



Deep Learning Based Rice Plant Disease Detection

Gaurav Shrivastava^{1*}  and Sachin Patel² 

^{1,2} Sage University, Indore MP 452020, INDIA

*gauravshrivastava31@gmail.com

Abstract. In this paper, rice plant leaf image classification has been discussed to develop a decision support model for Indian farmers. In this context, first an architecture of the proposed decision support model has been explained. Next the required components of the proposed model are discussed. Additionally, different Machine Learning (ML) and Deep Learning (DL) techniques have been implemented and experiments have been carried out. The experiments on the Mendeley dataset have been done. Furthermore, a comparative performance study was performed to select an accurate and efficient data model. Finally, the results have been discussed and a conclusion is provided.

Keywords: Deep Learning, Image Processing, Leaf Image Classification, Machine Learning, Rice Plant Disease Detection.

1 Introduction

India is the second largest country in the world in terms of population. Additionally, up to 65% of the population is dependent on agriculture. But, due to complexities involved in agriculture, the farmers are not making much profit [1]. The diseases in the plants are one of the biggest hurdles. Due to the diseases, the crop production quality and quantity is impacted (He et al, 2021). Plant diseases contribute 10–16% losses in the global harvest of crops each year (Golhani et al, 2018). In this paper, the rice plant disease detection system using Machine Learning (ML) is proposed. In traditional approaches of disease identification the human expertise is required, which may be incorrect sometimes. Therefore, some alternate method is required that can accurately perform the rice plant disease detection (Sladojevic et al, 2016).

The proposed work is classifying the plant leaf images into healthy and diseased plants using ML techniques. Therefore, a function (x) is required to derive from image processing as shown in equation 1, which can extract the features from input image x :

$$(x) = \{f_1, f_2, \dots, f_n\}$$

Or

$$(x) = F \dots \quad (1)$$

These features are accepted by a ML or DL classifier function to classify the features as shown in equation 2.

$$(F) = \{healthy, diseased\} \quad (2)$$

The key issue is to identify the parameters and derive functions according to the definitions of function (F) and (x).

2 Proposed System

The proposed model is an initiative to design a decision support system to help rice farmers, because rice is one of the main food grains in India. A large number of farmers are growing rice crops and completely depend on the production of rice. Additionally, a significant number of farmers suffer from low production of rice because of the different diseases in the rice plant. In order to develop a support model a basic scenario has been given in Fig. 1. According to the figure, a farmer will first click an image directly from the field and will upload it to the server. The server will receive the farmer's image, and then will apply the above discussed components to the image. The server will recognize whether the image is infected with some disease or it is healthy. When the model concludes the results, then it will send the message to the farmer. This message is providing the status of the plant infection.

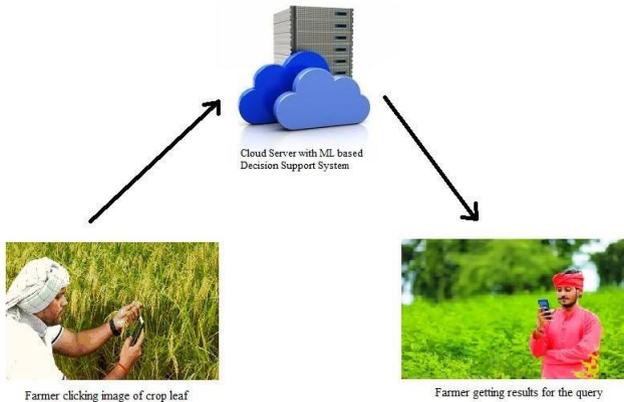


Fig. 1. Proposed Decision Support System (Shrivastava and Patidar, 2022)

2.1 Server Components

The server components of the proposed model are described using Fig. 2. In this figure the first component is the rice plant dataset. The dataset is pre-determined or well-defined historical records, which are used as an example for the machine learning and deep learning algorithms. In this paper, for simulating the decision support system the

The Mendeley dataset is used [6]. The Mendeley dataset contains roughly captured leaf images of four different rice plant diseases namely Tungro, Brownspot, Blast, and Bacterial Blight.

There are different numbers of images in each disease class but not affected by major class imbalance problems. A total of 5932 images are given in the dataset. These images are natural images and in the real world such images are easily available. Thus, it is a practical dataset to use with the application. In addition, the images sent by a farmer can also be similar to the dataset images. These dataset images are used with the machine learning process i.e. data preprocessing, feature selection and algorithm training.

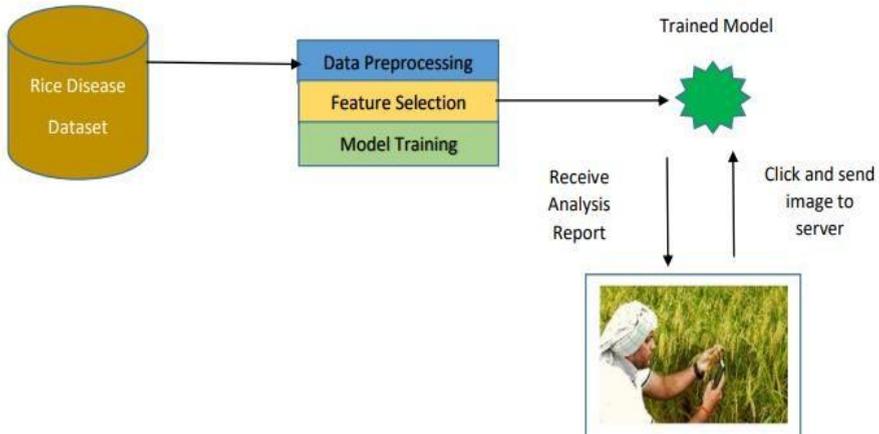


Fig. 2. Server Components of the Proposed Model

1. **Data Preprocessing:** In machine learning, the aim of data preprocessing is to minimize the noisy content of data and maximize the actual or informative data to the dataset for improving the learning process of algorithms. In this case, the data is image data, thus the pixel normalization is used as the preprocessing technique. This process scales the image pixel values between 0 and 1, which minimizes the computational resource need of the server.
2. **Feature Selection:** The feature selection process is used to recover the meaningful insights from the data which can be used by the ML or DL algorithms for training. In literature a number of image feature selection techniques are available and among them, Canny Edge Detection technique [8], Grid Color Moment, the Local Binary Pattern (LBP) (Guo, et al 2010), Sobel (Tian et al, 2021) and K-Means (KM) segmentation are used along with their combinations i.e. “Sobel and LBP”, “Sobel, LBP and color”, and “color and Sobel”. In addition, a deep learning model is also tested for image feature extraction. This model is known as Visual Geometry Group (VGG16) and a deep pre-trained Convolutional Neural Network (CNN) architecture is frequently used in image classification problems.
3. **Model Learning:** The aim of the learning or training is to use the features of input data and prepare a logical model to use and identify the similar objects on which the ML or DL algorithm get training. In literature a number of ML

and DL models are available among them in this experiment, Support Vector Machine (SVM) (Rtaylia and Enneya, 2020), Sequential Convolutional Neural Network (S-CNN), 2-Dimensional Convolutional Neural Network (2D-CNN) [12] are used and their variants prepared by variations of the different hyper parameters.

After training the model is called a trained model. This model can accept the new and unknown input images to perform analysis and return the name of disease. In this context, a series of experiments has been carried out to compare the combinations of feature selection techniques and learning algorithms. Additionally, based on the optimal results for selecting the features and performing classification a deep learning model has been proposed. The proposed deep learning model is demonstrated in Fig. 3.

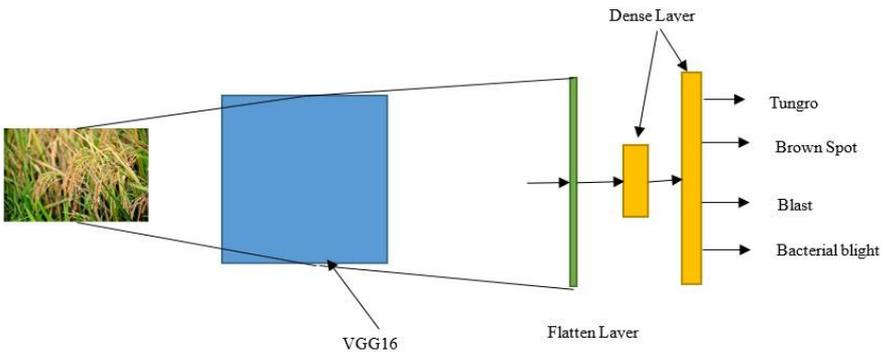


Fig. 3 Implemented Deep Learning Architecture

The implemented deep learning model accepts the input image using an input layer. Next, the input layer passes to the VGG-16 layer (Tammina, 2019). This layer processes the image and returns the features to flatten the layer. The output of VGG16 is converted to a linear vector using the flatten layer. Further, two dense layers are applied to work as fully connected layers. This layer is learning the extracted features of VGG16. First dense layer is configured with the ‘ReLu’ activation function and second dense layer is utilizing ‘Soft-Max’ function. The last layer is a four-neuron dense layer and used to produce the classification results

3 Results and Discussion

The proposed model is the conclusion of the different investigations and experiments involved in background. Therefore, the different feature selection techniques and ML algorithms are implemented and experiments performed. The following set of experiments has been conducted for selecting the optimal methods.

3.1 Plant Leaf Classification

The plant leaf classification model uses SVM and ANN classifiers. The performances in terms of accuracy, f-score and training time are measured. First, the accuracy is measured. The accuracy demonstrates the correctness in classification. The target is to recognize the plant leaf and leaf is a rice plant leaf or not. The accuracy with a different number of plant crops is given in table 1. According to the results, both the algorithms offer similar accuracy, but when the number of species are increased, then, the accuracy of both the models are slightly dropped. But the ANN is providing stable and robust performance as compared to SVM. In addition, the f-score has also been given in the table. The measured f-score is given with an increasing number of plant species. According to the results, the ANN is accurate as compared to SVM. Thus, the number of classes to learn can impact on performance.

Table 1. Accuracy, F-score and Training Time of Plant Species Identification

| Classes (Plant Species) | Accuracy | | F-Score | | Training Time | |
|-------------------------|----------|------|---------|------|---------------|-----|
| | SVM | ANN | SVM | ANN | SVM | ANN |
| 2 | 98.3 | 100 | 95.8 | 100 | 10 | 37 |
| 3 | 96.8 | 99.3 | 93.5 | 98.4 | 34 | 43 |
| 4 | 94.5 | 97.7 | 93.9 | 94.6 | 86 | 59 |
| 5 | 90.6 | 94.2 | 89.4 | 90.5 | 112 | 71 |
| 6 | 87.3 | 92.8 | 84.2 | 89.1 | 178 | 95 |
| 7 | 84.8 | 90.5 | 82.7 | 87.3 | 220 | 108 |
| 8 | 80.4 | 91.7 | 80.2 | 85.9 | 289 | 132 |

Next, the performance of the algorithms for training time requirement was measured. The training time is the amount of time required to perform training of the algorithm. It is measured in terms of seconds (Sec). The training time is increasing with the increasing number of plant species. According to the results, when the amount of learning data is limited, then the SVM algorithm provides better efficiency, but as the amount of learning data increases, it requires more learning time. Thus, the ANN not only provides the higher accuracy but also provides the robustness with the large training samples.

3.2 Rice Plant Disease Detection

In this section, the performance of the model is measured for the rice plant disease detection into four types namely Bacterial blight, Blast, Brown Spot and Tungro. The performance of the model is given in Table 2 and Fig. 4.

Table 2. Performance of Diseased Rice Plant Detection

| Dataset | Accuracy | | F-score | | Training Time | |
|----------|----------|------|---------|------|---------------|-----|
| | SVM | ANN | SVM | ANN | SVM | ANN |
| Kaggle | 92.4 | 99.5 | 90.2 | 99.1 | 98 | 74 |
| Mendeley | 83.6 | 92.7 | 84.8 | 89.7 | 370 | 183 |

According to the results, both the algorithms are providing acceptable accuracy but the ANN model is more accurate and efficient. Therefore, based on the efficiency and accuracy, the ANN is more suitable as compared to SVM.

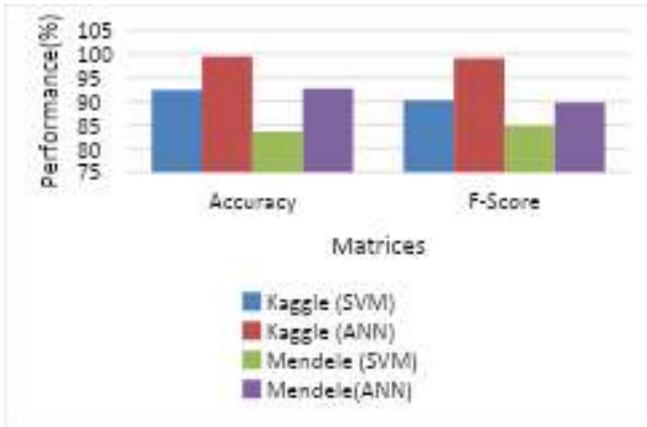


Fig.4. Mean Accuracy and F-Score

3.3 Comparing Deep Learning Models

In this section, the two architectures specifically S-CNN and 2D CNN have been considered. The aim is to compare both the models for accuracy to classify the rice leaf images. In this experiment, an original dataset without any preprocessing has been used. The comparative precision, recall and f1-score are measured class-wise and reported in Table 3. In addition, the accuracy of the models is also given

Table 3. Performance Comparison of S-CNN and 2D CNN Models

| Classes | S-CNN | | | 2D-CNN | | |
|----------|-----------|--------|---------|-----------|--------|---------|
| | Precision | Recall | F-score | Precision | Recall | F-score |
| 0 | 0.5 | 0.75 | 0.60 | 0.73 | 1 | 0.84 |
| 1 | 0.5 | 0.57 | 0.53 | 0.71 | 0.71 | 0.71 |
| 2 | 1.0 | 0.44 | 0.62 | 1.0 | 0.67 | 0.80 |
| Accuracy | 0.58 | | | 0.79 | | |

According to the performance, the 2D-CNN is more accurate than the S-CNN. Additionally, it provides higher precision, recall and F1-score. Thus, the 2D-CNN is more appropriate than the S-CNN.

3.4 Performance for Feature Extraction Techniques

To enhance the classification performance, different combinations of the features are prepared and used with the 2D-CNN. The following feature combinations has prepared:

- 1 **Sobel operator:** The data is processed using Sobel operator. The 2D CNN model is trained with the different epochs i.e. 200, 500 and 1000.
- 2 **LBP:** The dataset is processed using the LBP and the performance of 2D CNN model has been measured.
- 3 **Sobel and LBP features:** A combined image is prepared by using LBP and Sobel operators. This feature has the 3D vector with the size of $90 * 90 * 2$. Where the $90 * 90$ is the image size and 2 vectors combined.
- 4 **Sobel, LBP and color channels:** A combined image is prepared with the help of LBP, Sobel and 3 color channels. That consists of the size $90 * 90 * 5$.
- 5 **Sobel and color channels:** In these samples, each color channel is extracted and then channels are processed using the Sobel operator. Finally, a combined data is prepared.

The 2D-CNN and the combinations of the image features are used for training and validation. The increasing number of epochs is used in experiments. Table 4 demonstrates the class-based precision and the recall is given in Table 5. According to the results, the combination of 'Sobel, LBP and color feature is providing higher accuracy. Next, the F1-score is also measured and given in Table 6. The f-score is used to decide which algorithm is working better. It is the mean of precision and recall. The combination of color, LBP and Sobel is much better. The mean accuracy is also given in table 7 for 200, 500 and 1000 epochs. According to the results, the combination of features 'color, Sobel and LBP' is providing high accurate result a total of 83%. The data only used with 2DCNN and LBP is providing very poor accuracy. But, the combinations of features are providing higher accuracy as compared to individual features. The original images with the 2D-CNN model

provides higher accurate validation results.

Table 4. Class Wise Precision of 2D-CNN Model with Image Features

| Classes | Sobel | LBP | LBP + Sobel | Color + LBP + Sobel | Sobel + Color |
|-------------------|-------|------|-------------|---------------------|---------------|
| 200 Epoch | | | | | |
| 0 | 0.78 | 0.47 | 0.80 | 0.73 | 1.00 |
| 1 | 0.62 | 0.00 | 0.50 | 0.83 | 0.54 |
| 2 | 0.71 | 0.57 | 0.67 | 1.00 | 1.00 |
| 500 Epoch | | | | | |
| 0 | 0.88 | 0.47 | 0.83 | 0.75 | 1.00 |
| 1 | 0.78 | 0.50 | 0.75 | 0.75 | 0.70 |
| 2 | 0.82 | 0.60 | 0.60 | 1.00 | 0.67 |
| 1000 Epoch | | | | | |
| 0 | 0.78 | 0.62 | 0.86 | 0.89 | 1.00 |
| 1 | 0.78 | 0.43 | 0.64 | 0.78 | 0.73 |
| 2 | 0.83 | 0.75 | 0.83 | 1.00 | 1.00 |

However, the accuracies of some algorithms are improving with the epoch and for some are degrading accuracies. According to the validation accuracy, the 2D-CNN and combination of feature selection techniques ‘color and Sobel’ is providing higher accuracy.

Table 5. Recall for 200 Epochs

| Classes | Sobel | LBP | LBP + Sobel | Color + LBP + Sobel | Sobel + Color |
|-------------------|-------|------|-------------|---------------------|---------------|
| 200 Epoch | | | | | |
| 0 | 0.88 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1 | 0.71 | 0.00 | 0.57 | 0.71 | 1.00 |
| 2 | 0.56 | 0.44 | 0.44 | 0.78 | 0.33 |
| 500 Epoch | | | | | |
| 0 | 0.67 | 0.89 | 0.56 | 1.00 | 0.89 |
| 1 | 0.75 | 0.12 | 0.75 | 0.75 | 0.88 |
| 2 | 0.71 | 0.43 | 0.86 | 0.57 | 0.57 |
| 1000 Epoch | | | | | |
| 0 | 0.78 | 0.89 | 0.67 | 0.89 | 1.00 |
| 1 | 0.88 | 0.38 | 0.88 | 0.88 | 1.00 |
| 2 | 0.71 | 0.43 | 0.71 | 0.86 | 0.57 |

Additionally, with the increasing epoch, it is enhancing. The combined features ‘Sobel and Color’ and ‘Color, LBP and Sobel’ is providing higher correctness. The

combinations of the features are providing accuracy up to 88%.

Table 6. F-Score for all the Algorithms

| Classes | Sobel | LBP | LBP + Sobel | Color + LBP + Sobel | Sobel + Color |
|-------------------|-------|------|-------------|---------------------|---------------|
| 200 Epoch | | | | | |
| 0 | 0.82 | 0.64 | 0.89 | 0.84 | 1.00 |
| 1 | 0.67 | 0.00 | 0.53 | 0.77 | 0.70 |
| 2 | 0.63 | 0.50 | 0.53 | 0.88 | 0.50 |
| 500 Epoch | | | | | |
| 0 | 0.71 | 0.62 | 0.67 | 0.86 | 0.94 |
| 1 | 0.71 | 0.20 | 0.75 | 0.75 | 0.78 |
| 2 | 0.71 | 0.50 | 0.71 | 0.73 | 0.62 |
| 1000 Epoch | | | | | |
| 0 | 0.78 | 0.73 | 0.75 | 0.89 | 1.00 |
| 1 | 0.82 | 0.40 | 0.74 | 0.82 | 0.84 |
| 2 | 0.77 | 0.55 | 0.77 | 0.92 | 0.73 |

Based on the overall results, the increasing epochs can increase the accuracy. But this fact is not always true. Finally, the feature selection techniques and their combinations can influence the recognition performance. But, the 2D-CNN self-extracts the features and learns with it, so it can perform better. It can provide up to 90% accurate results.

Table 7. Accuracy of the Algorithms for Increasing Number of Epochs

| Epochs | Sobel | LBP | LBP + Sobel | Color + LBP + Sobel | Sobel + Color |
|--------|-------|------|-------------|---------------------|---------------|
| 200 | 0.71 | 0.50 | 0.67 | 0.83 | 0.75 |
| 500 | 0.75 | 0.50 | 0.71 | 0.79 | 0.79 |
| 1000 | 0.79 | 0.58 | 0.75 | 0.88 | 0.88 |

3.5 Deep Feature Selection

The aim is to conclude which type of dataset is more appropriate for learning with the deep models. The combination of appropriate data and deep learning can improve the quality of service of the decision support model. Two different datasets and a deep learning architecture has been considered. First dataset is based on pretreated images and the second dataset contains the raw images. The experiment has been carried out and the performances are revealing which dataset is appropriate. For measuring the correctness precision, recall, f1-score and accuracy are used. Moreover, for efficiency the training time is measured.

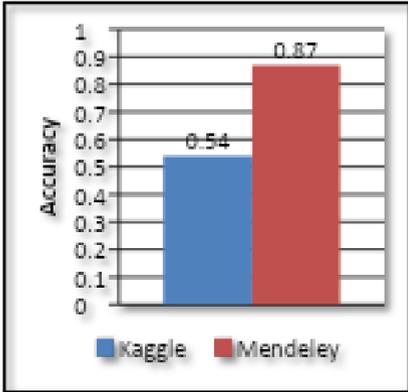


Fig. 5. Mean Accuracy of the Model

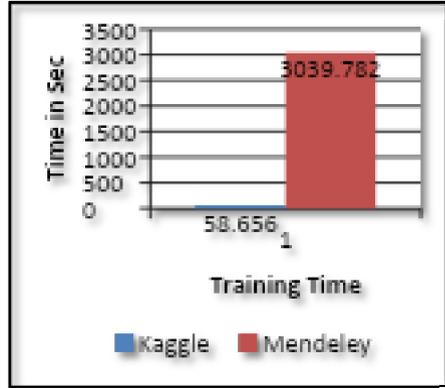


Fig. 6. Training Time for the Model

The accuracy with both the datasets are given in Fig. 5. According to the accuracy, the model is performing less accurately with the Kaggle dataset, but it works better with Mendeley dataset. The deep learning models can do better with the large data. Therefore, the accuracy of the Mendeley dataset is high. By using the performance, the Mendeley-based dataset provides better yield as compared to Kaggle dataset. The training accuracy for Mendeley is near about 90% and for validation it is approximately 88%. Similarly, for the Kaggle dataset the training accuracy is found near about 70% and the validation accuracy reaches near about 60%. The performance for Kaggle and Mendeley dataset in precision, recall, and f1-score is given in Table 8.

Table 8. Performance of the Model for Datasets

| Classes | Precision | Recall | F1-score |
|-------------------------|-----------|--------|----------|
| Kaggle Dataset | | | |
| 0 | 0.78 | 0.88 | 0.82 |
| 1 | 0.29 | 0.29 | 0.29 |
| 2 | 0.50 | 0.44 | 0.47 |
| Mendeley Dataset | | | |
| 0 | 0.78 | 0.89 | 0.83 |
| 1 | 0.81 | 0.87 | 0.84 |
| 2 | 0.95 | 0.88 | 0.91 |
| 3 | 0.98 | 0.82 | 0.89 |

According to the results, the model is providing better results with the Mendeley dataset in terms of all the considered parameters. The Mendeley dataset provides better classification as compared to the Kaggle. In addition, the training time of the model using both datasets are also measured. Fig. 6 demonstrates the training time of

the models. In this diagram, the datasets are given on the X-axis and the Y-axis shows the training time. According to the results, the Kaggle dataset utilizes a fewer time whereas the Mendeley dataset requires a higher amount of training time. The time is measured in terms of seconds. The training time of the Mendeley dataset is approximately 60 times higher than the Kaggle, additionally, the dataset size is 35 times higher. Therefore, the roughly taken images provide a better yield as compared to the specifically designed datasets, and the deep learning model can learn better with them.

3.6 Performance of the Proposed Model

In this section, the performance of the proposed model is evaluated and compared in terms of recall, f-score, and precision, which are accounted in Table 9.

Table 9. Performance Analysis

| Precision | | | | | |
|-----------|----------------|--------|----------------|---------|--------|
| Classes | Sequential CNN | 2D-CNN | Sobel Operator | K-Means | VGG-16 |
| 0 | 0.72 | 0.85 | 0.78 | 0.75 | 0.96 |
| 1 | 0.74 | 1.00 | 0.74 | 0.71 | 0.99 |
| 2 | 0.87 | 1.00 | 0.86 | 0.94 | 0.98 |
| 3 | 0.73 | 1.00 | 0.85 | 0.80 | 0.98 |
| Recall | | | | | |
| 0 | 0.79 | 1.00 | 0.69 | 0.74 | 0.99 |
| 1 | 0.59 | 0.93 | 0.79 | 0.80 | 0.95 |
| 2 | 0.85 | 0.96 | 0.79 | 0.69 | 0.98 |
| 3 | 0.83 | 0.93 | 0.99 | 0.96 | 0.98 |
| F1-Score | | | | | |
| 0 | 0.75 | 0.92 | 0.73 | 0.74 | 0.97 |
| 1 | 0.66 | 0.96 | 0.76 | 0.75 | 0.97 |
| 2 | 0.86 | 0.98 | 0.82 | 0.79 | 0.98 |
| 3 | 0.78 | 0.98 | 0.92 | 0.87 | 0.99 |

In addition, the validation accuracy and Training time of the model has also been evaluated, which are reported in Fig. 7 and Fig. 8 respectively. In both the diagrams, X-axis contains the used ML models and Y-axis shows the Accuracy and Training time. According to the results the proposed model is providing highest accuracy up to 99% with the minimum training time of 2661.31 seconds. Thus, the proposed method is acceptable for offering services to the real-world applications.

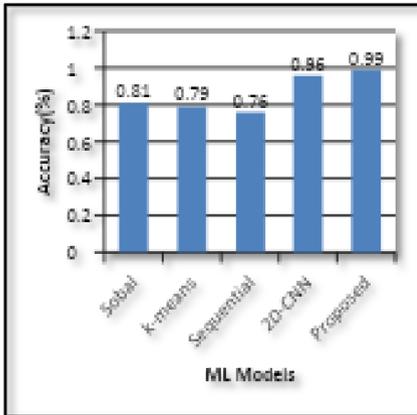


Fig. 7. Accuracy of ML Based Rice Plant Disease Detection System

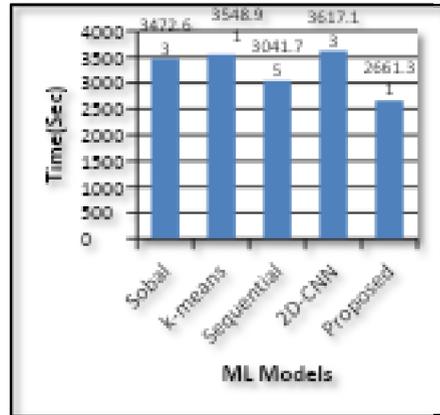


Fig. 8. Training Time of the Rice Plant Disease Detection Systems

Conclusion and Future Work

The proposed work is aimed to design an efficient and accurate machine learning model for rice plant disease detection. In this context, architecture to support the proposed concept has been discussed and a deep learning model has been implemented for accurate disease detection. The proposed model provides the high detection rate up to 90%.

References

1. "India at a Glance", Food and Agriculture organization of United Nations, <https://www.fao.org/india/fao-in-india/india-at-a-glance/en/>, last accessed 2024/12/06
2. He, D. C., He, M. H., Amalin, D. M., Liu, W., Alwindia, D. G., Zhan, J. **Biological Control of Plant Diseases: An Evolutionary and Eco-Economic Consideration** Pathogens 2021, 10, 1311,(2021).
3. Golhani, K., Balasundram, S. K., Vadamalai, G., Pradhan, B. "A Review of Neural Networks in Plant Disease Detection Using Hyperspectral Data", **Information Processing in Agriculture** 5, pp. 354–371,(2018)
4. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D. **Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification**, Hindawi Publishing Corporation **Computational Intelligence and Neuroscience** vol. 2016, Article ID 3289801, 11 pages,(2016)
5. Shrivastava, G., Patidar, H. "Rice Plant Disease Identification Decision Support Model using Machine Learning", **ICTACT Journal on Soft Computing**, April 2022, Volume: 12, Issue: 03,(2022)
6. <https://data.mendeley.com/datasets/fwcj7stb8r/1>, last accessed 2024/10/5
7. Tian, R., Sun, G., Liu, X., Zheng, B. "Sobel Edge Detection Based on Weighted Nuclear Norm Minimization Image Denoising", **Electronics** 2021, 10, 655.(2021)
8. Canny Edge Detection, March 23, 2009
9. Guo, Z., Zhang, L., Zhang, D. "A Completed Modeling of Local Binary Pattern

- Operator for Texture Classification”, IEEE trans. on img proc.(2010)
10. Rtylia, N., Enneya, N. “Selection Features and Support Vector Machine for Credit Card Risk Identification”, Procedia Manufacturing 46, 941–948,(2020)
 11. Tammina, S. “Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images”, International Journal of Scientific and Research Publications, Volume 9, Issue 10, October,(2019)
 12. “Introduction to Convolution Neural Network”, <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>, last accessed 2024/12/06.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

