Plag report

by Soumyadeep Laskar

Submission date: 13-May-2023 08:36PM (UTC+0530)
Submission ID: 2092134587
File name: Copy_of_Final_Report_Major_Project.pdf (996.44K)
Word count: 2981
Character count: 16097

Detection of Rice Leaf Disease Using Machine Learning Techniques

Submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Technology
in
Electronics and Communication Engineering

Submitted by
Soumyadeep Laskar
IEC2019031

Under the
Supervision of
Dr. Suneel Yadav

Department of Electronics and Communication Engineering
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
ALLAHABAD
Prayagraj -211015

May, 2023

ABSTRACT

In this project work, the focus was mainly on exploring different techniques to detect “Rice Leaf Disease”. This project proposes a deep learning-based approach for the detection of rice leaf disease using Convolutional Neural Networks (CNN). The dataset used to train our model was taken from Sambalpur University [2] consisting of 4 types of diseases based on the types of brown spots the rice leaves have grown. Altogether the proposed model has achieved an accuracy of more than 96% on the test dataset.

Moreover, the proposed model is also implemented in an android application, which can be used by farmers to detect the rice leaf disease using only their android mobile phones in offline mode, so that farmers living in remote Indian villages with weak internet connectivity can also use our application. With the use of the practical tool provided by the android application created for this project, farmers may easily recognise and treat diseases affecting rice leaves, potentially increasing yields and decreasing yield losses

CONTENTS

Introduction

In the Indian subcontinent, rice is a crucial crop and the main food supply for the vast majority of people. However, rice leaf diseases pose a serious threat to rice production and, if not well controlled, can result in large yield losses. Reduced grain quality, lower grain yields, and even crop failure are all possible consequences of these diseases. Therefore, prompt diagnosis and treatment of these diseases are essential for guaranteeing both sustainable agriculture and food security.

Convolutional neural networks (CNNs), in particular, have drawn interest recently for their ability to automate the identification of illnesses affecting plant diseases like millet [4] ,tomato [5] , banana leaf [7]. These methods make it possible to precisely diagnose and categorize various diseases by processing and analyzing massive amounts of data, including digital photographs of rice leaves.

In this thesis, a CNN-based model is suggested for identifying rice leaf diseases using digital photographs of the leaves. The suggested model is trained using a sizable dataset of labeled rice leaf images, which includes leaves affected with various diseases based on the types of the brown spots of the affected leaves. The model uses fully connected layers to categorize the leaves into various illness groups and convolutional layers to extract significant characteristics from the input photos.

There are a number of benefits to using digital photographs for disease detection rather than more conventional ones. First of all, because it uses a non-invasive method, diseases can be found more quickly and easily without endangering the plants. Second, the use of images enables the early detection of diseases, which can aid in stopping their spread and minimizing yield losses.

Here are the descriptions of four types of diseases that our model is going to detect and their symptoms.

Disease name	Bacterial/Fungi	Infected part of the plant	Symptoms	Solutions
Bacterial Blight [12]	Rhizoctonia solani	Panicles,upper leaf sheaths, leaf blades	There are ellipsoidal or oral spots on the leaf sheath. While the borders are brownish, the centre is somewhat green. The last step of panicked exertion is less successful.	Additionally, resistant variants are employed. By employing balanced plant nutrition, these diseases can be prevented.
Rice blast [11]	Xanthomonas oryzae pv. oryzae	Leaf collar, Panicle, panicle neck, culm nodes. culm	The borders of the spindly leaf patches are reddish or brown. Its ends are pointed. Lesions that have fully grown are 1.0–1.5 cm length and 0.3–0.5 cm broad.	Balanced amounts of plant nutrients may be utilised as a therapy for this ailment. Because dirty fields can act as hosts for pathogens, thus it's important to maintain the area around the field clean.
Brown Spot [10]	Pseudomonas syringae pv. syringae	Cleoptile, leaves, leaf sheath	Young rice plants can be identified by leaf dots. After it reaches maturity and the leaves start to senesce, it spreads.	Enhancing the soil's fertility. The early stages of this disease can also be controlled by soaking seeds in hot water prior to planting.
Tungro [9]	baciliform virus (RTBV) and rice tungro spherical virus (RTSV)	the lowest portion of the leaf blade and occasionally the leaf tip.	The plant's leaves turn yellow or orange-yellow and sometimes have rust-colored patches.	spraying insecticides

Dataset

The data set used contains 5932 number images includes four kinds of Rice leaf diseases i.e. Bacterial blight, Blast, Brown Spot and Tungro.

Later the dataset was divided into 80:20 ratio for train and test dataset

	Bacterial Blight	Blast	Brown Spot	Tungro
Train	1272	1152	1236	1044
Test	312	288	364	264

Fig.1 Images of the different rice leaf disease in the dataset

Literature Review

1. Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., and Traore, D. (2019). In Deep neural networks with transfer learning in millet crop images demonstrated that transfer learning in CNN can be used in plant disease identification. The classification accuracy has been 95% on pre trained based on feature extraction. [4]

2. Rangarajan and his colleagues classified six distinct diseases and a healthy category of the tomato crop from the picture dataset using two previously trained deep learning models, AlexNet and VGG16. For VGG16 and AlexNet, respectively, the classification accuracy was 99.24% and 96.51%. [5]

3. Amara and fellow researchers provide an automated system for classifying illnesses of banana leaves based on deep learning. To distinguish between banana Sigatoka and speckle, the authors adopt LeNet design as a deep convolutional neural network. They achieve 94.44% accuracy for grayscale photos and 98.61% accuracy for colour images.[7]

4. CNN was used by Y. Zhang and fellow researchers to identify rice diseases in his paper, Detection of rice diseases using deep CNN. On the basis of 500 photos of diseased rice and stems, the study has categorised 10 kinds of rice diseases. Experience has demonstrated that utilising machine learning and pattern recognition to diagnose illnesses in rice, CNN provides a better outcome than more conventional methods. [6]

5. By monitoring electrolyte leakage, investigating the membrane's integrity, examining the cell cycle with trypan blue staining, and assessing the depth of callose deposition with aniline blue staining, biotic stress is traditionally measured in plants.[13]

6. These techniques, however lab-based and ideally suited for examining the various factors that contribute to biotic stress, may not be practical or available to farmers. Apart from techniques like hyperspectral imaging, multispectral imaging, flourescence spectrography, and sensor combinations, thermography is employed for the detection of both abiotic and biotic stressors when non-laboratory based or contemporary sensing systems are taken into account. [14] -[16]

7. However, it is challenging to use lab-based techniques or other imaging systems at the edge. As a result, some research to address the implementation issues in networked node based agricultural vegetation monitoring have also been conducted. [17]

Methodology

In this project a deep learning-based approach for the detection of rice leaf diseases using Convolutional Neural Networks (CNN). CNNs, or Convolutional Neural Networks, are a type of deep learning model that are particularly well-suited for image classification tasks, which makes them a natural choice for detecting rice leaf disease using images of the rice leaf. CNNs are able to automatically learn and extract relevant features from the input images, which allows them to identify patterns and classify images with high accuracy.

In the case of rice leaf disease detection, CNNs can be trained to recognize the different visual characteristics of healthy rice leaves versus those with disease symptoms. For example, diseased leaves may have discoloration, deformities, or spots that are visible in the image. By training a CNN on a large dataset of labeled rice leaf images, the model can learn to detect these visual patterns and accurately classify whether a given leaf is healthy or diseased.

CNNs are particularly well-suited for image classification tasks because they use convolutional layers to scan and analyze the input image in a hierarchical manner. A succession of filters are applied to the input image by the convolutional layers, allowing the model to recognise particular elements at different degrees of abstraction, such as edges, forms, or textures. The result of the convolutional layers is then downsampled by the pooling layers, lowering the overall dimension of the data while maintaining the pertinent characteristics.

By using CNNs to detect rice leaf disease, we can develop an automated and efficient method for disease identification and management, which can ultimately benefit farmers and improve crop yields.

In this project three models were proposed and parameters like more layers, data augmentation were explored. Primarily a two convolution layer model was explored having input layer 224x224x3 and the subsequent convolution layers each having 32x32x3 shape and dropout of 20% were added. It had four dense layers and an output layer. For improving the performance of this model another convolutional layer of shape 32x32x3 was added; the result will be discussed in the next section.

In the final model, an optimizer, adam optimizer was used to optimize the model and also data augmentation was used so that the performance of our model can be seen on different images. Total of six layers were used and one hidden layer was used.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 16)      0
_________________________________________________________________
dropout (Dropout)            (None, 111, 111, 16)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 109, 109, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 54, 54, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 93312)             0
_________________________________________________________________
dense (Dense)                (None, 30)                2799390
_________________________________________________________________
dense_1 (Dense)              (None, 10)                310
_________________________________________________________________
dense_2 (Dense)              (None, 100)               1100
_________________________________________________________________
dense_3 (Dense)              (None, 133)               13433
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 536
=================================================================
Total params: 2,819,857
Trainable params: 2,819,857
Non-trainable params: 0
```

Fig. 2 Summary of primary three layer model.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 16)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 109, 109, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 52, 52, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 24, 24, 32)        9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dense (Dense)                (None, 30)                138270
_________________________________________________________________
dense_1 (Dense)              (None, 10)                310
_________________________________________________________________
dense_2 (Dense)              (None, 100)               1100
_________________________________________________________________
dense_3 (Dense)              (None, 133)               13433
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 536
=================================================================
Total params: 177,233
Trainable params: 177,233
Non-trainable params: 0
```

Fig. 3 Summary of primary three layer model with added another layer of same dimension and dropout

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential (Sequential)      (32, 256, 256, 3)         0
_________________________________________________________________
sequential_1 (Sequential)    (32, 256, 256, 3)         0
_________________________________________________________________
conv2d (Conv2D)              (32, 254, 254, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (32, 127, 127, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (32, 125, 125, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (32, 62, 62, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (32, 60, 60, 64)          36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (32, 30, 30, 64)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (32, 28, 28, 64)          36928
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (32, 14, 14, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (32, 12, 12, 64)          36928
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (32, 6, 6, 64)            0
_________________________________________________________________
conv2d_5 (Conv2D)            (32, 4, 4, 64)            36928
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (32, 2, 2, 64)            0
_________________________________________________________________
flatten (Flatten)            (32, 256)                 0
_________________________________________________________________
dense (Dense)                (32, 64)                  16448
_________________________________________________________________
dense_1 (Dense)              (32, 4)                   260
=================================================================
Total params: 183,812
Trainable params: 183,812
Non-trainable params: 0
```

Fig. 4 Summary of primary six layer model with added another layer of same dimension and dropout

Result and Application

In this section the result for the primary model proposed and the improvement suggested will be discussed

On our model in fig.2 with two convolution layers we obtained an accuracy of 89% and a loss of 35%

Following are the epoch vs loss and epoch vs accuracy graph of this model fig.2 .

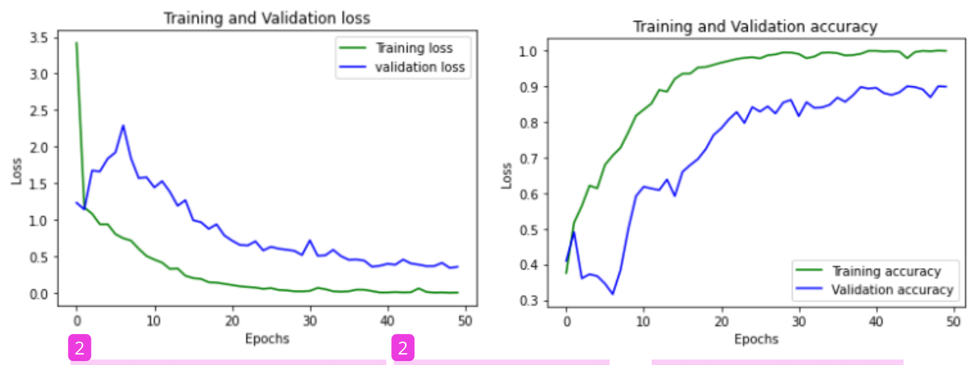


Fig.6 Graph of Loss vs Epoch and Accuracy vs Epoch on training and validation data of the model fig.2

Seeing the graph we can see that our model was not enough features so adding few more layers
By adding two more layers we have improved the accuracy of our model. We have now an accuracy of 96% and a loss of 8%

Following are the epoch vs loss and epoch vs accuracy graph of this model fig.3 .

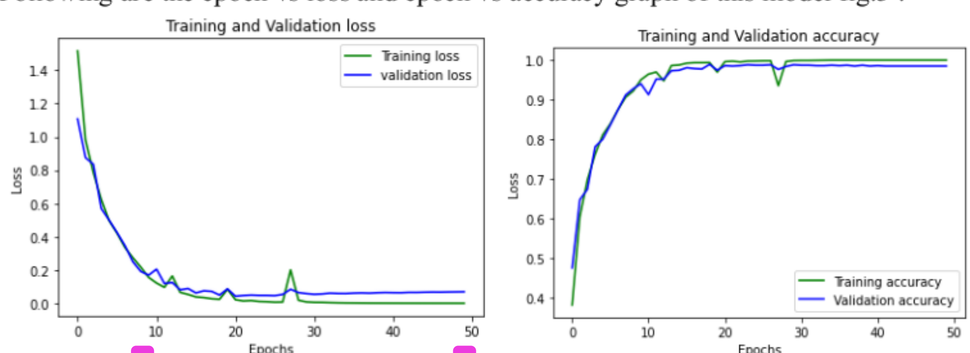


Fig.7 Graph of Loss vs Epoch and Accuracy vs Epoch on training and validation data of the model fig.3

Now in this model of fig.3 although we are having a good accuracy metric but the train and validation graph is not converging but going parallel to each other. .By adding a more convolutional layer and dense layer we can check if the two plots converge or not.

Following are the epoch vs loss and epoch vs accuracy graph of this model fig.4 .

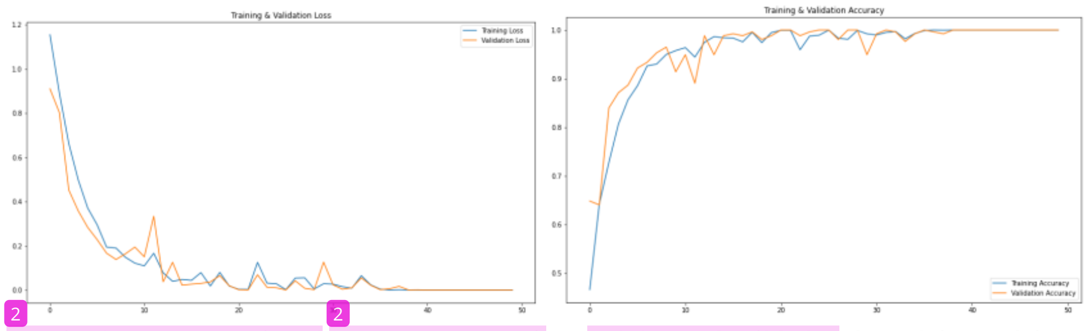


Fig.8 Graph of Loss vs Epoch and Accuracy vs Epoch on training and validation data of the model fig.4

We can see that the train and test plots almost converge giving us an accuracy of 99% and loss of 6% , so we run epochs 100 for this model.

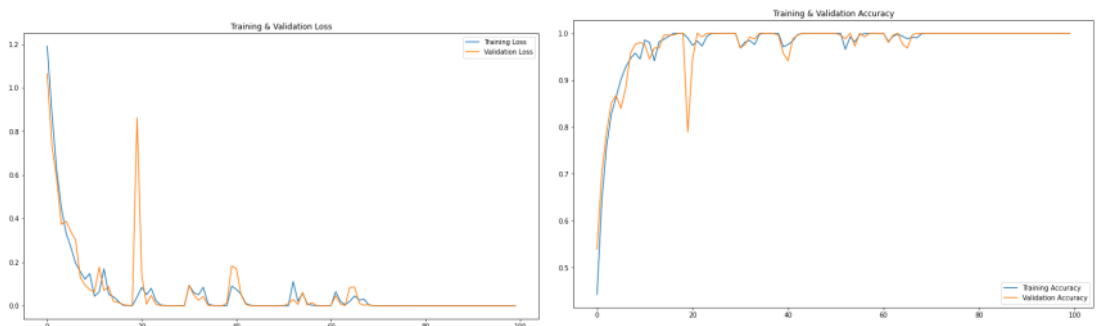
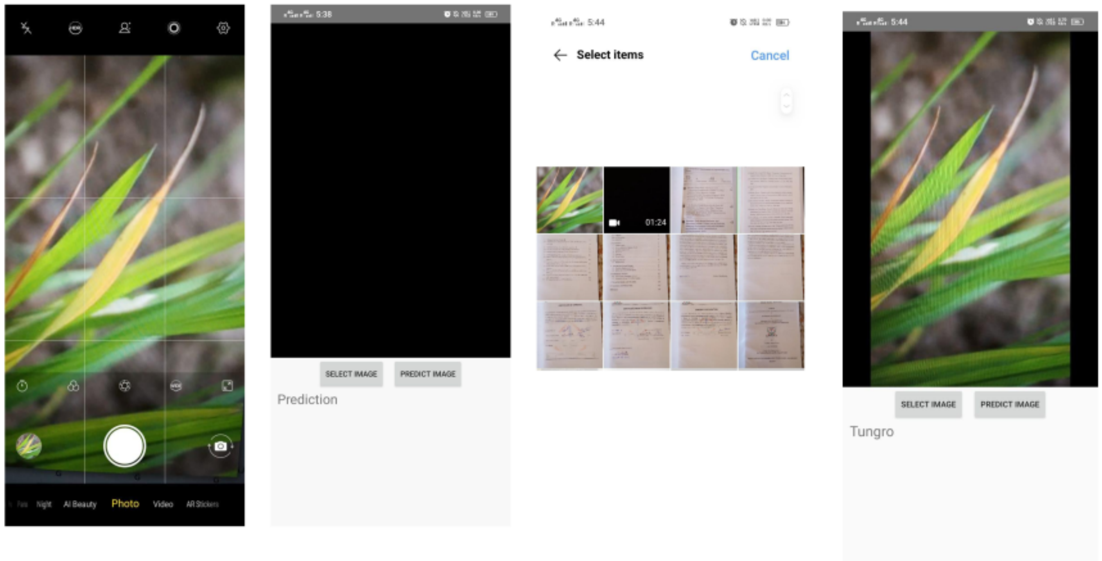


Fig.9 Graph of Loss vs Epoch and Accuracy vs Epoch on training and validation data of the model fig.4 for 100 epochs

Following is the workflow of using the android application where our project is deployed in the phone



a) b) c) d)

Fig.7 Capture image using smartphone camera (b) app launched from homescreen, (c) select image from gallery(d) the result is shown

Conclusion

In this thesis, we proposed a deep learning-based approach for the detection of rice leaf diseases using Convolutional Neural Networks (CNNs). We developed a model that is trained on a large dataset of labeled rice leaf images and achieved an accuracy of 96% and above on the test dataset. Moreover, we implemented the model in an android application that can be used by farmers for quick and easy disease detection.

The results of this thesis demonstrate the potential of deep learning techniques, specifically CNNs, for the detection of rice leaf diseases. The use of digital images for disease detection offers several advantages over traditional methods, including early disease detection, non-invasive techniques, and faster and more accurate diagnosis.

The android application developed in this project provides a user-friendly tool for farmers to identify and manage rice leaf diseases, potentially improving rice production and reducing yield losses. Moreover, the proposed model can be utilized by researchers and agricultural experts for the identification and classification of various rice leaf diseases.

In conclusion, this thesis provides a reliable and efficient tool for the detection of rice leaf diseases, which can potentially contribute to the sustainable agriculture and food security of the Indian subcontinent. The proposed CNN-based model and android application offer a valuable contribution to the field of deep learning and agriculture, demonstrating the potential of technology in addressing the challenges faced by farmers and researchers in disease management.

Future work

The following improvement can be implemented:

1. Dataset: It would be helpful to collect a larger and more diverse dataset to train and validate the model. Including more variations in the lighting, camera angle, and image resolution can improve the robustness and accuracy of the model.
2. Hyperparameters tuning: The model's performance can be optimized by adjusting the hyperparameters, such as learning rate, batch size, and optimizer, to improve the accuracy of the model.
3. Transfer learning: A Pre-trained CNN model, such as ResNet or VGGNet, and fine-tune it for rice leaf disease classification, which can help improve the model's performance with less data and fewer training iterations.
4. Multiclass classification: Expanding the model to include the classification of multiple diseases can be helpful for farmers in identifying the specific type of disease that affects their rice crops. Here, we are only able to classify four types of diseases, further data of other types of rice leaf disease images can be used for more multiclass classification.
5. Integration with other technologies: Integrating the model with other technologies, such as IoT sensors or drones, can provide more accurate and real-time data for disease detection and management for mechanized farming over large farm fields.

Overall, these improvements can help further enhance the accuracy, efficiency, and usability of the model, making it more effective for farmers and researchers in detecting and managing rice leaf diseases.

Bibliography

[1] Deng R, Tao M, Xing H, Yang X, Liu C, Liao K and Qi L (2021) Automatic Diagnosis of Rice Diseases Using Deep Learning. Front. Plant Sci. 12:701038. doi: 10.3389/fpls.2021.701038

[2] sethy, prabira Kumar (2020), “Rice Leaf Disease Image Samples”, Mendeley Data, V1, doi:10.17632/fwcj7stb8r.1 [**dataset**]

Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. Computers and Electronics in Agriculture, 175, 105527. doi:10.1016/j.compag.2020.105527.

[3] Udayananda, Viran & Shyalika, Chathurangi & Pathirage, Nandana. (2022). Rice plant disease diagnosing using machine learning techniques: a comprehensive review. SN Applied Sciences. 4. 10.1007/s42452-022-05194-7.

[4] Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., and Traore, D. (2019). Deep neural networks with transfer learning in millet crop images. Comput. Ind. 108, 115–120. doi:10.1016/j.compind.2019.02.003

[5] Aravind Krishnaswamy Rangarajan, Raja Purushothaman, Aniirudh Ramesh, Tomato crop disease classification using pre-trained deep learning algorithm, Procedia Computer Science, https://doi.org/10.1016/j.procs.2018.07.070

[6] Yang Lu, Shujuan Yi, Nianyin Zeng, Yurong Liu, Yong Zhang, Identification of rice diseases using deep convolutional neural networks, https://doi.org/10.1016/j.neucom.2017.06.023.

[7] Amara, J., Bouaziz, B. & Algergawy, A., (2017). A Deep Learning-based Approach for Banana Leaf Diseases Classification. In: Mitschang, B., Nicklas, D., Leymann, F., Schöning, H., Herschel, M., Teubner, J., Härder, T., Kopp, O. & Wieland, M. (Hrsg.), Datenbanksysteme für Business, Technologie und Web (BTW 2017) - Workshop Band. Bonn: Gesellschaft für Informatik e.V.. (S. 79-88). https://dl.gi.de/handle/20.500.12116/944

[8] Gong, A., Yu, J., He, Y., and Qiu, Z. (2013). Citrus yield estimation based on images processed by an Android mobile phone. Biosyst. Eng. 115, 162–170. doi: 10.1016/j.biosystemseng.2013.03.009

[9] Article title :Tungro Disease
URL http://www.agritech.tnau.ac.in/expert_system/paddy/cpdistungro.html

[10] Article title: Brown Spot (Helminthosporium oryzae)
URL https://agritech.tnau.ac.in/expert_system/paddy/cpdisbrownspot.html

[11] URL https://agritech.tnau.ac.in/crop_protection/rice_diseases/rice_1.html
Website title TNAU Agritech Portal :: Crop Protection

[12] Article title: Bacterial leaf blight (Xanthomonas oryzae pv. Oryzae) URL http://www.agritech.tnau.ac.in/expert_system/paddy/cpdisblb.html

[13] M. Bach-Pages and G. M. Preston, “Methods to quantify biotic-induced stress in plants,” in Host-Pathogen Interactions. New York, NY, USA: Humana Press, 2018, pp. 241–255.

[14] A. V. Zubler and J.-Y. Yoon, “Proximal methods for plant stress detection using optical sensors and machine learning,” Biosensors, vol. 10, no. 12, p. 193, Nov. 2020.

[15] N. Mastrodimos, D. Lentzou, C. Templalexis, D. I. Tsitsigiannis, and G. Xanthopoulos, “Development of thermography methodology for early diagnosis of fungal infection in table grapes: The case of aspergillus carbonarius,” Comput. Electron. Agricult., vol. 165, Oct. 2019, Art. no. 104972.

[16] E.-C. Oerke, P. Fröhling, and U. Steiner, “Thermographic assessment of scab disease on apple leaves,” Precis. Agricult., vol. 12, no. 5, pp. 699–715, Oct. 2011.

[17] P. Pal, R. P. Sharma, S. Tripathi, C. Kumar, and D. Ramesh, “2.4 GHz RF received signal strength based node separation in WSN monitoring infrastructure for millet and Rice vegetation,” IEEE Sensors J., vol. 21, no. 16, pp. 18298–18306, Aug. 2021.

Plag report

ORIGINALITY REPORT

4%	2%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	G. K. V. L. Udayananda, Chathurangi Shyalika, P. P. N. V. Kumara. "Rice plant disease diagnosing using machine learning techniques: a comprehensive review", SN Applied Sciences, 2022 Publication	2%
2	Ram Hari Regmi, Arun K. Timalsina. "Risk Management in customs using Deep Neural Network", 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018 Publication	2%

Exclude quotes	Off	Exclude matches	< 40 words
Exclude bibliography	On		