, which involve the integration of various business logic and AI models. Additionally, he specializes in designing responsive web applications. Previously, he worked at the Bangladesh Open-Source Network, where he served as a programming mentor from 2018 to 2020. He diligently coordinated programming camps and actively engaged in educational outreach.

MITHUN BISWAS is a dedicated Research Engineer at FS Solution Co., Ltd in South Korea. He pursued his B.Sc. degree in Computer Science and Engineering from the University of Liberal Arts Bangladesh, graduating in 2017. After graduation, he delved into software development, amassing two years of professional experience. In 2022, Mithun pivoted to the domain of Machine Learning Engineering. He has a keen interest in machine learning, image processing, and computer vision research. Mithun's dedication to academia is evident through his notable contributions. He has published several papers in esteemed conferences and journals, including the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop, Expert System, and the IEEE Journal of Biomedical and Health Informatics.

SUVODEEP MAZUMDAR is a senior lecturer in Data Analytics. His research explores developing techniques and mechanisms for reducing the barriers that impede user communities' understanding of vast complex multidimensional datasets. He conducts interdisciplinary research on highly engaging, interactive, and visual mechanisms in conjunction with complex querying techniques for seamless navigation, exploration, and understanding of complex datasets. Dr Mazumdar has applied his research in a wide range of application domains, such as Aerospace Engineering, Sports Informatics, Crisis/Emergency Management, Smart Cities, and Mobility Planning. As a part of his research, he collaborates with large multi-disciplinary teams of academics, industry partners, city councils, and planners. He has worked in several extensive research and industrial projects funded by the UKRI, European Union, Innovate UK, and the European Space Agency. Dr Mazumdar's research includes: (1) Studying and developing data and visual analytic techniques to analyze massive volumes of dynamic data in near real-time; (2) Citizen Science and Crowdsourcing Techniques for observing physical phenomena, events and environments; (3) User Interface Development, HumanComputer Interaction and user-centred Design; and (4) Assistive technologies to support independent activities of daily living.

ANDREA JIMENEZ is a lecturer in Information Management at the University of Sheffield's Information School. She holds a PhD from the School of Management at Royal Holloway University of London, and a Masters in Sustainable Development, with a focus on social innovation and ICTs for Development (ICT4D). As part of her PhD, she conducted in-depth qualitative research on technology and innovation hubs, focusing on innovation processes occurring within hubs in both the Global North and Global South. Her interest revolves around intersectionality, decoloniality, and innovation. She compliments this academic career with industry experience, having worked for the United Nations in both the Food and Agriculture Organization (FAO) and the International Telecommunications Union (ITU), in addition to the Alliance for Affordable Internet (A4AI). Her research interests revolve around the impact of digital innovation, social innovation and entrepreneurship

FAISAL AHMED is a seasoned professional with more than 18 years of experience in industry, rural development, market systems, international development in several organizations, and managing projects across different countries in Asia. He is highly experienced in applying technology innovations for social and economic development of various communities and working with large teams of diverse cultural and professional background. He earned numerous accolades for applying innovation in the space of Technology of Development (T4D) in multiple domains that include climate change mitigation, adaptation, healthcare, COVID-19 crisis management and more. He has worked in several renowned organizations like Shakti Foundation, BRAC, ReliSource, IQVIA and so on and worked with bilateral and corporate donors including USAID, EKN, SIDA, RSA, ADB, FCDO etc.

MIRZA RASHEDUZZAMAN is currently an Associate professor in the department of Electrical and Electronic Engineering at the University of Liberal Arts Bangladesh. He obtained his Ph.D. and M.Sc. degrees from the University of Sheffield in United Kingdom, and his B.Sc. degree from the North South University in Dhaka, Bangladesh. His current research interests are internet of things, smart systems, energy harvesting and electric vehicles. He received several research grants including the ICT innovation fund from the formation and Communication Technology (ICT) Division, Ministry of Posts, Telecommunications and Information Technology, Bangladesh Government.

SIFAT MOMEN received his PhD degree in 2011 from the University of Sheffield, UK. In September 2017, Dr. Momen joined as an Assistant Professor in the department of Electrical and Computer Engineering (ECE) of North South University (NSU). He is currently employed in the same department as an Associate Professor. Prior to joining NSU, Dr. Momen worked as an Assistant Professor for nearly six years in the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh (ULAB). He also served as the acting Head of the Department at ULAB, for some time. His PhD research was at the crossroads of biology and engineering, and was heavily influenced by the behavior of eusocial insects, which are well-known for their self-organizing abilities. His current research interests include complex systems, machine learning, information systems, and modeling and simulation of natural systems. He is an active researcher and regularly reviews numerous conference papers and journal articles. He has published over 60 peer-reviewed papers in notable conference and journal venues including IEEE Access, Plos One and Diagnostics.

• • •