## A Project Report

**on**

**RICE CROP DISEASE DETECTION USING MACHINE LEARNING**

**submitted in partial fulfillment of the requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**by**

**17WH1A0588 Ms. V. RESHMA REDDY**

**17WH1A05A5 Ms. CH. SRAVANI**

**17WH1A05A8 Ms. A. RAMYA**

**under the esteemed guidance of**

**Dr. N. SREEKANTH**

**Associate Professor**



**Department of Computer Science and Engineering**

BVRIT HYDERABAD

College of Engineering for Women

(NBA Accredited – EEE, ECE, CSE and IT)

**(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)**

**Bachupally, Hyderabad – 500090**

**May, 2021**

**DECLARATION**

We hereby declare that the work presented in this project entitled **“RICE CROP DISEASE DETECTION USING MACHINE LEARNING** ”submitted towards completion of Project Work in IV year of B.Tech., CSE at ‘BVRIT HYDERABAD College of Engineering For Women**’**, Hyderabad is an authentic record of our original work carried out under the guidance of Dr. N. Sreekanth, Associate Professor, Department of CSE.

Sign. with date:

**Ms. V. RESHMA REDDY**

**(17WH1A0588)**

Sign. with date:

**Ms. CH. SRAVANI**

**(17WH1A05A5)**

Sign. with date:

**Ms. A. RAMYA**

**(17WH1A05A8)**

BVRIT HYDERABAD

College of Engineering for Women

(NBA Accredited – EEE, ECE, CSE and IT)

**(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)**

**Bachupally, Hyderabad – 500090**

**Department of Computer Science and Engineering**



###### Certificate

This is to certify that the Project Work report on “ **RICE CROP DISEASE DETECTION USING MACHINE LEARNING**” is a bonafide work carried out by Ms. V. RESHMA REDDY (17WH1A0588) ; Ms. CH. SRAVANI (17WH1A05A5) ; Ms. A. RAMYA (17WH1A05A8) in the partial fulfillment for the award of B.Tech. degree in **Computer Science and Engineering, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under our guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

**Head of the Department Guide**

**Dr. K.Srinivas Reddy Dr. N.Sreekanth**

**Professor and HoD, Associate Professor**

**Department of CSE**

**External Examiner**

**Acknowledgements**

We would like to express our sincere thanks to **Dr. K V N Sunitha, Principal**, **BVRIT HYDERABAD College of Engineering for Women**, for providing the working facilities in the college.

Our sincere thanks and gratitude to our **Dr. K. Srinivas Reddy, Professor**, Department of CSE, **BVRIT HYDERABAD College of Engineering for Women** for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Mr. N. Sreekanth, Associate Professor**, Department of CSE**, BVRIT HYDERABAD College of Engineering for Women** for his constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinator, all the faculty and staff of **CSE** Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

**Ms. V. RESHMA REDDY**

**(17WH1A0588)**

**Ms. CH. SRAVANI**

**(17WH1A05A5)**

**Ms. A. RAMYA**

**(17WH1A05A8)**

## 

## 

## ABSTRACT

Rice leaves are affected by various kinds of diseases. Rice leaf diseases can be detected based on image processing techniques. Diseases are classified in rice plants by analyzing photographs of rice leaves. The method uses image processing algorithms to detect leaves.Attributes are computed based on the dimensions of leaves and lesions, the numbers and shapes of lesions, as well the color characteristics of lesions and intact portions of leaves.Bacterial Leaf Blight, Blast and Brown Spot are a major bacterial and fungal inflammation respectively in rice (Oryza sativa) crops, it causes harvest loss and reduces the grains quality.VGG19 algorithm is used for the classification.This system can classify the percentage of infected leaves by using image data analysis.This system can make our work easy with big agro farm to identify infected plants. This helps the farmer in protecting the crop right from early stages.

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **Page no** |
| 1 | Images of diseased plant leaves | 2 |
| 2 | Methodology at a glance | 7 |
| 3 | CNN VGG 19 Model | 10 |
| 4 | Block Diagram | 11 |
| 5 | Use Case Diagram | 12 |
| 6 | Class Diagram | 13 |
| 7 | Sequence Diagram | 13 |
| 8 | Activity Diagram | 14 |
| 9 | Test Case 1 | 15 |
| 10 | Test Case 2 | 15 |
| 11 | Test Case 3 | 16 |
| 12 | Result Graph | 17 |
| 13 | S-Shaped Curve | 21 |

**Contents**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Topic** | **Page No.** |
|  | Abstract | i |
|  | List of Figures | ii |
| 1. | Introduction | 1 |
|  | 1.1 Objective | 1 |
|  | 1.2 Related Works | 2 |
|  | 1.3 Challenges | 3 |
|  | 1.3.1 Data Collection | 3 |
|  | 1.3.2 Model Selection  1.3.3 Data Labeling | 3  3 |
| 2. | Main Text | 4 |
|  | 2.1 Literature Review | 4 |
|  | 2.2 Tools & Technologies | 5 |
|  | 2.3 Detailed Report  2.3.1 Data Collection Procedure  2.3.1.Data Preprocessing  2.3.2 Data Organising  2.3.3 Labeling Data  2.3.4 Data Storing  2.3.5 Model Summary  2.3.2 Block Diagram  2.3.3 Use Case Diagram  2.3.4 Class Diagram  2.3.5 Sequence Diagram  2.3.6 Activity Diagram | 5  5  6  6  6  6  7  11  12  13  13  14 |
|  | 2.4 Test Cases  2.4.1 Bacterial leaf blight  2.4.2 Leaf blast  2.4.3 Brown spot  2.5 Results & Discussions  2.5.1 Results  2.5.2 Discussion | 15  15  15  16  16  16  17 |
| 3. | Conclusion & Recommendations | 18 |
|  | 3.1 Summary | 18 |
|  | 3.2 Conclusion | 18 |
|  | 3.3 Recommendation | 18 |
|  | 3.4 Future Scope | 19 |
| 4. | Appendices | 20 |
|  | 4.1 Classifiers Used | 20 |
|  | 4.2 K-Nearest Neighbour | 21 |
|  | 4.3 Decision Tree | 22 |
|  | 4.4 Naive Bayes Classifier | 23 |
|  | 4.5 Python Libraries | 23 |
|  | 4.6 Experimental Setup | 25 |
|  | 4.7 Code  4.7.1 Rice Algorithm  4.7.2 Application  4.7.3 Output Check  4.7.4 HTML Pages  4.7.4.1 Bacterial Leaf Blight  4.7.4.2 Blast  4.7.4.3 Brownspot | 26  26  29  31  33  33  35  37 |
| 5. | References | 39 |
| 6. | Glossary | 42 |

**1. INTRODUCTION**

**1.1 OBJECTIVE**

The main focus of this project is making a system that can identify the disease of the rice plant by analyzing the diseased leaf . While providing a vital role in the country’ s economy, rice serves as a staple food for the mass population and provides two-thirds of the per capita daily calorie intake. As per the USDA’ ers report, total rice yielding area and corresponding production are projected to be 11.8 million hectares and 35.3million metric tons respectively for 2019-2020 (May to April). These economic turnouts clearly indicate that proper rice cultivation is a high priority for India. Disease free rice cultivation would play a dominant role in ensuring stable economic growth and maintaining the desired targets. Moreover,to keep pace with the emerging fourthindus-trial revolution, India needs to work for its industrial advancements which will involve smart systems that can take decisions without any human interventions. To that end, we have come up with an automated system using machine learning techniques, a system that will contribute to a country’ s agricultural development by automatically identifying and classifying diseases from the images of rice leaves. Twenty rice diseases were revealed in India from a survey conducted in 1979-1981, among which 13 diseases were detected as the major ones. Rice blast and brown spots were considered as the most prominent diseases then, but now brown spot and bacterial blight are considered as the most prominent and dangerous rice diseases. In this paper, we have focused on the identification of three rice leaf disease detection (bacterial blight, brown spot and leaf blast). The Reason for choosing these three diseases is the prevalence of these diseases in India.

These three different diseases have their characteristic pat-terns and shapes. The features of the diseases is described below and illustrated in Fig.1:

• Leaf blast: small brown linear lesions on leaf blades, leaf tips may turn grey and dry and spindle shape dryness.

• Bacterial blight: elongated lesions near the leaf tips and margins, and turns white to yellow and then grey due to fungal attack.

• Brown Spot:dark brown colored and round to oval shaped lesions on rice leaves.



Fig -1: Images of diseased plant leaves

**1.2 Related Works**

In a research done by Deepika Jaswal, Saoura.V, K.P.Soman in Image Classification Using Convolutional Neural Networks classified different kinds of image, scene and environments. They used CNN layers to make the classifier to do the classification. They proved that CNN can work with both face and non-face classification, building and nature classification, dense and rational classification. They used 250 epochs in their classification.They got about 77% accuracy in forest and agricultural classification and about 91% of accuracy in residential vs agricultural classification. They got about 97% of accuracy by working with green regions.They also proved that CNN gives the best accuracy in the green scene rather than the building scene. So it works not only with material or building images but also with nature and fruits.So from this research we can Ensure that CNN is a great algorithm to work with nature and nature products like rice leaf disease detection.

A research done by Bingquan Hua and Fengling Yin on the topic Research on Novel Image Classification Algorithm based on Multi-Feature Extraction And Modified SVM Classifier. In their research they worked with image object classifiers and SVM,Kernel And Optimization.They used common vector machines (SVM), K-neighbours neural networks and random forest to make the model in order to classify images. They found that SVM has the best classification results and reduced amount of training time etc. They successfully overcome the most common problem in computer vision classification and object detection by their proposed model. But their accuracy and result accuracy was not very high in their proposed model. So I can decide from the research that CNN will work better than SVM in scene and nature image classification.

**1.3 Challenges**

**1.3.1 DataCollection**

Since there isn't any available image data online, data collection was very difficult for this project. But when we start collecting rice leaves image data locally it becomes too difficult because of the COVID 19 pandemic. For more accuracy we need more training data. But we weren't able to collect many pictures and visit many fields. Then we decided to take the data from the local agricultural research station ANGRAU(Acharya N G Ranga Agricultural University).

**1.3.2** **Model selection**

Project success depends on Dataset and Model selection. The right choice will lead you to your goal quickly and the wrong one will ruin it. We tested different types of model with the test data and tried to find the best one..In this project a CNN algorithm VGG 19 because the model is used for training data sets.

**1.3.3** **Data labeling**

Data labeling was the most important part of our work. Because it makes code faster to run and less time consuming for data training. In this project labeling data is done based on image data and folder names.

**2. MAIN TEXT**

**2.1 LITERATURE REVIEW**

Sladojevic and colleagues aimed to detect plant diseases using Deep Learning techniques that will help the farmers to quickly and easily detect diseases which in turn would enable the farmers to take proper steps at an early stage. They used 2589 original images in performing tests and 30880 images for training their model using the Caffe deep learning framework. For achieving a higher accuracy in evaluating a predictive model, the authors used a 10-fold cross validation technique in their dataset.

The accuracy of prediction of this model is 96.77%.Depending on only the extracted percentage of the RGB value of the affected area of rice leaf using image processing, amodel was developed to classify the disease. The RGB percentages were fed into Naive Bayes classifier to finally categorize the diseases into three disease classes: Bacterial leaf blight, Rice blast and Brown spot. The accuracy of the model to classify the diseases is over 89%.A higher accuracy was found in paper where a plant disease detection model was developed using CNN. This model can identify 13 different types of diseases of plants.

The final accuracy achieved from this model is 96.3%.In another study, the affected parts were separated from the rice leaf surface using K-means clustering and the model was then trained with SVM using color, texture and shape as the classifying features.Maniyath et al. used random forest, an ensemble learning method, to classify between healthy and diseased leaf. For Extracting the features of an image, the authors used Histogram Of Oriented Gradient (HOG). Their work has claimed an accuracy 92.33%.Image Processing and machine learning techniques were also used for the detection and classification of rice plant diseases.

Authors of this paper used K-means clustering for the segmentation of the diseased area of the rice leaves and Support Vector Machine (SVM) for classification. They Achieved a final accuracy, 93.33% and 73.33% on training and test dataset respectively. The same dataset was also used in our work but our methodology resulted in a higher accuracy both in training and test dataset.

**2.2TOOLS AND TECHNOLOGIES**

**TOOLS**

* ANACONDA
* FLASK
* JUPYTER NOTEBOOK

**TECHNOLOGIES**

* PYTHON
* HTML
* CSS
* JAVA SCRIPT
* TENSOR FLOW
* KERAS
* NUMPY

**2.3 DETAILED REPORT**

**2.3.1 DataCollectionProcedure**

In Experiment,we used Disease vs. disease image dataset. These data are selected because these are available label data and for our project we need labeled data. We used 2684 images as training data and 671 images as validation data

**2.3.1.1 Data Pre-processing**

Data pre-processing refers to the pre-phase of processing datasets. Generally raw data sets are notable to perform operations and generate expected outcomes. As a result, data preprocessing is required. And it is considered to be one of the most important parts of research. In this phase we collected more than 6000 image data from different sources and tried to find unnecessary or noisy data.

**2.3.1.2 Data Organizing**

In this phase we divided data and stored them in two data folder validation and train, we also used a revalidation folder for check train data validation. Then we divided those validation and training folders data into more folders like Brown spot,Hipsa, leafBlast etc…

**2.3.1.3 Labeling Data**

In this phase we renamed all images as their name and also numbered them sequentially.

**2.3.1.4 Data Storing**

In This part we store All the data in Google drive because it make our mark easier in Google Drive.We can use those online stored data in our project by following some simple step or code which is shown in fig 2

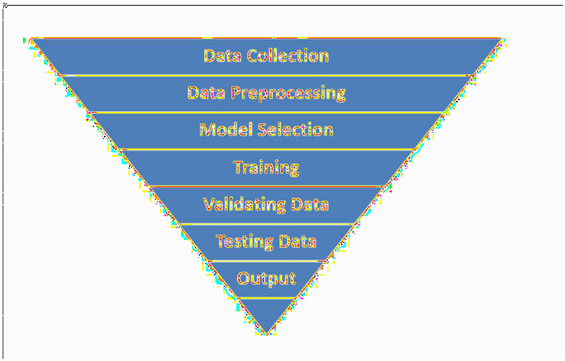


Fig -2 : Methodology At A Glance

## 

**2.3.1.5 Model Summary**

This project uses a deep learning convolutional neural network model-based target detection and positioning method, and proposes an improved VGG model. The trained VGG-19 network is used as a pre-trained model of this model.

By fine-tuning the transfer learning method, the parameters of the pre-trained model optimize the model parameters of the convolution layer and solve the classification problem of rice leaf disease detection. The parameters in VGG-19 are concentrated in 3 FC layers.

Therefore, it is proposed to replace the three fully connected layers of VGG-19 with one Flatten layer and two fully connected layers. Since the convolution layer cannot be directly connected to the Dense fully connected layer, a Flatten layer is added.

The improved model training framework mainly uses the fine-tuning transfer learning to transfer the VGG-19 pretrained model parameters to the convolution layer, pooling layer and fully connected layer of the rice disease detection model, and replaces the original with a 2-label Softmax classifier Softmax classification layer, sparse features through Dropout, Max pooling, and fit a detection model with good accuracy.

The main operation process is as follows

(1) Enter sample pictures of rice diseases. The pictures are extracted from the folder as a training sample set for input.

(2) Pre-processing. In order to improve the training efficiency, the input image is standardized to a resolution of 224 \* 224.

(3) Construct new and improved models. using the VGG-19 Net model, the 3 FC layers are optimized as 1 Flatten layer and 2 FC layers with reduced parameters. Replace the Softmax classification layer of the original model with a 2-label Softmax classifier.

(4) Micro-transfer learning. Using the parameters of the 16 convolutional layers and pooling layers of the VGG pre-trained model, the parameters of the detection model were optimized by transfer learning.

(5) Model training. To train and optimize the parameters of 2 FC layers and Softmax layers, it is necessary to freeze the parameters of 16 convolutional layers and their pooling layers and initialize the model parameters using a random method, set the momentum parameters, the learning rate, and the accuracy standard to iterate.

(6) Model testing. Extract the data from input image pictures from the data set as the test sample set for model testing, and predict the result.

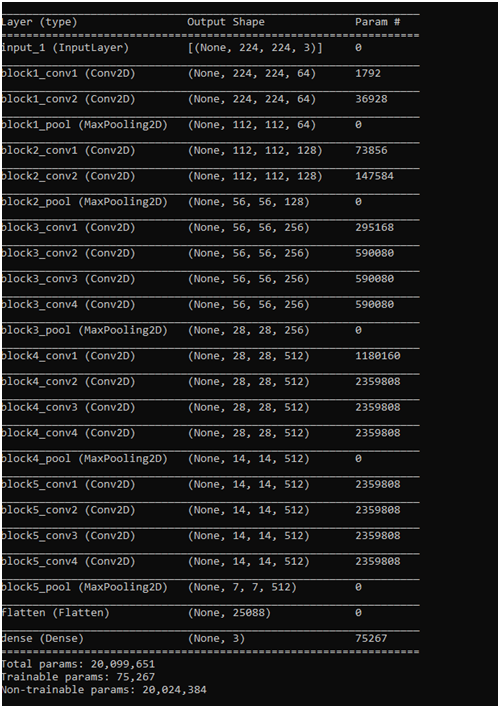


Fig- 3:The CNN VGG 19 model

**2.3.2 BLOCK DIAGRAM**

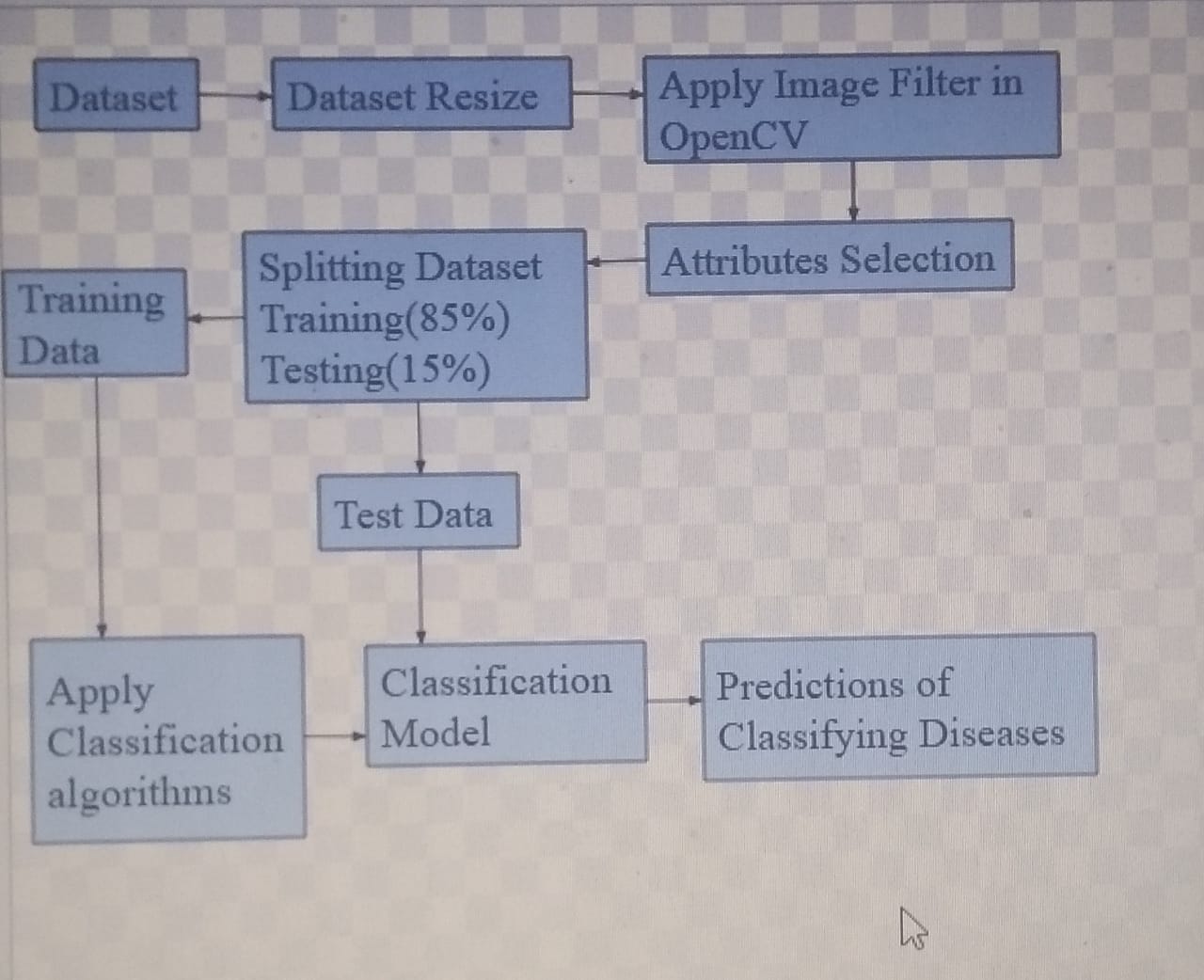
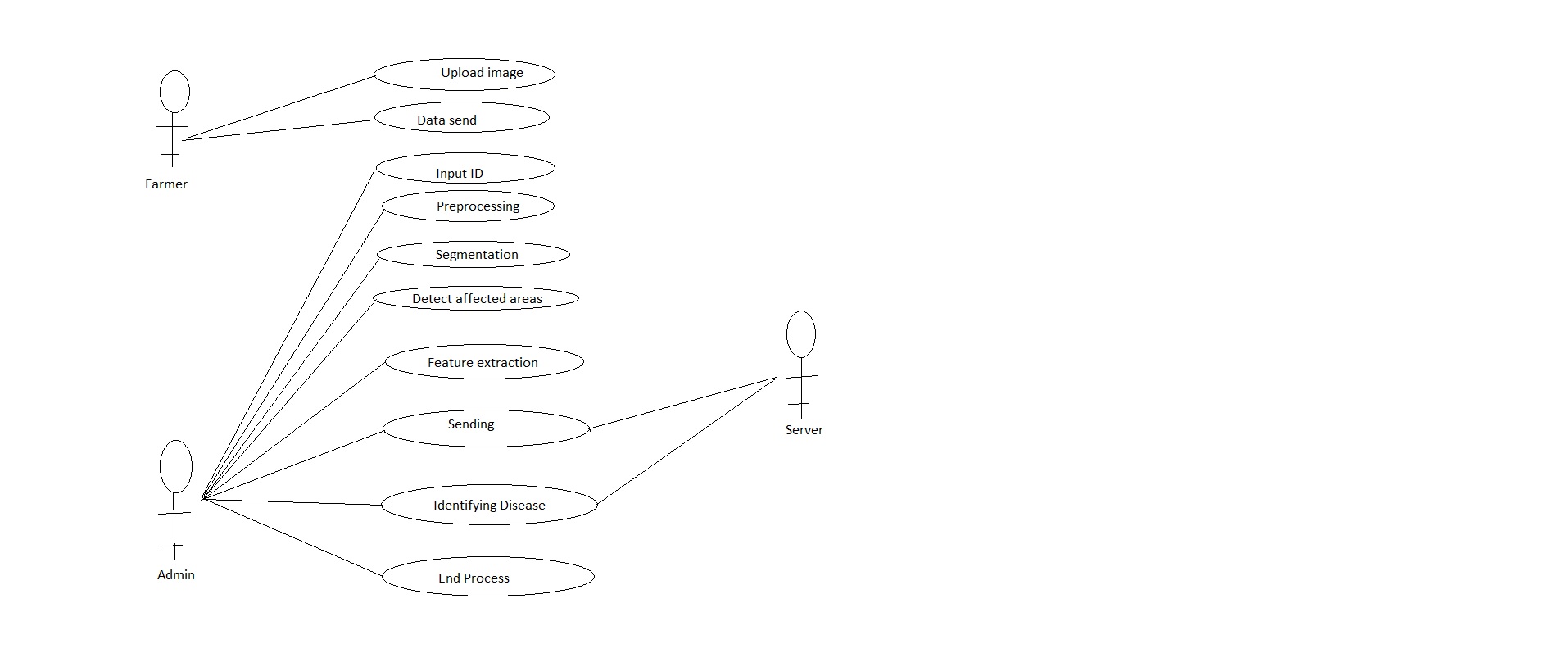
****

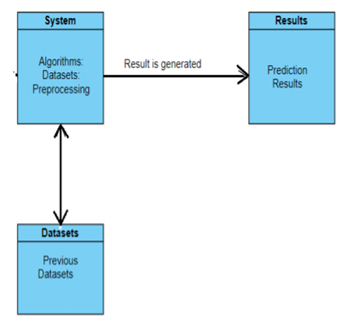
Fig -4: Block diagram

**2.3.3 USE CASE**

****

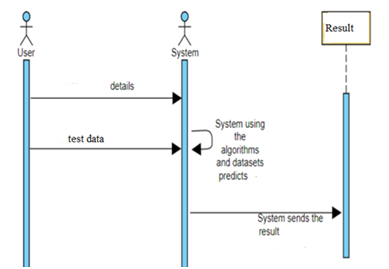
**Fig-5 : Use case diagram**

**2.3.4 CLASS DIAGRAM**

****

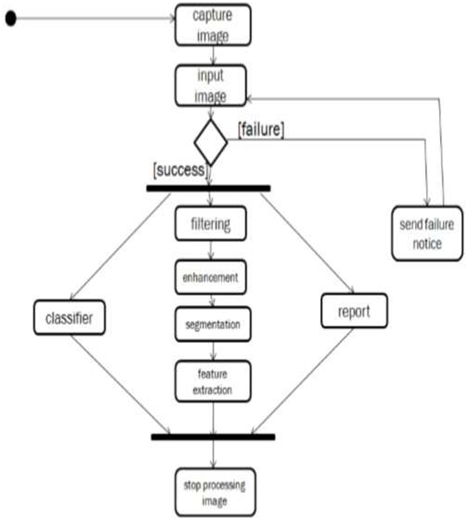
**Fig-6:Class diagram**

**2.3.5 SEQUENCE DIAGRAM**

****

**Fig-7:Sequence diagram**

**2.3.6 ACTIVITY DIAGRAM**

****

**Fig-8: Activity diagram**

**2.4 TEST CASES**

**2.4.1 Bacterial leaf blight**

****

Fig-9:Test case 1

**2.4.2 Leaf blast**

****

Fig-10:Test case 2

**2.4.3 Brown spot**

****

Fig-11:Test case 3

**2.5 RESULTS & DISCUSSIONS**

**2.5.1 RESULT**

After we ran our data set and created the model from the given data,we found the output that we wanted.We compared specific disease with specific disease which gave about 90% accuracy and also compared one disease vs another disease leaves which gave about 60%-79% of matching accuracy. The final output chart has given below

|  |  |
| --- | --- |
| **One Disease Vs.AnotherDisease** | **Specific Disease vsspecificDisease** |
| 60%-79% | 80%-100% |

So from the table it is clear that when we get an output comparison value about 60% it's the worst stage and 79% it’s still edible. When it's above 80 it's good and when it is above 80% we can sayitis fresh and healthy. The Accuracy Loss Graph is shown in the fig

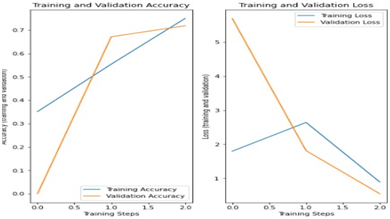


Fig -12: Result Graph

**2.5.2 DISCUSSION**

After revising our model and dataset we can say that this project has concluded that this classifier can be used in any kind of comparing dataset to compare and predict the accuracy among them. Successfully defined the accuracy of 89% of specific disease and specific disease (brown spot). It is also able to differentiate between one disease and another disease in a range of 60% to 79%.So,we will be able to predict the range of infection and how much it is from starting very bad at 60% and to mildly rotten at 79%.

**3. CONCLUSION & RECOMMENDATIONS**

## 

## 3.1 SUMMARY

## The project is developed for finding the diseases of rice leaves . For this project we needed image data. So data sets are collected from a trusted source that is an agricultural research station. After that data pre-processing rules are maintained for all the data to make them compatible with the systems environment. Datasets are trained for data handling purposes. Heresomeofdata sets are trained data sets and some test data sets.

## 

## 3.2 CONCLUSION

## In this era of Technology data brought us new opportunities as well as new complexity.Handling new data requires new methods, sometimes new technology. Finding healthy & unhealthy rice or plants one by one for the government agricultural inspector is very difficult and time consuming. It's Also difficult and a waste of time for farmers & owners of the farms to find inappropriate plants. So this is simple and can solve those problems.This System can be simple but efficient.

**3.3 RECOMMENDATIONS**

Several adaptive algorithms may develop to identify objects from image data. Because our future is AI based and image data in machine learning is very important for the AI sector. It makes our machine or technology more productive. So every one should need to work with an image classifier. It may change the entire concept of what you have understood earlier.

## 

## 3.4 FUTURE SCOPE

## To make this more efficient shall make more data.

## We can use more types or classes of data.

## In the future we will build a complete open source working platform with huge amounts of image data.

* In future we are also planning to provide remedies for the disease detected.
* We also are planning to add the soil types so that we can give crop recommendations depending on the soil type.

**4. APPENDICES**

**4.1 CLASSIFIERS USED**

Supervised classification algorithms were applied on the Rice Leaf Disease Dataset to detect three diseases of rice leaf. In this work, four classification algorithms were applied to detect the diseases. At first we applied classification algorithms before attributes selection and achieved different results for four algorithms. After that we applied classification algorithms using five selected relevant attributes with applying 10-fold cross validation and achieved better results.

1) Logistic Regression: Logistic regression can only be applied if the target class has categorical values. As the aim was to predict and categorize the disease of the affected rice leaf,logistic regression was a suitable model to train our dataset with. This paper works on predicting three distinct diseases,so we used multiclass logistic regression. In multiclass logistic regression, for given iclasses, different binary classifiers(i)\_ x are trained for each class i to determine the probability of the target class. Then, a new input x can be predicted to belong to the class if it maximizes mach(i)

\_ x:

h\_(x) = g(\_T x) =

1

1 + e􀀀\_T x (1)

where,

g(\_T x) = g(z) =

1

1 + e􀀀z (2)

h\_ is the hypothesis that determines the predicted output; y is predicted to be 1 if h\_ is greater or equal to 0.5 and it is predicted to be 0 if h\_ is less than 0.5. g(z) maps real valued numbers within a range of 0 to 1 and it plots an S-shape curve

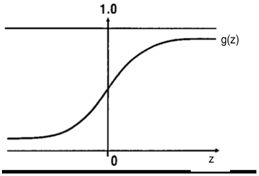


Fig-13:S-Shape Curve

Here we performed 10-fold cross validation using logistic regression algorithm and achieved 75.463% accuracy on training set and 70.8333% accuracy on test set in detecting three

**4.2 K-NEAREST NEIGHBOUR**

Like logistic regression, KNN also works well for discrete target classes. It calculates the distances of the query point from each of the instances and finds the K minimum distances, that is, it determines the K nearest neighbours for the query point from which it can predict the class of the query point. The value of K needs to be chosen by inspecting the data; in case, we found when K = 1 the accuracy is 98.8426% on training set and 91.6667% on testing set after performing 10-fold cross validation.And when K=3 the accuracy is 85.6481% on training set and 72.9167% on testing set after performing 10- fold cross validation. We found, if the value of K is increased then accuracy is decreased.

**4.3 DECISION TREE**

Decision tree is one of the most commonly used machine learning classifiers. Taking the best suitable attribute at the root, this algorithm breaks the dataset into partitions. The goal of the partition is to unmix the dataset. The splitting iterates until eventually the partitions group the data such that they are homogeneous. Iterative dichotomiser 3 (ID3), which uses a greedy approach, is the core algorithm for decision trees. In this approach, entropy and information gain, concepts borrowed from information theory, are used for constructing the tree. Entropy measures the impurity of arbitrary attributes; zero entropy means all instances belong to 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 24-25 December, Dhaka in the same class. As entropy becomes more and more positive, the instances become more and more heterogeneous.

E =

Xc

i=1

􀀀pi log2 pi (3)

Here c is the number of classes.

Information gain allows the attribute to be selected as the next node in the tree. The attribute with the most information gain would be selected for this purpose.

Gain(S;A) = Entropy(S) 􀀀

XjSvj

jSj

Entropy(Sv) (4)

Here, A is the known attribute and Svis the subset of A for which, A has the value v. Using five selected attributes, the decision tree algorithm was able to correctly classify 94.9074% data on a training set where 10-fold cross validation is performed. The model achieved 97.9167% accuracy on test data.

**4.4 NAIVE BAYES CLASSIFIER**

Naive Bayes algorithm is a probabilistic algorithm that is based on Bayes theorem. Based On this theorem, the best hypothesis is chosen based on equation 5

^y = argmax(y)

Yn

i=1

(P(xiji)) (5)

In this work Naive Bayes algorithm achieved the lowest accuracy to correctly classify three diseases.

**4.5 PYTHON LIBRARIES**

1. Requests. The most famous http library written by kennethreitz. It’s a must have for every python developer.

2. NumPy. It is for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices.

3. Matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.

4. Keras. It allows useof distributed training of deep-learning models on clusters of Graphics processing units

5. Tensorflow. It is an open-source library for numerical computation and large-scale machine learning that eases Google Brain.

6. sklearn. Thelibrary contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

7. Twisted. The most important tool for any network application developer. It has a very beautiful api and is used by a lot of famous python developers.

8. NumPy. How can we leave this very important library ? It provides some advanced math functionalities to python.

9. SciPy. When we talk about NumPy then we have to talk about scipy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. openCv. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

11. argparse. This module makes it easy to write user-friendly command-line interfaces.

**4.6 EXPERIMENTAL SETUP**

In our model implementation and code implementation .We have collected the data first. Procedure given below

• As we worked with disease detection(bacterial leaf blast,brown spot)we had to collect Both Good and bad leaf images from different areas.

• For a larger part of our project we have collected images from the farm, field and from agricultural research station

• We also scraped the internet to find out more images about this topic but couldn’t find much information.

• Then we have finalized and normalized the data in order to perform our training.

• After labelling data data was usable and good for further processes.After that we had to preprocess our image in below steps.

• First we cropped all image With Same Dimensions

• Then we had to resize and reshape all image in order to normalize dataset

• Then we used a pooling layer to remove all negative pixels to make final usable images.

**4.7 CODE**

**4.7.1 RICE ALGORITHM**

from tensorflow.keras.layers import Input,Lambda,Dense,Flatten

from tensorflow.keras.models import Model

from tensorflow.keras.preprocessing import image

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

import numpy as np

from glob import glob

import matplotlib.pyplot as plt

IMAGE\_SIZE = [224,224]

train\_path = "Rice/Train/"

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale=1./255,horizontal\_flip=True,zoom\_range=0.2,validation\_split=0.15)

training\_set = train\_datagen.flow\_from\_directory(

train\_path,target\_size=(224,224), batch\_size=32,class\_mode='categorical',

subset='training')

validation\_set = train\_datagen.flow\_from\_directory(

train\_path,target\_size=(224,224), batch\_size=32,class\_mode='categorical',shuffle = True,

subset='validation')

from tensorflow.keras.applications import VGG19

from tensorflow.keras.layers import GlobalAveragePooling2D,Dropout

mv = VGG19(input\_shape=IMAGE\_SIZE+[3],weights='imagenet',include\_top=False)

for layers in mv.layers:

layers.trainable = False

x = Flatten()(mv.output)

prediction = Dense(3,activation='softmax')(x)

model = Model(inputs=mv.input,outputs=prediction)

model.summary()

import tensorflow as tf

class myCallback(tf.keras.callbacks.Callback):

def on\_epoch\_end(self,epoch,logs={}):

if(logs.get('loss')<=0.05):

print("\nEnding training")

self.model.stop\_training = True

callbacks = myCallback()

from tensorflow.keras.optimizers import Adam

model.compile(optimizer=Adam(lr=0.0001),loss='categorical\_crossentropy',metrics=['categorical\_accuracy'])

history = model.fit(training\_set,

validation\_data=validation\_set,

epochs=50,

verbose=1,

steps\_per\_epoch=len(training\_set),

validation\_steps=len(validation\_set),

callbacks = [callbacks]

)

acc = history.history['categorical\_accuracy']

val\_acc = history.history['val\_categorical\_accuracy']

loss = history.history['loss']

val\_loss = history.history['val\_loss']

epochs = range(len(acc))

import matplotlib.pyplot as plt

plt.plot(epochs,acc)

plt.plot(epochs,val\_acc)

plt.title("Training and validation Accuracy")

plt.figure()

plt.plot(epochs,loss)

plt.plot(epochs,val\_loss)

plt.title("Training and validation Loss")

plt.figure()

model.save("rice1.h5")

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import numpy as np

img\_width, img\_height = 224,224

model = load\_model('rice1.h5')

img=image.load\_img('FruitsDB/Test/bacterial\_leaf\_blight/1.jpg', target\_size=(img\_width, img\_height))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

classes = model.predict(x)

print (classes)

**4.7.2 APPLICATION**

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing.image import load\_img, img\_to\_array

import numpy as np

from flask import Flask, request, render\_template

from werkzeug.utils import secure\_filename

import os, sys, glob, re

app = Flask(\_\_name\_\_)

model\_path = "rice.h5"

classes = {0:"bacterial\_leaf\_blight:-{ About bacterial\_leaf\_blight disease }",1:"blast:-{ about blast disease} ",2:"brown spot:-{ about brown spot disease }"}

def model\_predict(image\_path):

print("Predicted")

image = load\_img(image\_path,target\_size=(224,224))

image = img\_to\_array(image)

image = image/255

image = np.expand\_dims(image,axis=0)

model = load\_model(model\_path)

result = np.argmax(model.predict(image))

prediction = classes[result]

if result == 0:

print("bacterial\_leaf\_blight.html")

return "bacterial\_leaf\_blight","bacterial\_leaf\_blight.html"

elif result == 1:

print("blast.html")

return "blast", "blast.html"

elif result == 2:

print("brownspot.html")

return "brownspot" , "brownspot.html"

@app.route('/',methods=['GET'])

def index():

return render\_template('index.html')

@app.route('/predict',methods=['GET','POST'])

def predict():

print("Entered")

if request.method == 'POST':

print("Entered here")

file = request.files['image'] # fet input

filename = file.filename

print("@@ Input posted = ", filename)

file\_path = os.path.join('static/user uploaded', filename)

file.save(file\_path)

print("@@ Predicting class......")

pred, output\_page = model\_predict(file\_path)

return render\_template(output\_page, pred\_output = pred, user\_image = file\_path)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True,threaded=False)

**4.7.3 OUTPUT CHECK**

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications import VGG16

from tensorflow.keras.layers import AveragePooling2D

from tensorflow.keras.layers import Dropout

from tensorflow.keras.layers import Flatten

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Input

from tensorflow.keras.models import Model

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.utils import to\_categorical

from sklearn.preprocessing import LabelBinarizer

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

from imutils import paths

import matplotlib.pyplot as plt

import numpy as np

import argparse

import cv2

import os

from tensorflow.keras.preprocessing import image

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import numpy as np

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import numpy as np

img\_width, img\_height = 224,224

model = load\_model('rice.h5')

img = image.load\_img('Rice/Train/brownspot/3.jpg', target\_size=(img\_width, img\_height))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

classes = model.predict(x)

print (classes)

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

result = np.argmax(model.predict(x))

print(result)

**4.7.4 HTML PAGES**

**4.7.4.1 Bacterial leaf blight**

<!DOCTYPE HTML>

<!--

Aerial by HTML5 UP

[html5up.net](http://html5up.net/) | @ajlkn

Free for personal and commercial use under the CCA 3.0 license ([html5up.net/license](http://html5up.net/license))

-->

<html>

<head>

<title>Rice Disease Type (bacterial\_leaf\_blight)</title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />

<link rel="stylesheet" href="static/css/Alluvial.css" />

<noscript><link rel="stylesheet" href="static/css/noscript.css" /></noscript>

</head>

<body class="is-preload">

<div id="wrapper">

<div id="bg"></div>

<div id="overlay"></div>

<div id="main">

<!-- Header -->

<header id="header">

<h1>Rice Disease Type : bacterial\_leaf\_blight </h1>

</br>

<nav>

<ul>

</ul>

</nav>

</header>

<!-- Footer -->

<footer id="footer">

</footer>

</div>

</div>

<script>

window.onload = function() { document.body.classList.remove('is-preload'); }

window.touchmove = function() { return false; }

window.onorientationchange = function() { document.body.scrollTop = 0; }

</script>

</body>

</html>

**4.7.4.2 Blast**

<!DOCTYPE HTML>

<!--

Aerial by HTML5 UP

[html5up.net](http://html5up.net/) | @ajlkn

Free for personal and commercial use under the CCA 3.0 license ([html5up.net/license](http://html5up.net/license))

-->

<html>

<head>

<title>Rice Disease Type (blast)</title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />

<link rel="stylesheet" href="static/css/Black.css" />

<noscript><link rel="stylesheet" href="static/css/noscript.css" /></noscript>

</head>

<body class="is-preload">

<div id="wrapper">

<div id="bg"></div>

<div id="overlay"></div>

<div id="main">

<!-- Header -->

<header id="header">

<h1>Rice Disease Type : blast </h1>

<p>About blast disease </p>

</br>

<nav>

<ul>

</ul>

</nav>

</header>

<!-- Footer -->

<footer id="footer">

</footer>

</div>

</div>

<script>

window.onload = function() { document.body.classList.remove('is-preload'); }

window.touchmove = function() { return false; }

window.onorientationchange = function() { document.body.scrollTop = 0; }

</script>

</body>

</html>

**4.7.4.3 Brown Spot**

<!DOCTYPE HTML>

<!--

Aerial by HTML5 UP

[html5up.net](http://html5up.net/) | @ajlkn

Free for personal and commercial use under the CCA 3.0 license ([html5up.net/license](http://html5up.net/license))

-->

<html>

<head>

<title>Rice Disease Type (brown spot) </title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />

<link rel="stylesheet" href="static/css/Red.css" />

<noscript><link rel="stylesheet" href="static/css/noscript.css" /></noscript>

</head>

<body class="is-preload">

<div id="wrapper">

<div id="bg"></div>

<div id="overlay"></div>

<div id="main">

<!-- Header -->

<header id="header">

<h1>Rice Disease Type : brown spot </h1>

<p>About brown spot disease </p>

</br>

<nav>

<ul>

</ul>

</nav>

</header>

<!-- Footer -->

<footer id="footer">

</footer>

</div>

</div>

<script>

window.onload = function() { document.body.classList.remove('is-preload'); }

window.touchmove = function() { return false; }

window.onorientationchange = function() { document.body.scrollTop = 0; }

</script>

</body>

</html>

**5. REFERENCES**

1. Chawathe, S. S. (2020). Rice Disease Detection by Image Analysis.2020 10th Annual Computing and Communication Workshop and Conference(CCWC). (Base paper)
2. Pukkila, P. and Borra, S., 2018. Machine Learning Based Plant Leaf Disease Detection and Severity Assessment Techniques: State-of-theArt.In Classification in BioApps. (Springer)
3. Zhang, X., Qiao, Y., Meng, F., Fan, C. and Zhang, M., 2018. Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. (IEEE)
4. Zhou Z., Zang Y., Li Y., Zhang Y., Wang Luo X.

Rice plant-hopper infestation detection and classification algorithms based on fractal dimension values and fuzzy C-means Mathematical and Computer Modelling, 58 (3-4) (2013), pp. 701-709

1. Yao Q., Xian D., Liu Q., Yang B., Diao G., Tang J.

Automated Counting of Rice Planthoppers in Paddy Fields Based on Image Processing Journal of Integrative Agriculture, 13 (8) (2014)

1. Pukkela P., Borra S.

Machine Learning Based Plant Leaf Disease Detection and Severity Assessment Techniques: State-of-the-Art Classification in BioApps, Springer, Cham (2018)

1. Huang T., Yang R., Huang W., Huang Y., Qiao X.Detecting sugarcane borer diseases using support vector machine Information processing in agriculture, 5 (1) (2018)
2. Anthonys, G. and Wickramarachchi, N., 2009. An image recognition system for crop disease identification of paddy fields in Sri Lanka, 2009 International Conference on Industrial and Information Systems (ICIIS)
3. Pugoy RADL, Mariano V.Y., 2011. Automated rice leaf disease detection using color image analysis. In 3rd international conference on digital image processing, volume 8009. Chengdu: SPIE; F1-F7.
4. Verma, P., (2017). Rice Productivity and Food Security in India, Centre for Management in Agriculture, Indian Institute of Management Ahmedabad.
5. Rewar, E., Singh, B.P., Chhipa, M.K., Sharma, O.P., and Kumari, M., 2017. Detection of Infected and Healthy Part of Leaf Using Image Processing Techniques.
6. Sarangdhar, A.A. and Pawar, V.R., 2017. April. Machine learning regression technique for cotton leaf disease detection and controlling using IoT. In Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE. 2, 449-454.
7. Gajanan, D.E., Shankar, G.G. and Keshav, G.V., 2018. Detection of Leaf Disease Using Feature Extraction for Android Based System.
8. Phadikar S., Sil J, Das A.K.Rice diseases classification using feature selection and rule generation techniques Comput. Electron. Agric., 90 (2013)
9. Lorstat BPornpanomchai C.

Image analysis based on color, shape and texture for rice seed(Oryza sativa L)germination evaluation. Agriculture and Natural Resources, 51 (2017)

**6. GLOSSARY**

|  |  |
| --- | --- |
| **WORD** | **Pg.No** |
| **EPOCHS :**  An epoch is a term used in machine learning and indicates the number of passes of the entire training dataset the machine learning algorithm has completed | **2** |
| **RGB VALUE:**  Color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors | **4** |
| **VGG 19:**  VGG19 is a variant of the VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layers, 5 MaxPool layers and 1 SoftMax layer). | **7** |
| **MAX POOLING:**  Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter | **8** |
| **SOFTMAX:**  Softmax activation is normally applied to the very last layer in a neural net, instead of using ReLU, sigmoid, tanh, or another activation function | **8** |
| **LEARNING RATE:**  In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function. | **27** |