ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

MITIGATION OF CROP DISEASE DETECTION

RAJESH BANALA[1] N. KARTHIK REDDY[2] K. MADHUSUDHAN KUMAR[3] G. NEETHA REDDY[4] D. SUMALINI GOUD[5]

Assistant Professor (CSE-DS) TKR College Of Engineering and Technology, India rajesh.banala@gmail.com
IV Final Year of CSE (DS) TKR College of Engineering and Technology Telangana, India nagaralakarthikreddy@gmail.com
IV Final Year of CSE (DS) TKR College of Engineering and Technology Telangana, India ms16032003@gmail.com
IV Final Year of CSE (DS) TKR College of Engineering and Technology Telangana, India neethareddy1204@gmail.com
IV Final Year of CSE (DS) TKR College of Engineering and Technology Telangana, India smalini9800@gmail.com

ABSTRACT

Agriculture stands as a foundation of financial improvement, providing food for over 58% of the worldwide populace. However, customary cultivating strategies confront challenges, outstandingly in disease detection, prompting the selection of progressed strategies. This study digs into the combination of image processing and deep learning techniques for automated plant disease discovery. Our strategy envelops image acquisition, pre-processing, segmentation, feature extraction, and classification, with accentuation on utilizing Convolutional Neural Network (CNN) models. Outstandingly, our proposed architecture integrates parallel attention mechanism modules and residual blocks, increasing accuracy and real-time performance. Through broad experimentation, our CNN show accomplishes an amazing 98.17% accuracy in crop disease image classification. These results emphasize the potential of CNNs in real-world applications and catalyze encourage exploration in bug and plant disease recognition.

Keywords: CNN, plant disease, deep learning, real-world

1. INTRODUCTION

1.1 Motivation

Agriculture, the backbone of the economy, heavily depends on different production components like bugs, fertilizers, and water. This project aims to survey the effect of plant infections, bug control, and suggest fundamental steps for agricultural victory. Recognizing plant maladies and controlling pests are vital assignments for agricultural exactness. Automatic recognition of plant disease using portable devices can cultivate a way better development environment for crops at a low cost, subsequently expanding agricultural production.

1.2 Problem Definition

Plant diseases, inadequate pest control measures and imprecise crop predictions are some of the multiple factors that lead to crop failures. It is very important to have accurate crop prediction which can be obtained through observations and quality assessments. By applying fertilizers only when a disease has been identified, fewer diseases may occur in the coming years. Effective measures for controlling diseases will need improved identification as well as better management for plant diseases.

1.3 Limitations of existing system

While existing systems aid in forecasting plant diseases, they repeatedly lack accurateness. For case, a model achieved an 86.1% accuracy after training but fell short of expectances. It employed 11 learnable layers, including eight convolutional and three fully connected layers within the PestDetNet frame and inception v3. Although various preprocessing strategies and transfer learning were employed, the accuracy didn't meet the desired range.

1.4 Proposed system

The proposition is for a system that tries to improve the accuracy of plant disease detection and pest control using Deep Learning (DL). This platform gives precise and vital agricultural technology proposals by means of an integrated platform. This interface identifies all plant diseases in one location while offering fast predictions through FastAPI. The proposed model ensures accuracy as well as efficiency.

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

2. REVIEW OF LITERATURE

The problem of plant disease detection is addressed by S. Khirade et Al in 2015 using digital image processing techniques and back propagation neural network (BPNN). Different approaches for identifying plant diseases using leaf images have been examined by authors who have further extracted features like color, texture, morphology, edges etc, to classify the plant diseases. For that purpose BPNN is used for classification means to detect the plant disease. Akhtar et al. suggested that SVM, k-NN, RNN(recurrent neural network), Naïve Bayes, decision tree and wavelet-based plant disease detection and gray-level co-occurrence matrix could be used to train features. Fathima et al. developed a method of detecting diseases from hyperspectral measurements . Karthik et al. viewed the severity of diseases in leaves as an issue for recommendation. Semary et al.'s automated approach employed texture and color features with SVM for classification purposes. By using Gabor wavelet features and GLCM (gray-level co-occurrence matrix), weighted KNN was trained in ref. Padol et al.in their work on detecting disease through texture and color features which were then utilized to classify the infected area with support from k-means clustering algorithm and SVM classifier. This was used in ref., where they introduced using k-means for disease detection or classification process. In another study by Mehra et al., it is shown how 'k-means' can be applied to identify the presence of fungal infection in leaves . Clustering algorithm's main challenges are determining cluster count and adjusting parameters for separation between clusters. Multiple problems were solved using scale-invariant features in the image.

3. DESIGN

The proposed System architecture comprises of

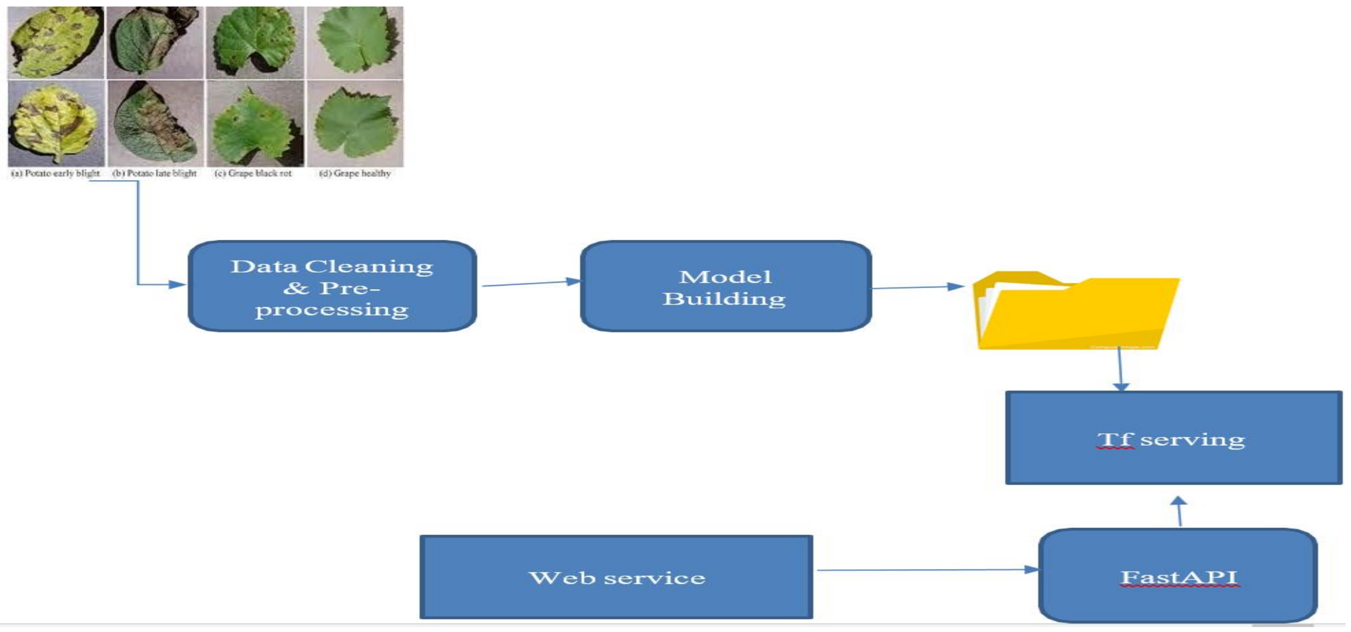


Fig 1: Architecture Diagram

data acquisition from a huge dataset, processing at different convolutional layers and then the classification of plant diseases which declares if the plant image is of a healthy class or diseased class.

3.1 DFDs and UML diagram

Data Flow Diagrams (DFDs) illustrate the steps which data gos through starting from input and ending in output prediction.

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

3.1.1 Training Dataset:

This step involves gathering and preparing a training dataset. Such a dataset may contain images of plants that are classified by their health status, such as healthy or infected. These inputs form part of training data for CNN model. This information can be obtained from different sources including field surveys, databases or image repositories.

3.1.2 CNN (Convolutional Neural Network):

The features are extracted from the input images by the CNN model using convolutional operations that aids in classifying them as to whether they are healthy or diseased plants among others (e.g., healthy or diseased plants). The model adjusts its parameters towards minimizing the difference between predicted classes and true classifications during training process.

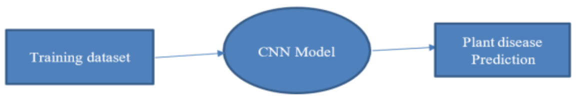


Fig 2: Data flow Diagram

3.1.3 Plant disease detection

When the dataset is trained with the CNN model, it could be used to detect plant diseases once deployed for that purpose. In this step, newly captured plant images on fields or gardens are fed into an already trained CNN model.

3.2 Use Case Diagram

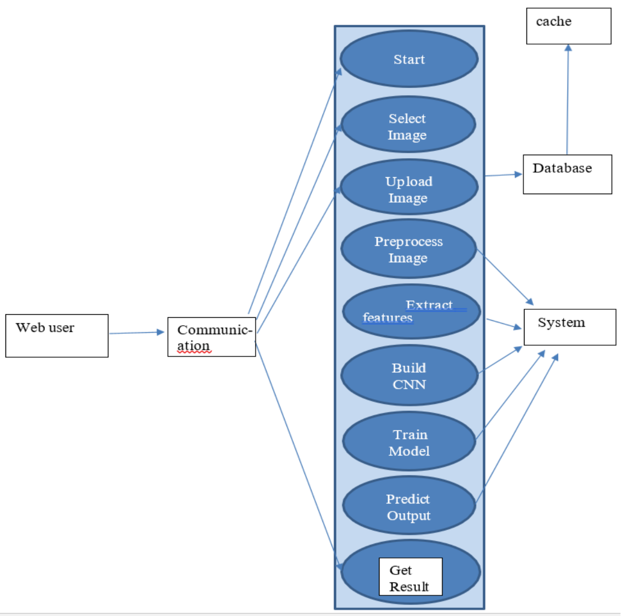


Fig 3: Use Case Diagram

A Use Case Diagram is a diagram that shows the actors of the system and the use cases. From an actor's perspective a use case represents functions or groups of functions. Entities interacting with the modeled system using its functionalities are called actors. Such actors may be human users, other hardware devices, or software systems. Use cases are shown as ellipses. They can also be connected by dotted lines that indicate which actor uses them (and who started it).

3.3 Sequential Diagram

In this sequential diagram above, there is a representation on how disease can be diagnosed in a simplified manner by utilizing machine learning. The algorithm has to do several tasks that follow one another in sequence. User Image Upload: The user uploads an image of the K-user plant.

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

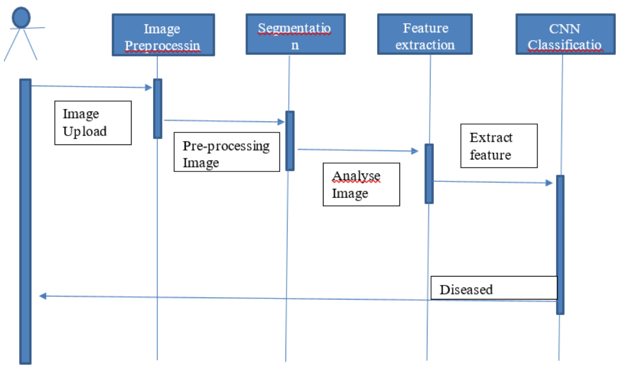


Fig 4: Sequential Diagram

User Image Upload: The user uploads an image of a K-user plant.

Image Preprocessing: This is done to resize the image and normalize its pixel values in order to enhance performance and consistency.

Segmentation: In this case, the algorithm applies some segmentation techniques to separate various structures within the plant such as stem, leaves, flower etc.

Preprocessing Image: The other thing that is done next involves going further and processing each segmented part of the plant so as to make feature extraction more efficient.

Feature Extraction: Distinctive features or patterns are recognized by the algorithm from the preprocessed images that will be used as input for the CNN model.

Analysis Image: This processes extracted features by classifying them on a CNN model trained using diseased and healthy k user plants images to determine if they are healthy or not.

CNN Classification: To classify this, we have a convolutional neural network (CNN) model that operates on the features based on what was learned during training.

Extract Feature: These relevant features are gotten from preprocessing image which are needed for final classification decision making process

Diseased or Healthy: Based on given image, this algorithm predicts whether K-user plant is healthy or sick.

4. Methodology

Convolutional Neural Networks(CNN) The architecture is shown in Figure, As you can see that the object can be of varying sizes. The solution to this problem is to create an image pyramid, which scales the image. The idea behind resizing the image at multiple scales is that we assume our chosen window size will enclose the object in one of these resized images entirely. In most cases, the image is down-sampled or reduced up to when certain condition typically a minimum size has been reached. On each of these images a fixed size window detector is run. It's common to have as many as 64 levels on such pyramids. Now, all these windows are input into a classifier for recognition of objects of interest. This will help us address both scale and position issues.

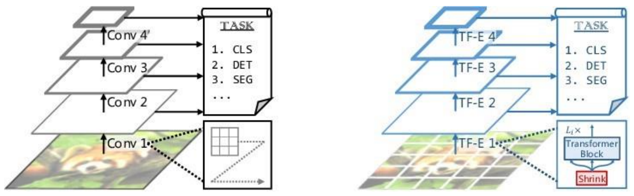


Fig 5: Image Precision

Another challenge was to generate an input of a fixed size for the fully connected layers of the CNN so, SPP adds one cleverer move. It does spatial pooling after the last convolutional layer unlike max-pooling that is traditionally used. An SPP layer divides any arbitrary region into a fixed number of bins and maximum pooling is done on each bin. This gives a constant-size vector since the number of bins remains unchanged as seen in the figure below.

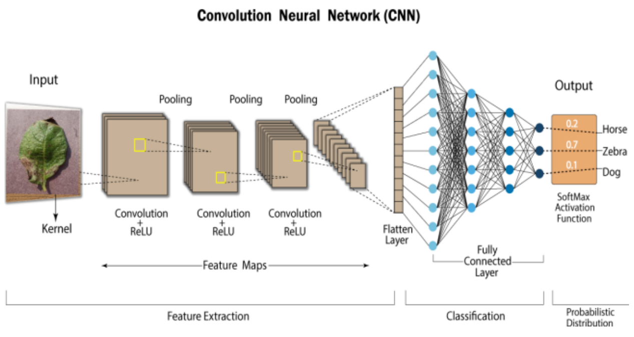


Fig 6: CNN Architecture

CNN introduces anchor boxes to deal with disparities in object scale and aspect ratio. The original paper uses 3 types of anchor boxes at each position for scales 128×128, 256×256 and 512×512. In the same way, it takes three aspect ratios: 1:1, 2:1 and 1:2. Therefore, we have a total of nine boxes per location that RPN estimates as either background or foreground. We make use of bounding box regression at each location to improve the anchor boxes. RPN provides output bounding boxes with their predicted sizes and class probabilities. CNN can also receive varying size bounding boxes which are forward passed through it. That's why CNN has been one of the most

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

accurate object detection algorithms. Here is a quick comparison between various versions of CNN.

1.Input Layers: This is the layer where we put our inputs in our model. The number of neurons in this layer equals the total features in our data (pixels' number for example, an image).

2.Hidden Layer: The input from the Input layer is then fed into the hidden layer. Depending on our model and data size there can be several hidden layers. Each hidden layer has different numbers of neurons that are usually greater than the features' number. For each previous layer's output, its next output is got by multiplying them with learnable weights and adding learnable biases and activation function which makes the network non-linear.

3.Output Layer: After passing through hidden layers, it goes to a logistic function like sigmoid or softmax which changes the output score for each class into a probability score for each class.

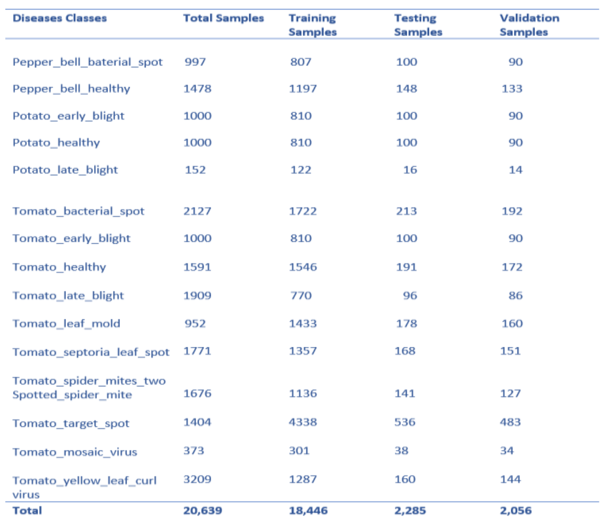
5. EXPERIMENTS

A GPU NVIDIA workstation was the baseline system used to evaluate our experiments. The operating environment was Windows 11, GDDR5 graphic memory type, Core i5 10th generation, 8 GB RAM. Anaconda3, Keras, TENSORFLOW, Numpy, and matplotlib libraries were used in implementing the software.tf serving simple libraries specifically made for deep learning implementations with reduced memory usage and faster execution. Both these libraries were designed by NVIDIA to work in the Theano backend. Fastapi supports both academic and commercial project development and supports Linux, Windows, Mac OS, iOS, Python, Java and Android interfaces. For each experiment in this work training accuracy and testing accuracy have been computed.

The losses obtained during the testing and training phases were calculated for each model. With intent of speeding up CNN learning rate of transfer learning models trained on PlantVillage dataset as CNN-using pre-trained models that were trained using Plant Village dataset with 1.2 GB images image categories.

5.1 Description of Dataset

The PlantVillage Dataset is one dataset available to the public that has various categories of plant diseases. The PlantVillage Dataset is made up of 15 classes with 20,645 images. In our experimentation, we divided the dataset into training samples, testing samples as well as validation samples. For pre-trained models, they were trained on 80% of the PlantVillage dataset and used the remaining 20% for validation and testing. Further, there was a total of 20645 samples available for the plant classes which included 18032 for training, 1376 for validation and 1654 for testing. All these train, test and validation sets include all the fifteen different plants' diseases. The data set size details can be found in table 1 below:



Diseases Classes	Total Samples	Training Samples	Testing Samples	Validation Samples
Pepper_bell_baterial_spot	997	807	100	90
Pepper_bell_healthy	1478	1197	148	133
Potato_early_blight	1000	810	100	90
Potato_healthy	1000	810	100	90
Potato_late_blight	152	122	16	14
Tomato_bacterial_spot	2127	1722	213	192
Tomato_early_blight	1000	810	100	90
Tomato_healthy	1591	1546	191	172
Tomato_late_blight	1909	770	96	86
Tomato_leaf_mold	952	1433	178	160
Tomato_septoria_leaf_spot	1771	1357	168	151
Tomato_spider_mites_two Spotted_spider_mite	1676	1136	141	127
Tomato_target_spot	1404	4338	536	483
Tomato_mosaic_virus	373	301	38	34
Tomato_yellow_leaf_curl virus	3209	1287	160	144
Total	**20,639**	**18,446**	**2,285**	**2,056**

Table 1: Details of samples

5.2 Preprocessing And Augmentation

The dataset had 15 categories with 15 diseases and 3 species of crops. We selected the color pictures from the PlantVillage dataset for our experiment as they go well on transfer learning models. In order to standardize the format, the images were downscaled to 256 × 256 pixels since we used various pre-trained network models that need different input sizes. For CNN, the input size is (224 × 224 × 3) while for Inception V4, images had height:299,width:299 and channel width:3 as inputs. However, although this data set includes more than twenty thousand different crop disease images, it resembles the real-life images taken by farmers using various image acquisition methods such Kinect sensors, high definition cameras or smart phones. Moreover, a dataset of this magnitude is prone to overfitting thus in order to mitigate this overfitting regularization techniques like data augmentation after preprocessing were introduced. The augmentation steps applied on preprocessed images involved clockwise and

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

anticlockwise rotation; horizontal and vertical flipping; zoom intensity; rescaling etc. During training process these are not duplicated but augmented so that no physical copies of the augmented images are kept but only used temporarily during the whole exercise. This augmentation technique not only prevents the model from overfitting and model loss but also increases the robustness of the model so that, when the model is used to classify real-life plant disease images, it can classify them with better accuracy.

5.3 Fine-Tuning of Hyperparameters in Pre-Trained Models

Some of the pros and cons of using this approach include: it is less computationally intensive since it does not require retraining the entire model. In the beginning, different standardizations were done on various hyperparameters for different pre-trained models. The details of the hyperparameter tuning are listed in Table2.

Hyperparameters	Epochs
Dropout	0.5
Epochs	30
Activation	ReLU
Regularization	Batch normalization
Optimizer	Stochastic gradient descent
Learning rate	0.001
Output classes	38

Table 2: Details of hyperparameter

5.4 Network Architecture Model

The pre-trained network models were selected based on their suitability for plant disease classification task. Different filter sizes are present in each of the networks used to extract specific features from feature maps. Filters play a crucial part in the process of extracting features. Also, each filter when convolved with the input brings out different features from it and depending on the filters' peculiar values, the feature extraction is achieved within the maps. In our experiments, we utilized an actual pre-trained network model that had real combinations of convolutional layers and filters sizes as specified by each network model.

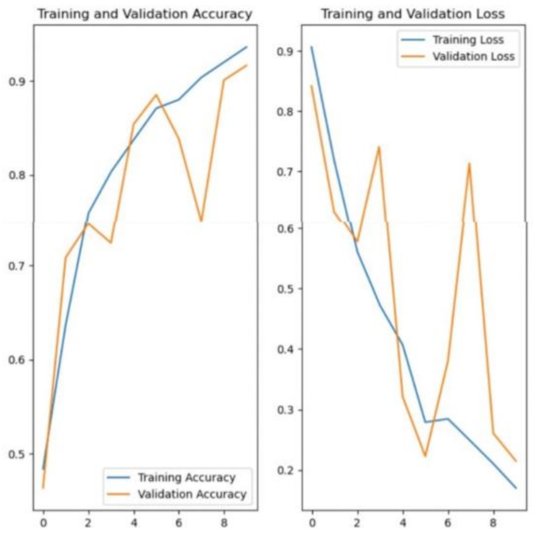
5.5 Results

The purpose of this study is to employ transfer learning through state-of-the-art deep learning models for disease diagnosis in plants. For further training, the publicly accessible PlantVillage dataset was used to re-train pre-trained deep CNN networks that were initially trained with the ImageNet dataset. Every model was standardized by experiment using some specific parameters such as 0.01 learning rate, 0.5 dropout and fifteen output classes. The specificity, sensitivity and F1 score are subjected to a performance analysis on the various pre-trained models.

sensitivity (recall) = (True positive + False Negative)

True Positive

Specificity is a degree of the extent of really unhealthy plants predicted to be unhealthy (true negative) and the really unhealthy leaves predicted to be healthy(false positive)



specificity = True Negative

(True Negative + False positive)

Results with those from state-of-the-art studies from the literature that utilized exchange learning models. We considered state-of the-art studies from the literature that experimented on the PlantVillage dataset. It was observed from the examination that our work considered more plant disease classes. Further, our fine-tuned, pre-trained model accomplished the best precision of 99.81%.

6. ACKNOWLEDGMENTS

We extend our sincere gratitude to the reviewers and editors for their invaluable feedback and constructive criticism, which significantly contributed to enhancing the quality of this manuscript. Their thorough review and insightful comments have greatly strengthened the clarity and coherence of our work.

Furthermore, we would like to express our gratitude to our respective institutes for their support and resources, which facilitated the completion of this research.

Thank you for your unwavering support and guidance throughout the publication process.

7. REFERENCES

1.Ferentinos, K.P. Plant infection detection and identification using deep learning algorithms for machine vision systems.Comput. Electron.Agric.2018, 145,311–318.

2.Lee, S.H.; Goëau, H.; Bonnet, P.; Joly, A. New looks at deep learning based plant disease characterization. Comput.Electron.Agric.2020,170 ,105220.

3.Zhong, Y.; Zhao, M.Research on apple leaf disease recognition with deep learning approach. Comput. Electron. Agric.,168 ,105146 .

4.Zhang,Y.; Song,C.;Zhang,D.Deep Learning-Based Object Detection Improvement of Tomato Disease.IEEE Access 2020,8 ,56607–56614.

5.Chen,J.,Chen,J.,Zhang,D.,Sun,Y.&Nanehkaran,Y.Use of deep transfer learning in image-based plant disease identification.Comput.Electron.Agric.,173 ,105393.

6.Shrivastava,V.K.; Pradhan,M.K.andMinz,S.Rice plant disease classification via transferring knowledge from pre-trained convolutional neural networks.International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences Volume XLII-2/W13 ISPRS 2019 3(42),631-635 .

7.Shrivastava,V.K.andPradhan,M.K.Machine learning approaches for rice plant disease classification based on color features.Journal of Plant Pathology (2021) Vol.. 103 s17-s26

8.He,K., Zhang,X.,Ren,S.&Sun,J.Deep residual learning for recognition of images.Proceedings of the IEEE Confere nce on Computer Vision and Pattern Recognition (CVPR), Las Vegas,NV,(USA),June 27-30,(2016).

9. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556.

10. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA; July 21-26 2017; pp.4700–4708.