IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13374

Recognition and Classification of Paddy Leaf Disease using CNN

Divyata J[1], Amrutha[2], Harshitha[3], Likhitha[4], Pavana[5]

Student, Department of Computer Science & Engineering, Mangalore Institute of Technology and Engineering, Moodabidre, India[1]

Assistant Professor, Department of Computer Science & Engineering, Mangalore Institute of Technology and Engineering, Moodabidre, India [2]

Student, Department of Computer Science & Engineering, Mangalore Institute of Technology and Engineering, Moodabidre, India[3,4,5]

Abstract: Paddy leaf diseases pose a significant threat to global rice production, impacting food security and economic stability. This study explores the application of machine learning, specifically convolutional neural networks (CNNs), for the automated recognition and classification of paddy leaf diseases. The proposed CNN model analyzes leaf images to detect common diseases such as brown spot, leaf blast, and leaf blight. Leveraging advanced image processing techniques, the system achieves high accuracy in disease identification, enabling timely interventions to mitigate crop losses. Key aspects of the project include dataset preparation, model training, and performance evaluation. Through this research, we contribute to the advancement of precision agriculture and sustainable crop management practices.

Keywords: Paddy leaf diseases, Machine learning, Convolutional neural networks, Automated recognition, Crop management.

I. INTRODUCTION

This project revolves around the creation of a machine learning system designed to swiftly and accurately identify and categorize paddy leaf diseases within rice crops. Paddy leaf diseases, caused by a spectrum of pathogens, have long been a recurring menace to rice crop health. The repercussions of these diseases extend beyond mere agricultural concerns, impacting global food security and the livelihoods of countless farmers.

In today's rapidly evolving technological landscape, the development of robust machine learning solutions holds the promise of addressing critical challenges across various domains. This project is poised to contribute to the field of agriculture by harnessing the power of advanced technology to enable the accurate detection and classification of paddy leaf diseases.

At the core of this endeavor lies the utilization of Convolutional Neural Networks (CNNs). CNNs are a class of deep learning models that excel in image recognition and classification tasks. By harnessing the capabilities of CNNs, our project aims to provide a precise and efficient solution for the diagnosis of various paddy leaf diseases. This technology empowers us to unlock new possibilities in the realm of agricultural disease management.

The proposed system will be trained on a large dataset of paddy leaf images. The network will learn to extract meaningful features from the images and categorize them to different diseases through a series of convolutional and pooling layers. Additionally, techniques like data augmentation and transfer learning may be employed to enhance the model's generalization capabilities and overcome data scarcity issues.

II. OBJECTIVE

The main purpose of this project is to develop an efficient system for the early detection and management of paddy leaf diseases using Convolutional Neural Networks (CNNs). By leveraging CNNs, we aim to automate the process of disease recognition and classification to save time and effort for farmers. This automation will facilitate prompt identification of disease symptoms in paddy leaves, enabling timely interventions to prevent the spread of diseases and minimize crop losses. Through the development of a user-friendly interface, we strive to ensure accessibility and ease of use for farmers and agricultural stakeholders, promoting seamless integration into existing agricultural workflows.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13374

III. PROBLEM STATEMENT

Although rice plays a vital role as a global staple food, paddy leaf diseases are still a major threat to rice crops. These diseases, caused by various Pathogenic organisms, lead to loss of yield, financial setbacks for farmers, and global food security concerns. Timely recognition and identification of disease are challenging because of naked eye prediction, which can be inaccurate. Furthermore, the lack of reliable, cost-effective, and efficient disease recognition tools hinders effective control measures.

For detecting the leaf diseases, the traditional methods are human vision-based approaches or asking the expert advice. In these cases, seeking the expert advice is time consuming and very expensive for the farmers. The human vision-based methods cause many disadvantages.

The accuracy and precision of human vision approach is dependent on the eyesight of the person's expertise or expert hired. Machine learning based method enables to identify the types of diseases, make the right decision and to select proper treatment.

One of the advantages of using machine learning based method is that it performs tasks more consistently than human experts. Therefore, to overcome the drawbacks of conventional methods there is a need for a new machine learning based classification approach. Very few recent developments were recorded in the field of plant leaf disease detection using machine learning approach and that too for the paddy leaf disease detection and classification is very limited.

IV. LITERATURE REVIEW

Swathika R, N Radha, K Sowmya, "Disease Identification in paddy leaves using CNN based Deep Learning" [1].

Each of the collected images were converted into grayscale in order to process faster. The CNN is created with 3 convolution layers. Here Contour detection module is used where contour function defines contour in the image and estimates affected area by considering contours based on size.

S. Ramesh *, D. Vydeki Electronics and Communication Engineering Department, VIT University, Chennai 600127, "Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm" [2].

Their research showed that they collected datasets from farm field and dataset contains images having leaves with various degree of disease spread. For the image segmentation K- means clustering is used and the diseased portion is obtained by clustering of images. This is implemented in Python platform version 3.6.

Radhika Deshmukh, Manjusha Deshmukh, "Detection of paddy leaf diseases" [3]

Images are taken from the internet and some samples were manually clicked and stored in jpg format. The K-Means clustering algorithm is used to classify objects based on a set of features into K number of classes using squared Euclidean distance. The features from K-means is sent to the neural network to detect the paddy leaf disease.

S Pavithra, A Priyadharshini, V Praveena, T Monika, "Paddy leaf disease detection using SVM classifier" [4]

Image clipping is performed to get the infected region and to remove the noise then image smoothing is carried out using smoothing filter. The segmentation is done using Otsu' method, k-means clustering. The features extracted are color, texture, morphology, edges etc. The backpropagation algorithm is used for the final classification.

M. Ramkumar Raja a,*, Jayaraj V b, Francis H Shajin c, E.M. Roopa Devi d ,"Radial basis function Neural Network optimized with Salp Swarm algorithm espoused paddy leaf disease classification" [5]

The paddy leaf disease classification is done with the help of RBFNN optimized with SSA for classifying the paddy leaf disease on Gaussian density function. The SSA is used for enhancing RBFNN classifier to discover optimum parameters.

Basavaraj S. Anami, Naveen N. Malvade, Surendra Palaiah, "Deep Learning Approach for recognition and classification of yield affecting paddy crop stresses using field images" [6]

In this paper the effort is to automate the recognition and classification of paddy crop stresses and demonstrate using deep learning techniques. The pre-trained deep learning model VGG-16 has been used in the classification. The future scope is to compare the VGG-16 with the ResNet, Inception-v3 to get better results.

V. METHODOLOGY

There are numerous methods involved in building this project, from collecting datasets and converting them into masks to loading the masked images into the model and then training them to give accurate results. We chose Convolutional Neural Network (CNN) which is a deep learning algorithm mainly used for classification. Then the trained model is integrated to the user interface that we built to give real time experience for the users.

A. System Architecture

System architecture defines the structure of a software system, including its components, interactions, and deployment. Components are the building blocks, each with specific functions. Interactions detail how components communicate,

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13374

while layers organize functionality. Deployment considers hardware, networking, and scalability. Scalability and performance address system responsiveness. Security ensures protection against threats, and reliability ensures availability. Integration and extensibility support seamless connections and future enhancements.

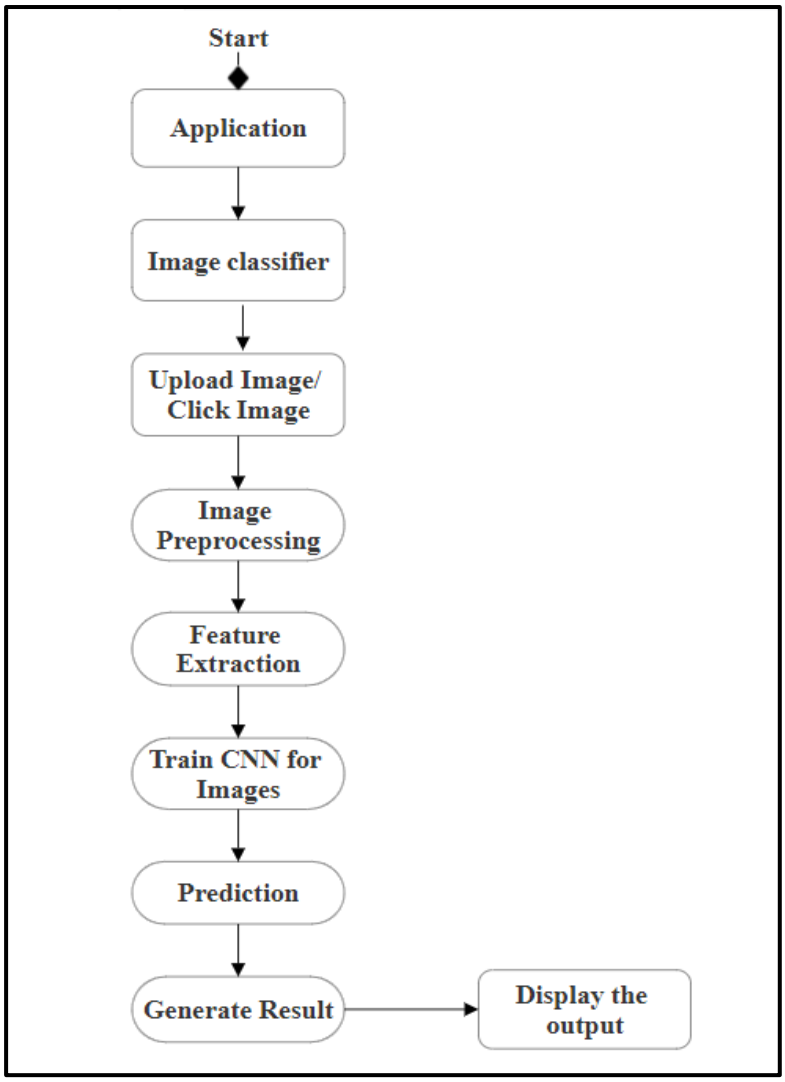

Fig. 1 System Architecture

B. Datasets

The datasets for paddy leaf were downloaded from Kaggle with over 1000+ images. The datasets were already segregated into training, testing and validation. Some of them were incorrectly labelled so we had to clean the datasets.

TABLE I LIST OF DATASETS COLLECTED

Leaf Samples	Datasets
Brown Spot	195
Leaf Blast	180
Light Blight	249
Leaf Smut	245
Healthy Leaf	246

DOI: 10.17148/IJARCCE.2024.13374

C. Convolutional Neural Network (CNN)

CNN stands for Convolutional Neural Network, a powerful deep learning technique widely utilized for tasks like image classification, object recognition, and various image and video analysis applications. CNNs are designed to autonomously learn and extract relevant information from images through a series of layers including convolutional layers, pooling layers, fully connected layers, and an output layer. The convolutional layers utilize convolution, a mathematical operation, to extract key features such as edges, textures, and shapes from input images. Pooling layers are then employed to downsample the feature maps generated by convolutional layers, reducing computational complexity and preventing overfitting. Fully connected layers perform the final classification or regression based on the extracted features. CNNs have revolutionized computer vision, exhibiting exceptional performance in tasks like image classification, object recognition, and segmentation.

D. Implementation

To implement this project, we first had to mask all the images. Masking is a technique commonly used in image processing to selectively manipulate or extract specific parts of an image based on certain criteria. To do so, we had to import the required modules such as OpenCV, NumPy and PIL. Then each image is read by cv2.imread() function and then it converts from BGR to HSV. Define upper and lower threshold values in HSV format to create a mask that isolates the desired portion of the image. A mask was generated using these values using cv2.inRange() function and then we saved all these files with .png extension. Overall, this process effectively removes the background from the original images, leaving only the foreground objects or elements of interest.

After masking, we need to train the CNN model, for that we had to import some libraries into our Project. It includes TensorFlow which is used to build the CNN model in our Project. We also imported matplotlib library, which is used for creating visualizations. Next, we imported the required layers from the TensorFlow Keras API for building a CNN model. These layers include Conv2D, MaxPooling2D, Dropout, Flatten, and Dense. We also imported the Sequential model from the TensorFlow Keras API, which is a linear stack of layers. We use pathlib module, which provides an object-oriented way to manipulate the file system paths in Python through which we import the datasets into the project. We used 'ImageDataGenerator' class from Keras to prepare batches of image data for training and validation.

Data augmentation techniques such as rescaling, rotation, width/height shift, shear, zoom, and horizontal flip are applied to augment the training images. This helps in increasing the diversity of the training dataset and improving the model's generalization. Two generators are created: 'train_generator' for training data and 'val_generator' for validation data. These generators yield batches of images along with their corresponding labels. For visualization, we used plots() function to visualize training images with their corresponding labels.

A Sequential model is built using Keras, which is a linear stack of layers where you can add layers sequentially. The Sequential model allows us to easily add layers one by one, making it convenient for building simple architectures like CNNs.

Convolutional layers are fundamental building blocks of CNNs. These layers apply convolution operations to input images using learnable filters. Each filter slides over the input image and performs element-wise multiplication followed by summation, producing a feature map that highlights different patterns in the input.

Pooling layers are used to downsample the feature maps generated by convolutional layers. MaxPooling is a type of pooling operation where the maximum value within each window of the feature map is retained, effectively reducing the spatial dimensions of the feature map while retaining the most important information. Downsampling helps in reducing the computational complexity of the model and controlling overfitting by extracting the most relevant features.

Dropout is a regularization technique used to prevent overfitting in neural networks. During training, the Dropout layer randomly deactivates a fraction of neurons in the previous layer, forcing the model to learn redundant representations and improving its generalization ability. We chose a dropout rate of 0.5, meaning that 50% of the neurons are randomly dropped during training.

The Flatten layer is used to convert the output of the last convolutional layer (which is typically a 3D tensor) into a 1D tensor. This flattening step is necessary before passing the feature map to the fully connected layers for classification.

Dense layers are fully connected layers where each neuron is connected to every neuron in the previous and next layers. These layers perform classification by computing weighted sums of the input features and applying activation functions. We have used a Dense layer with softmax activation is used as the output layer for multi-class classification. Softmax activation ensures that the output values represent probabilities of each class, and the class with the highest probability is predicted.

The model is compiled using the compile() method. Categorical cross-entropy is chosen as the loss function, which is suitable for multi-class classification problems. We used Adam optimizer for optimization, which adapts the learning rate during training and accuracy is chosen as the metric to monitor during training.

The model architecture is visualized using the model_to_dot() function, which generates a graphical representation of the model's layers and connections. The model is then trained using the fit() method. Training data (train_generator) and validation data (val_generator) are passed to the model for training. The number of epochs is set to 30, indicating the

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13374

number of times the entire training dataset is passed through the network. During training, the model's performance metrics such as loss and accuracy are monitored on both training and validation data for each epoch.

VI. RESULT AND DISCUSSION

The model was trained using the above specified methodology and we achieved an accuracy of 80.56% using the CNN model.

```
# Fitting the data
history = model.fit(train_generator,
    epochs=30,
    validation_data = val_generator,
)
Epoch 2/30
49/49 [==============================] - 145s 3s/step - loss: 1.1652 - accuracy: 0.5064 - val_loss: 1.2534 - val_accuracy: 0.4463
Epoch 3/30
49/49 [==============================] - 141s 3s/step - loss: 1.0962 - accuracy: 0.5435 - val_loss: 1.0307 - val_accuracy: 0.5207
Epoch 4/30
49/49 [==============================] - 132s 3s/step - loss: 1.0490 - accuracy: 0.5972 - val_loss: 1.9376 - val_accuracy: 0.4876
Epoch 5/30
49/49 [==============================] - 152s 3s/step - loss: 1.0871 - accuracy: 0.5806 - val_loss: 1.3554 - val_accuracy: 0.5207
Epoch 6/30
49/49 [==============================] - 132s 3s/step - loss: 0.9598 - accuracy: 0.6266 - val_loss: 0.9766 - val_accuracy: 0.5785
Epoch 7/30
49/49 [==============================] - 142s 3s/step - loss: 0.9873 - accuracy: 0.6100 - val_loss: 1.3391 - val_accuracy: 0.5372
Epoch 8/30
49/49 [==============================] - 146s 3s/step - loss: 0.8959 - accuracy: 0.6701 - val_loss: 1.1753 - val_accuracy: 0.5620
Epoch 9/30
49/49 [==============================] - 143s 3s/step - loss: 0.8154 - accuracy: 0.6675 - val_loss: 1.4652 - val_accuracy: 0.5537
Epoch 10/30
49/49 [==============================] - 141s 3s/step - loss: 0.8759 - accuracy: 0.6675 - val_loss: 0.9513 - val_accuracy: 0.6364
Epoch 11/30
49/49 [==============================] - 141s 3s/step - loss: 0.7428 - accuracy: 0.7097 - val_loss: 1.0951 - val_accuracy: 0.5868
Epoch 12/30
49/49 [==============================] - 132s 3s/step - loss: 0.7620 - accuracy: 0.7161 - val_loss: 1.1598 - val_accuracy: 0.6281
Epoch 13/30
49/49 [==============================] - 137s 3s/step - loss: 0.6718 - accuracy: 0.7442 - val_loss: 0.9773 - val_accuracy: 0.6529
Epoch 14/30
49/49 [==============================] - 140s 3s/step - loss: 0.7209 - accuracy: 0.7251 - val_loss: 0.9981 - val_accuracy: 0.6612
Epoch 15/30
49/49 [==============================] - 140s 3s/step - loss: 0.6689 - accuracy: 0.7430 - val_loss: 1.2607 - val_accuracy: 0.5950
Epoch 16/30
49/49 [==============================] - 140s 3s/step - loss: 0.7059 - accuracy: 0.7289 - val_loss: 0.9647 - val_accuracy: 0.6777
Epoch 17/30
49/49 [==============================] - 141s 3s/step - loss: 0.6905 - accuracy: 0.7442 - val_loss: 0.9008 - val_accuracy: 0.6694
Epoch 18/30
49/49 [==============================] - 132s 3s/step - loss: 0.6808 - accuracy: 0.7251 - val_loss: 0.9402 - val_accuracy: 0.6777
Epoch 19/30
49/49 [==============================] - 140s 3s/step - loss: 0.6860 - accuracy: 0.7417 - val_loss: 1.1076 - val_accuracy: 0.6777
Epoch 20/30
49/49 [==============================] - 140s 3s/step - loss: 0.6269 - accuracy: 0.7558 - val_loss: 0.9708 - val_accuracy: 0.6777
Epoch 21/30
49/49 [==============================] - 130s 3s/step - loss: 0.5864 - accuracy: 0.7788 - val_loss: 0.8949 - val_accuracy: 0.7025
Epoch 22/30
49/49 [==============================] - 146s 3s/step - loss: 0.5852 - accuracy: 0.7864 - val_loss: 1.0493 - val_accuracy: 0.7107
Epoch 23/30
49/49 [==============================] - 140s 3s/step - loss: 0.5861 - accuracy: 0.7916 - val_loss: 1.2857 - val_accuracy: 0.6529
Epoch 24/30
49/49 [==============================] - 139s 3s/step - loss: 0.5894 - accuracy: 0.7826 - val_loss: 1.1683 - val_accuracy: 0.6529
Epoch 25/30
49/49 [==============================] - 140s 3s/step - loss: 0.6473 - accuracy: 0.7519 - val_loss: 0.9931 - val_accuracy: 0.6942
Epoch 26/30
49/49 [==============================] - 146s 3s/step - loss: 0.5855 - accuracy: 0.7826 - val_loss: 1.1938 - val_accuracy: 0.6860
Epoch 27/30
49/49 [==============================] - 131s 3s/step - loss: 0.5471 - accuracy: 0.7980 - val_loss: 0.9013 - val_accuracy: 0.7107
Epoch 28/30
49/49 [==============================] - 131s 3s/step - loss: 0.5382 - accuracy: 0.8018 - val_loss: 1.0832 - val_accuracy: 0.6777
Epoch 29/30
49/49 [==============================] - 131s 3s/step - loss: 0.5396 - accuracy: 0.8031 - val_loss: 0.9532 - val_accuracy: 0.7603
Epoch 30/30
49/49 [==============================] - 148s 3s/step - loss: 0.4826 - accuracy: 0.8056 - val_loss: 0.9949 - val_accuracy: 0.6529
```

Fig. 2 Accuracy of the model

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13374

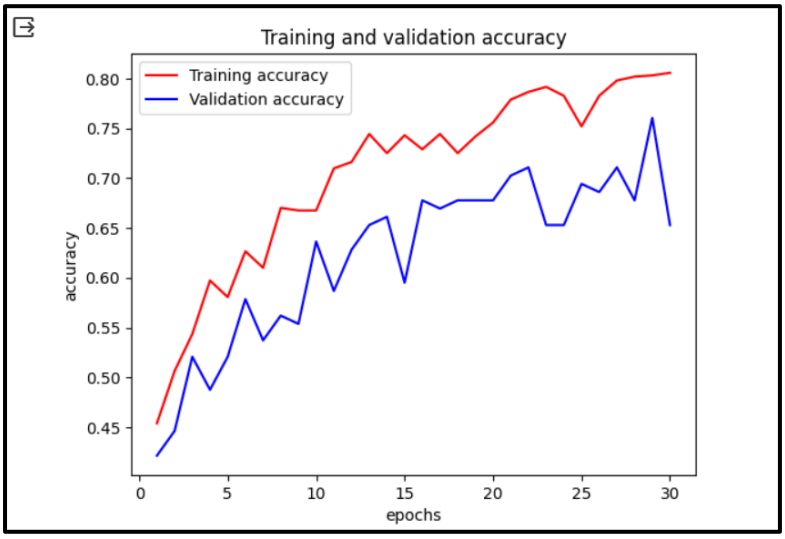

Fig. 3 Graph of Model Accuracy

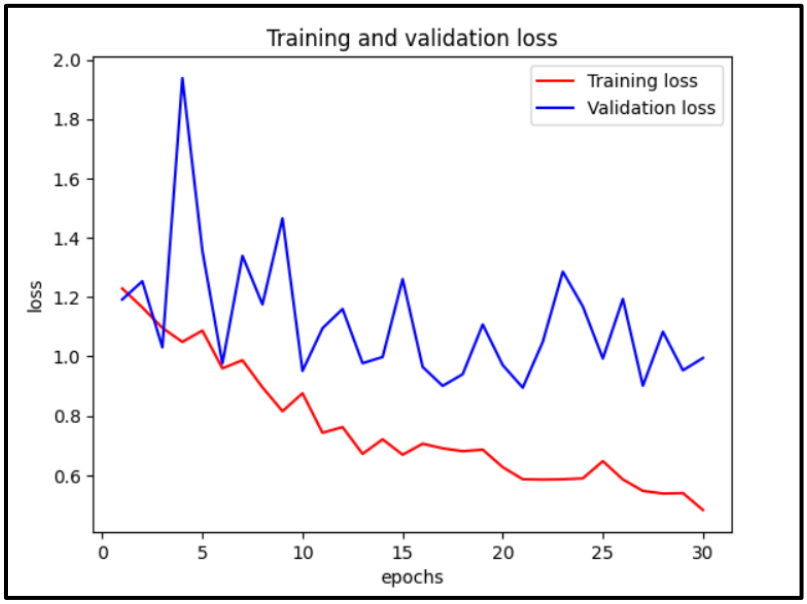

Fig. 4 Graph of Model Loss

VII. CONCLUSION

In conclusion, the implementation of the paddy leaf disease detection system using machine learning techniques represents a significant advancement in precision agriculture. By leveraging convolutional neural networks (CNNs) and deep learning, the system demonstrates remarkable accuracy in identifying various types of paddy leaf diseases. This technology holds immense potential for revolutionizing crop health management practices, enabling farmers to detect diseases early, implement timely interventions, and optimize resource utilization.

Moreover, the integration of this system into agricultural workflows can lead to substantial improvements in crop yields, economic outcomes for farmers, and global food security. By providing a reliable and efficient tool for disease detection, the system empowers stakeholders in the agricultural sector to make informed decisions and mitigate the adverse effects of paddy leaf diseases on crop production.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13374

REFERENCES

[1] Swathika R, N Radha, K Sowmya, "Disease Identification in paddy leaves using CNN based Deep Learning", Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV 2021) IEEE, 04 February 2021, https://doi.org/10.1109/icicv50876.2021.9388557

[2] S. Ramesh *, D. Vydeki Electronics and Communication Engineering Department, VIT University, Chennai 600127, "Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm", Information Processing in Agriculture volume 7, Issue 2, June 2020, https://doi.org/10.1016/j.inpa.2019.09.002

[3] Radhika Deshmukh, Manjusha Deshmukh, "Detection of paddy leaf diseases", International Journal of Computer Applications 975, 8887, 2015

[4] S Pavithra, A Priyadharshini, V Praveena, T Monika, "Paddy leaf disease detection using SVM classifier", International Journal of communication and computer Technologies, 01 January 2019, https://doi.org/10.31838/ijccts/03.01.04 .

[5] M. Ramkumar Raja a,*, Jayaraj V b, Francis H Shajin c, E.M. Roopa Devi d ,"Radial basis function Neural Network optimized with Salp Swarm algorithm espoused paddy leaf disease classification", Biomedical Signal Processing and Control, September 2023, https://doi.org/10.1016/j.bspc.2023.105038

[6] Basavaraj S. Anami, Naveen N. Malvade, Surendra Palaiah, "Deep Learning Approach for recognition and classification of yield affecting paddy crop stresses using field images", Artificial Intelligence in Agriculture, Volume 4, 2020, https://doi.org/10.1016/j.aiia.2020.03.001