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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.12630

Detection of Okra Disease

Shraddha C[1], Maanyatha M[2], Suloni Praveen[3], Supriya T C[4], Swathi Meghana K R[5]

Assistant Professor, Computer Science and Engineering, Vidyavardhaka College of Engineering, Mysuru, India[1]

Student, Computer Science and Engineering, Vidyavardhaka College of Engineering, Mysuru, India[2-5]

Abstract: The okra crop, commonly called lady's finger, is a commercial crop grown by farmers. But when these crops are infected, they wreak havoc on the life of farmers as well as on the economy. Plant disease issues are due to problems arising in agricultural activities and climate change. It causes money problems and losses to profits, farmers, and to whole industry depending on okra. If disease identification is done without appropriate techniques, then the problem remains the same. Various research has proposed several techniques to overcome these infections. CNN, Deep learning, and many other machine-learning techniques are used for detecting and classifying plant diseases.

Keywords: Okra, CNN, Deep learning, Machine learning, detecting, classifying

I. INTRODUCTION

One of the most important and fundamental pillars of any economy, and perhaps more so in India, is the agricultural sector. One of India's largest industries, agriculture employs a sizable portion of the country's people.

Fig. 1 OKRA Plant

Farmers that are on the cutting edge of agriculture typically rely on visual inspection and identification of various crop illnesses and problems. The okra crop, commonly called lady's finger, is a commercial crop grown by farmers**.** Crop experts employ the same procedure with a bit more expertise.

However, both farmers and experts are only capable of precisely identifying a certain number of crops. Despite the use of methods that appear to be infallible, there is a chance that the outcomes will be inaccurate; they may have revealed improper assessment or be incorrect diagnosis. This takes time and wastes material and financial resources, which reduces the effectiveness of crop production, has an impact on farmers' investments, and could be seen as a blow to the farming industry and the country's economy. It thus reduces crop growth further, creating a shortage in the market and inflation in price. The use of image processing methods in automatic detection may prove to be a useful strategy that produces quicker and better results. Deep learning and the use of recognition models are simply two of the few contemporary methods that could deliver an ideal outcome.
In this project, we'll employ deep convolutional networks, which are typically used to classify images of leaves, to try to tackle the challenge of detecting and identifying various crop/plant illnesses.

Digital image processing is the process of applying various operations to a picture in order to gain useful information from it or create a better image for additional analysis. It permits the use of more algorithms on the input data and can lessen processing noise and signal distortion. Image processing is a method for applying some action to a picture to enlarge it or extract important information. It is comparable to signal processing in that an image serves as the input, and the result could either be another image or the attributes of that image.

Image processing is a method for converting a physical image into a digital one so that you can edit, add to, or remove information from it. An image, such as a photo or image sequence, acts as the input in this type of signal distribution, while the output can either be another image or the attributes of the original image. The following steps are significantly involved in image processing:

1. Importing the image using image acquisition software.
2. Reviewing and adjusting the image.
3. The output, which may be a revised image, or a report based on the analysis of the original image.

The application of computational models to the image processing of photographic files is referred to as digital image processing in computer science. Contrary to analog image processing, it enables the application of a larger variety of algorithms to the data entered and can also prevent problems like the buildup of noise and signal distortion during processing. It provides a platform for a selection of processes, such as the processing of analog and digital signals, picture signals, speech signals, and image signals.

Digital image processing can be treated as a multidimensional system because images are specified over two dimensions. The three primary reasons that influence the development of digital image processing are the evolution and development of computers, the development of mathematics, and the demand for a wide range of technology imaging services.

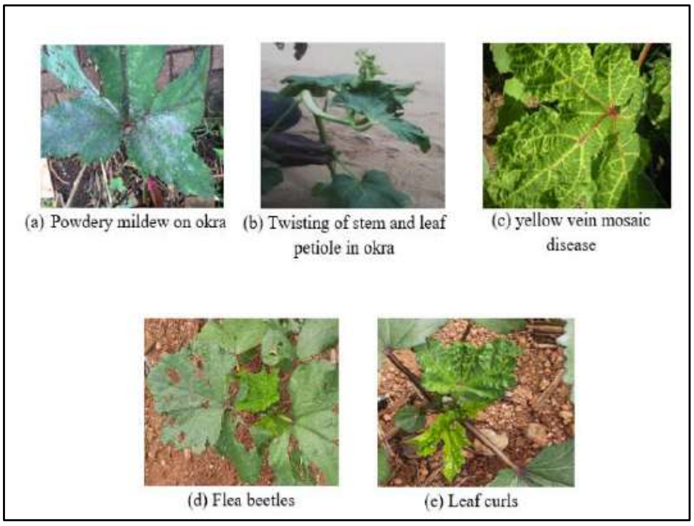
Image restoration (CT, MRI, PET), image reformatting (multi-plane, multi-view reconstruction), quick and easy image storage and retrieval, and images with high distribution are only a few of the benefits of digital image processing.

There is furthermore viewed viewing (windowing, zooming). The time-intensive nature of digital image processing is one of its drawbacks. Depending on the system, it can be very expensive. It is only accessible to qualified users [1].

II. PROPOSED METHODOLOGY

A. Dataset

Carrying out disease/fungal infection detection during the early stage and taking precautionary measures abrupt from spreading to other parts of the crop. Generally, farmers monitor the color and shape of plant leaves for disease detection from time to time and interact with plant pathologists for the right recommendations for the use of fertilizers. This process is time-consuming and is not practical in certain situations. Variations in signs, spots, color, and other characteristics can be tracked to identify infections in various parts of the plant. In this context, there exists a need of an automated mechanism to perform timely monitoring of fungal infections through ease-of-use hand-held devices such as mobiles. If these infections are treated at an earlier stage and followed the right method of yielding, we can overcome the loss that farmers are facing. To enable this, we want to develop a system that is user-friendly and can work with the help of handheld devices like mobile phones and can automatically detect what type of disease the particular plant is infected with and how to treat those diseases.


(a) Powdery mildew on okra (b) Twisting of stem and leaf petiole in okra (c) yellow vein mosaic disease

(d) Flea beetles (e) Leaf curls

Fig. 2 Sample images of various infections on Okra plant

DOI: 10.17148/IJARCCE.2023.12630

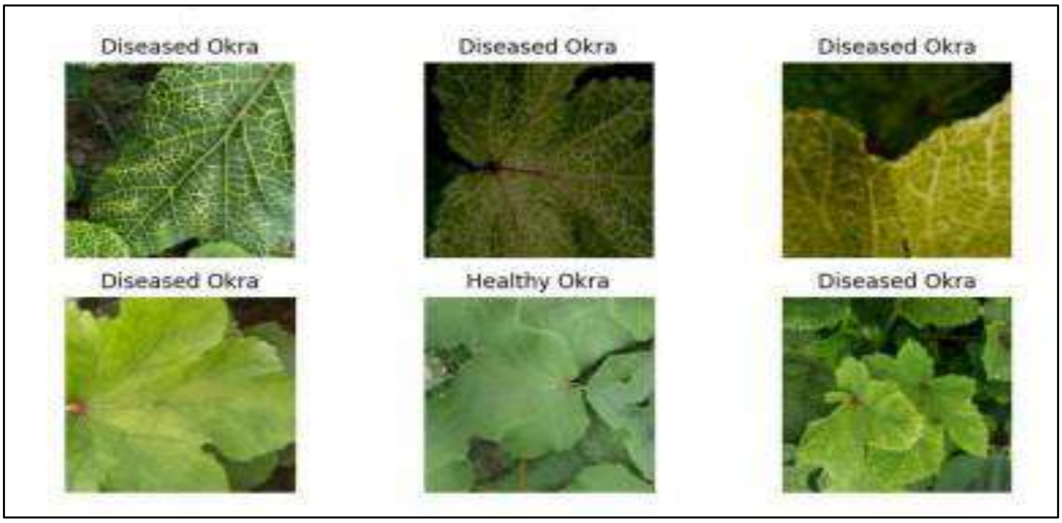


Fig. 3 Dataset Images

The dataset is obtained from Kaggle - https://www.kaggle.com/datasets/manojgadde/yellow-vein-mosaic-disease. A total of 1949 images have been used with a resolution of 224×224 pixels. About 80% of the dataset (587 images) is used for training, 15% of the dataset (247 images) is used for testing and 5% of the dataset(88 images) is used for validation. Healthy Okra and Diseases Okra are the two classes used here.

B. Image Acquisition

We used 6 mobile phones with camera resolutions of 48MP, 48MP,48MP,12MP,9MP, and 13MP respectively. On 30th October 2022, we visited a village named Hadinaru near Nanjangudu taluk for clicking images, we took permission from the landlord and captured around 2500 to 2700 pictures. Captured images vary in dimension from a minimum of 1800x4000 to a maximum of 4128x2322. The images are taken with variable distances approximately 20 to 50cm, as the brightness of the phone was at maximum to capture images with proper light setting the image contrast is least here. We not only clicked images but also recorded clips and then using recorded videos we sliced them into images for further usage. Two days before they had sprayed pesticides for the crops which are used to remove cribs and for folded leaves. Fig. 4 supporting to the explained process of data acquisition.

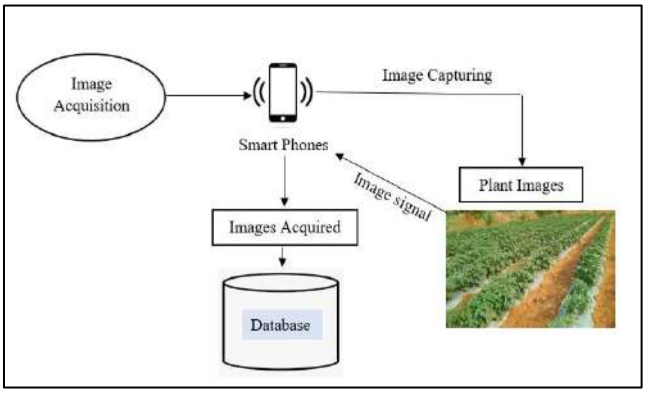


Fig. 4 Process of Image Acquisition

We adopted a process called Hue Saturation Value (HSV) color thresholding in MATLAB. We tested with different value parameters to check if they meet our expectations. Before choosing HSV, we tried the same with RGB, YCbCr, and L*a*b color thresholding in MATLAB; after setting variable parameters, we could see the results of all these techniques where HSV upheld the proper image for the particular dataset we chose.

Before any further steps, the acquired images are pre-processed. To reduce computational complexities, the background of all captured RGB images is blacked out and converted into grey-level images in this phase by setting parameters in HSV for a single image, we extracted the same function and post-processed the remaining images manually using MATLAB code. For the U-net model, both the original and grayscale images were used as input.

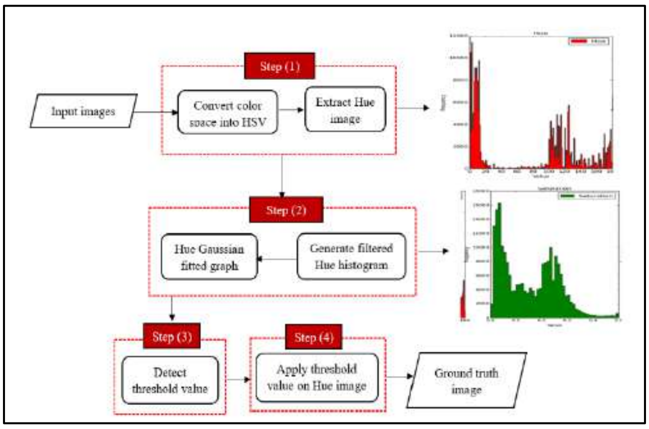

Input images
Step (1)
Convert color space into HSV
Extract Hue image
Step (2)
Hue Gaussian fitted graph
Generate filtered Hue histogram
Step (3)
Detect threshold value
Step (4)
Apply threshold value on Hue image
Ground truth image

Fig. 5 Block diagram for HSV thresholding process

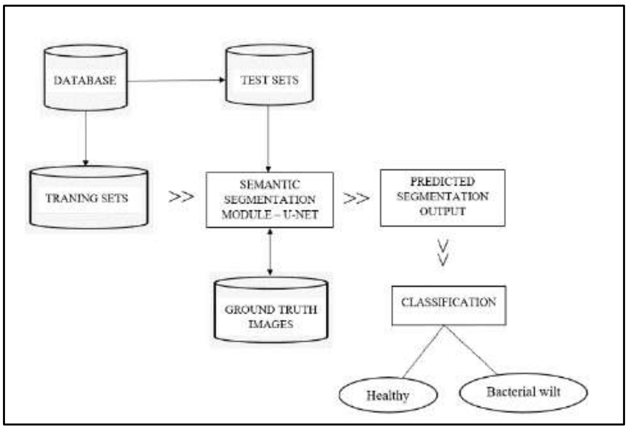


Fig. 6 Architecture of the proposed method for early-stage disease detection in Abelmoschus Esculentus

C. Semantic Segmentation – U-Net

U-net has been used to segment Abelmoschus Esculentus in a complex environment from multi-resolution images. The goal of semantic segmentation is to assign a categorical label to each pixel using U-Net, which consists of two paths, one of which contracts and the other of which expands. U-Net can recognize and localize pixel features, which will aid in boundary discrimination from one object to the next. The result of semantic segmentation for an RGB input image is a binary image that is commonly referred to as a mask.

The proposed methodology aims to segment leaf regions from a variety of complex environments and classify them as healthy or bacterial wilt-infected plants in the early stages. The database is first divided into training and test sets. The ground-truth creation method is also applied to training sets, resulting in a database of ground truths corresponding to training sets. The semantic segmentation module uses U-net to process training sets with their corresponding ground truths. To achieve the segmented result, further test sets are used as input to the semantic segmentation module. The Resnet18 model is used for classification [9]. The U-net architecture for semantic segmentation is shown in Fig. 8.

The down-sampling path is made up of four encoder blocks, each of which consists of two consecutive convolution layers with ReLU followed by a max pooling operation to complete the down-sampling. Down-sampling reduces the dimensions of the input image to half in each encoder block because of the contracting path. In addition, the number of filters used for convolutions will be doubled from one encoder block to the next, increasing the depth of the down-sampled input image as illustrated in Fig. 7. The results of converting acquired images to predicted labels are shown in Table I.

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.12630

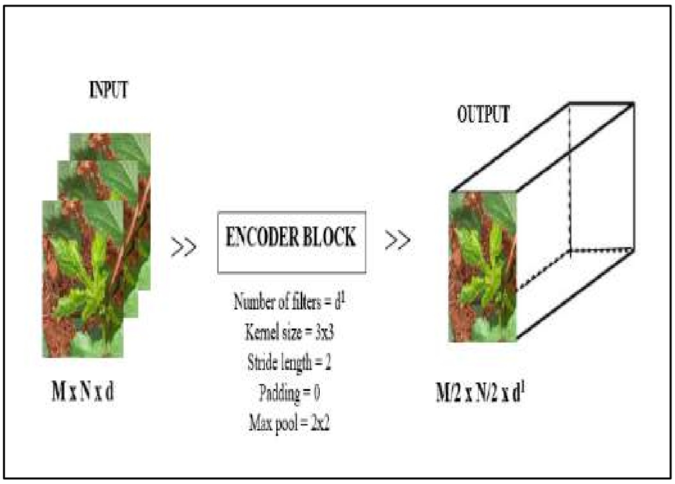


Fig. 7 Down sampling of input image in contracting path

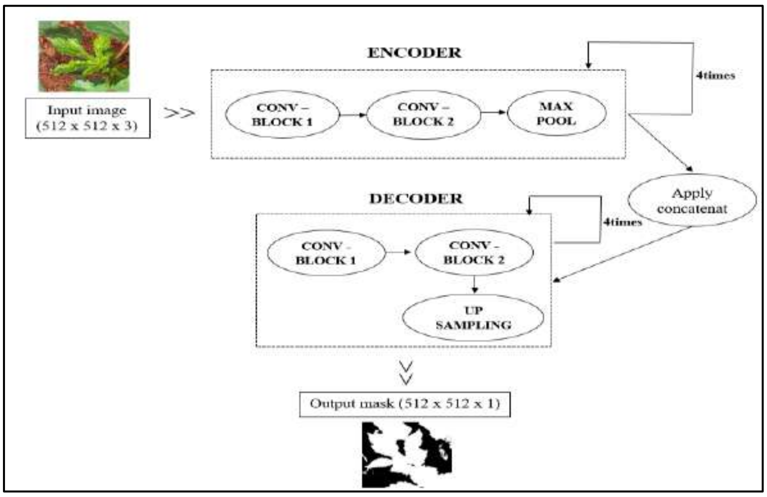
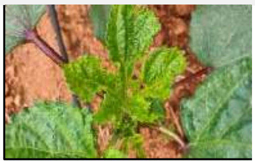
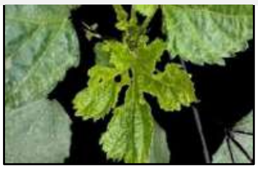
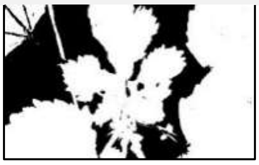
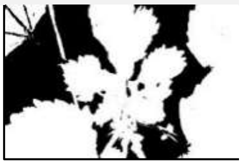
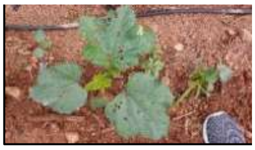
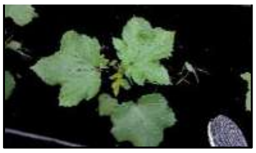
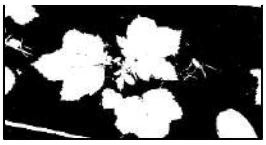
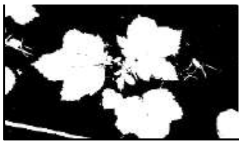
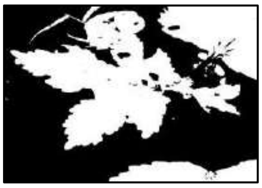


Fig. 8 Architecture of semantic segmentation module -U-Net

TABLE I OUTCOME OF SEMANTIC SEGMENTATION USING U-NET

Image number	Original Image	Segmented using HSV thresholding	Binary Mask- Segmented using HSV thresholding	Predicted output using U-Net
1.				
2.				

DOI: 10.17148/IJARCCE.2023.12630

3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				

D. Classification – Resnet18

Renet-18 is a convolutional neural network that is 18 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

In Fig. 9 we use the Resnet 18 model which consists of 18 layers. These connections primarily function by hopping over one or multiple layers forming shortcuts between these layers. This Classification Model is used to classify whether the image acquired by the machine contains an image of a Healthy Okra or a Diseased Okra.

DOI: 10.17148/IJARCCE.2023.12630

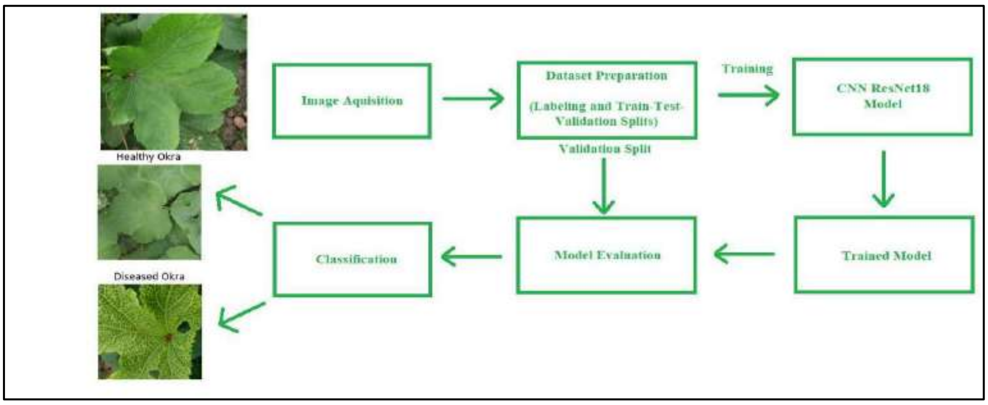


Fig. 9 System Architecture with Resnet18 Classification Model

III. RESULTS AND DISCUSSIONS

The proposed method evaluates segmentation and classification modules sequentially, employing datasets of 4418 samples that include both healthy and infected variants. In both variants, around 2872 samples are considered ground truth samples. We additionally consider 877 healthy samples from the Kaggle challenge repositories' Village plant database.

The efficiency of semantic segmentation using U-Net is first evaluated with a batch size of 16, number of steps per epoch of 240, and the number of epochs of 30 for which an accuracy of 63 % is attained for a learning rate of 0.001. On the next attempt, we achieved an accuracy of 94.51 % and validation accuracy of 91.33 % with 11.78 % loss using batch size 8, a learning rate of 0.0001, and 80 epochs. In subsequent trials, with a batch size of 8, a number of steps per epoch of 100, and several epochs of 50 at a learning rate of 0.0001, the best training accuracy was 99.95%, and the best validation accuracy was 99.78%. Fig. 10 show the performance of U-Net on training and validation data in terms of accuracy and loss over 50 epochs with a learning rate of 0.0001 and 100 steps.

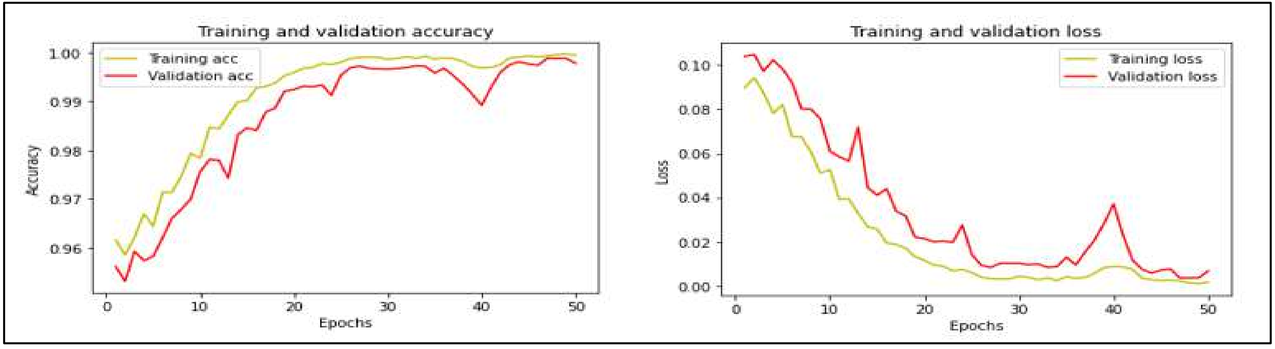


Fig. 10 Performance of U-Net on training and validation data in terms of accuracy and loss

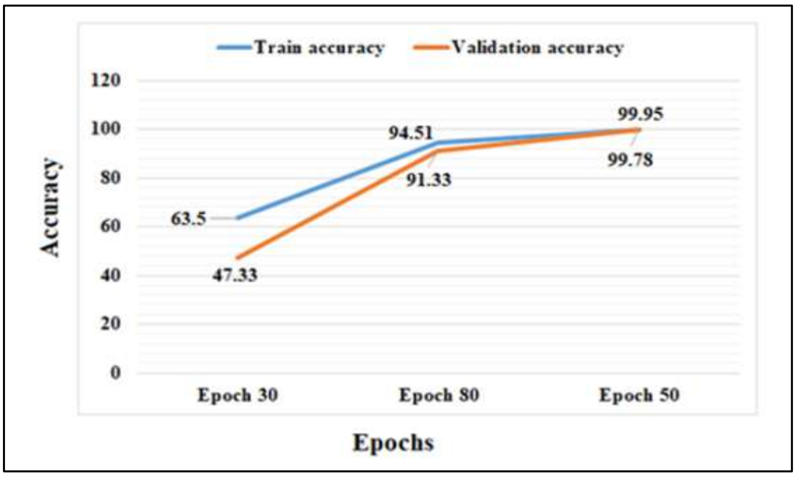


Fig. 11 Number of epochs versus accuracy – Semantic Segmentation

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.12630

Fig. 11 indicates that with 30 epochs and a batch size of 8, the learning model exhibits underfitting behavior even after 240 iterations, with a learning rate of 0.0001. When the number of epochs is increased to 80, we see that the model overfits the learning model with 240 iterations. By limiting the number of iterations to 100 and the number of epochs to 50, the behavior of the learning model towards validation data has reached an optimal state, resulting in an accuracy of better than 99 percent in both training and validation.

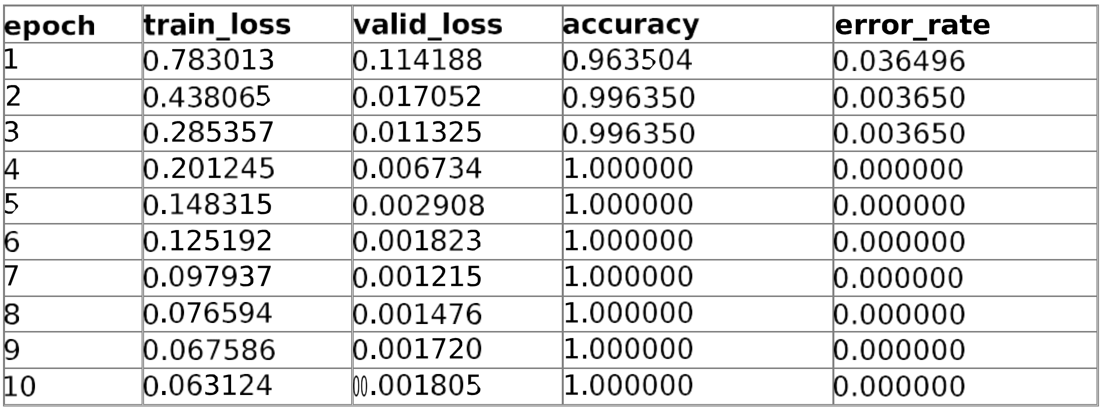
We have used Intersection over Union (IoU) [51] metrics to assess the efficiency of U-Net. IoU is a statistic for determining the accuracy of an object detector on a certain dataset. It yields a numerical result between 0 and 1, indicating how much the expected and ground truth bounding boxes overlap, as determined by (1). During segmentation, IoU values of 0.5 and above are usually considered reliable.

$$IoU = Area\ of\ Overlap\ /\ Area\ of\ Union \qquad (1)$$

In the case of semantic segmentation applying U-net, the total mean IoU computed for the test set of input samples is 0.8922.

The model was trained for 10 epochs to reach maximum accuracy and the lowest error rate. The training details are specified in Table II.

TABLE II TRAINING DETAILS



epoch	train_loss	valid_loss	accuracy	error_rate
1	0.783013	0.114188	0.963504	0.036496
2	0.438065	0.017052	0.996350	0.003650
3	0.285357	0.011325	0.996350	0.003650
4	0.201245	0.006734	1.000000	0.000000
5	0.148315	0.002908	1.000000	0.000000
6	0.125192	0.001823	1.000000	0.000000
7	0.097937	0.001215	1.000000	0.000000
8	0.076594	0.001476	1.000000	0.000000
9	0.067586	0.001720	1.000000	0.000000
10	0.063124	00.001805	1.000000	0.000000

The following Fig. 12 shows the decreased error in the case of Resnet18 as compared to other CNN-based models like inception V3.

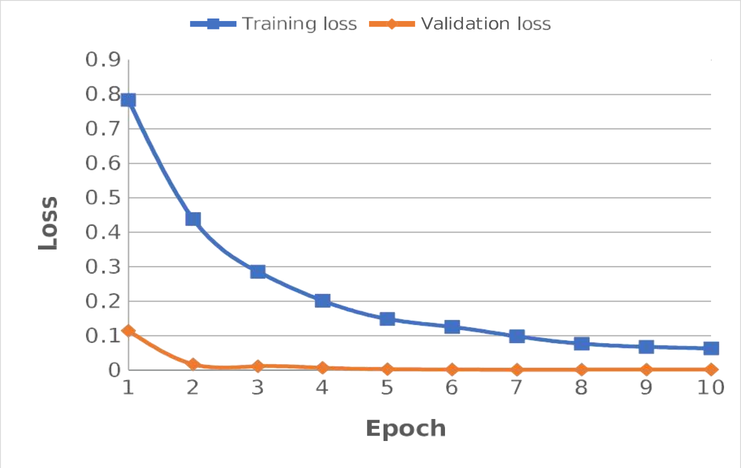


Fig. 12 Training and Validation dataset loss vs epoch

The Testing Dataset consists of a total of 274 images and the confusion matrix of the testing dataset is mentioned in Fig. 13.

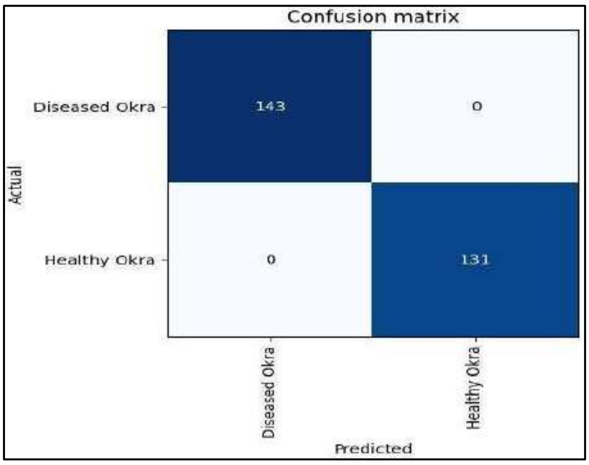


Fig. 13 Confusion matrix for the Testing Dataset

Similarly, the confusion matrix for the Validation Dataset is shown in Fig. 14. It has a total of 88 images, where 1 image has been misclassified. Hence we gained an accuracy of 98.8%.

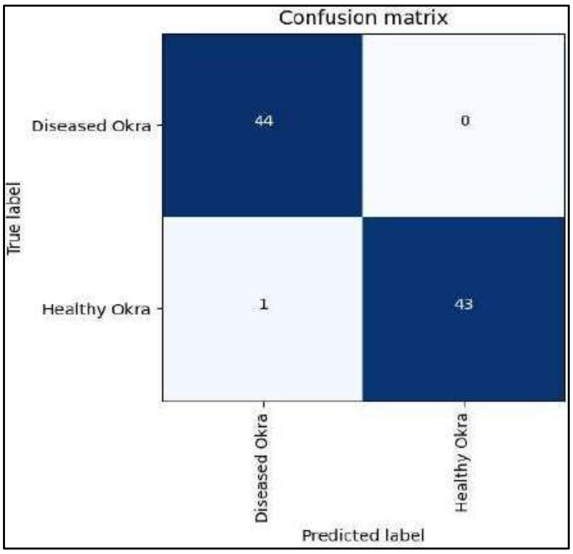
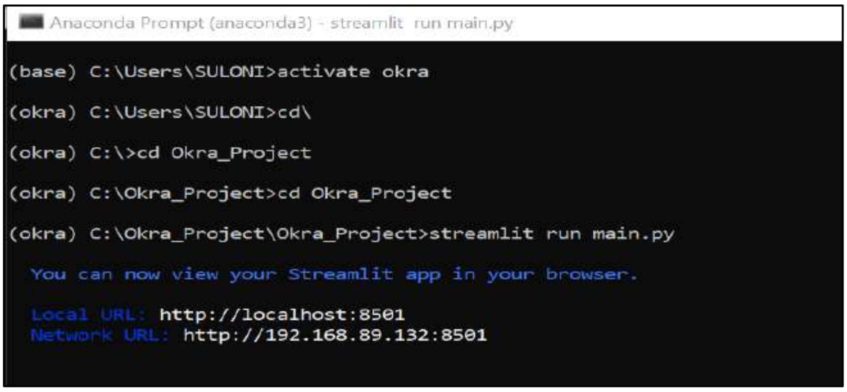


Fig. 14 Confusion Matrix for Validation Dataset

The result snapshots are stated in Fig. 15 - 23.

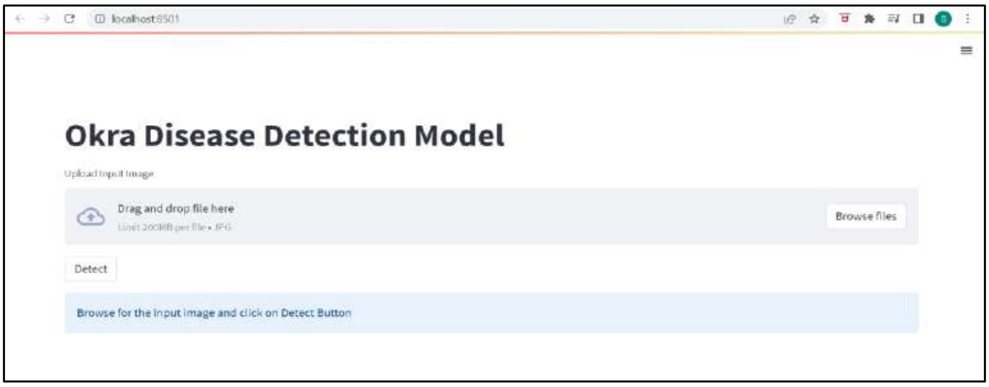


```
(base) C:\Users\SULONI>activate okra

(okra) C:\Users\SULONI>cd\

(okra) C:\>cd Okra_Project

(okra) C:\Okra_Project>cd Okra_Project

(okra) C:\Okra_Project\Okra_Project>streamlit run main.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.89.132:8501
```

Fig. 15 Anaconda prompt

DOI: 10.17148/IJARCCE.2023.12630

Anaconda prompt is used to run the application which is stored in the Okra_Project file. Streamlit platform is used for the project.



Okra Disease Detection Model

Browse for the input image and click on Detect Button

Fig. 16 Frontend page of Okra Disease Detection Model

After running the commands in the prompt, the frontend page of the project looks like as shown in Fig. 16.

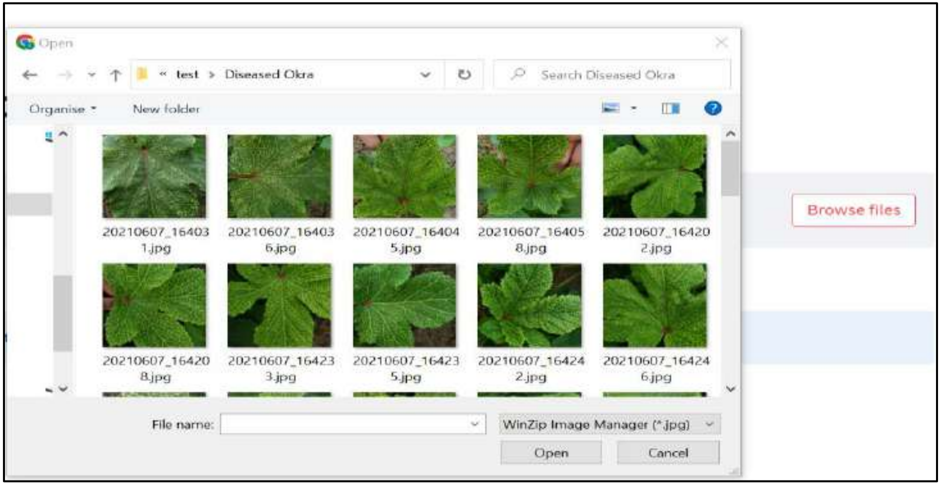


Fig. 17 Browsing diseased image file

On clicking the browse file option, the user needs to upload the image. Over here we are selecting the diseased leaf from the file system.

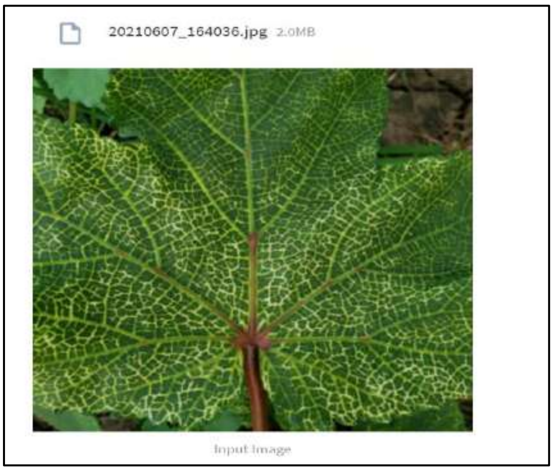
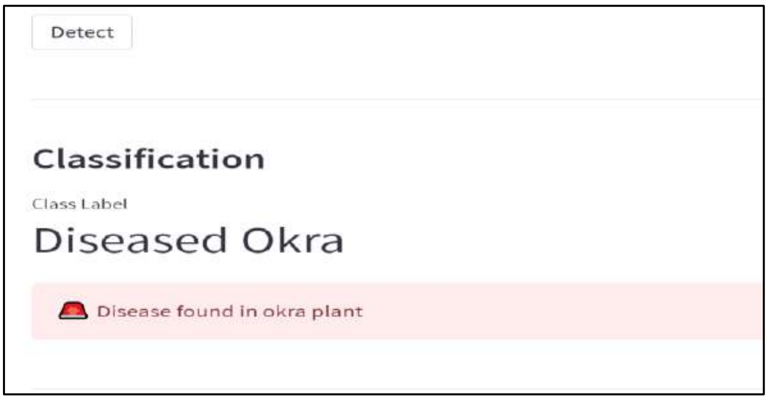


Fig. 18 Selected diseased image display

The selected image appears on the screen with the image number to the user.

DOI: 10.17148/IJARCCE.2023.12630



Classification

Class Label

Diseased Okra

Disease found in okra plant

Fig. 19 Disease Detection

On clicking the detect button the classification of the image will be done. Over here it is detected as a disease present.

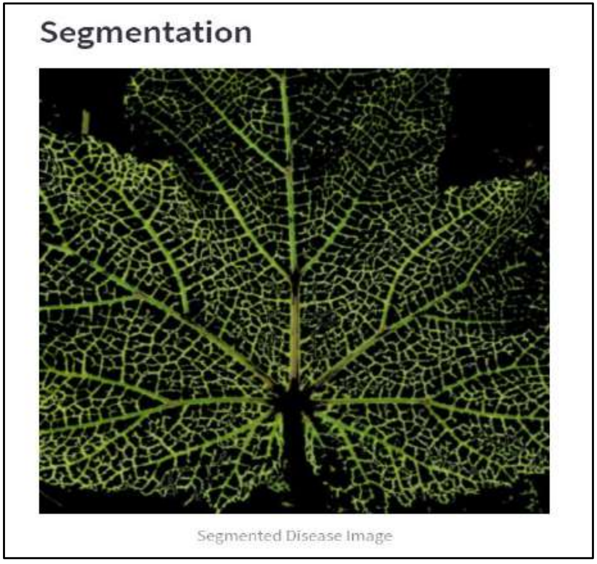


Fig. 20 Segmentation of diseased image

The segmented image of diseased image will be shown on the occasion of detection of the disease.

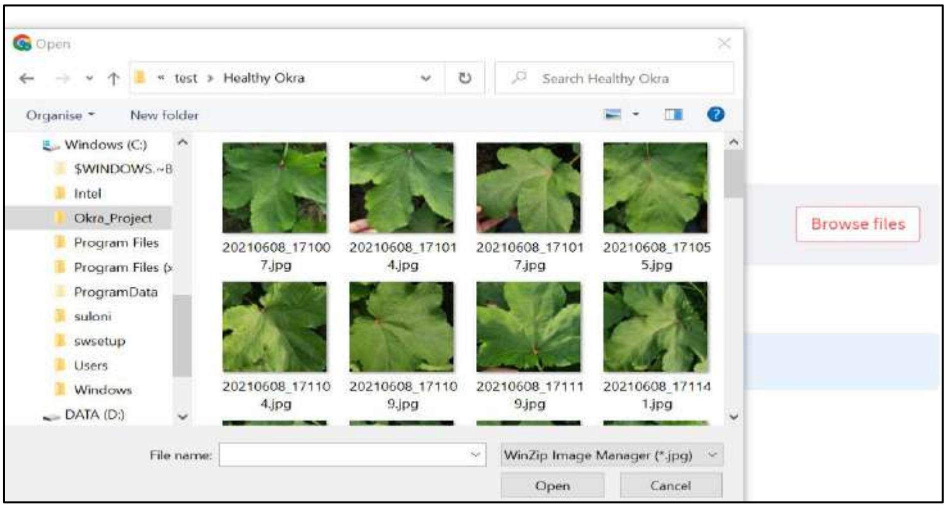


Fig. 21 Browsing a healthy image file

On clicking the browse file option, the user needs to upload the image. Over here we are selecting the healthy leaf from the file system.

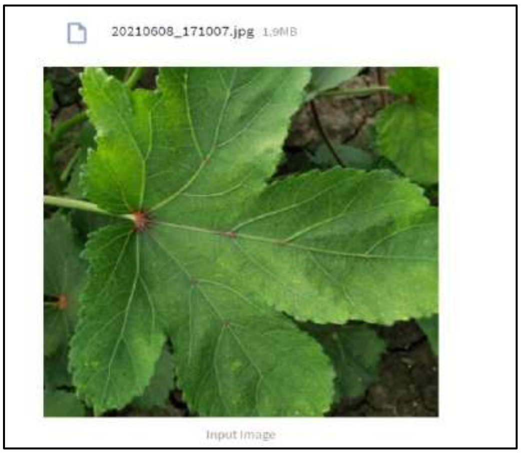
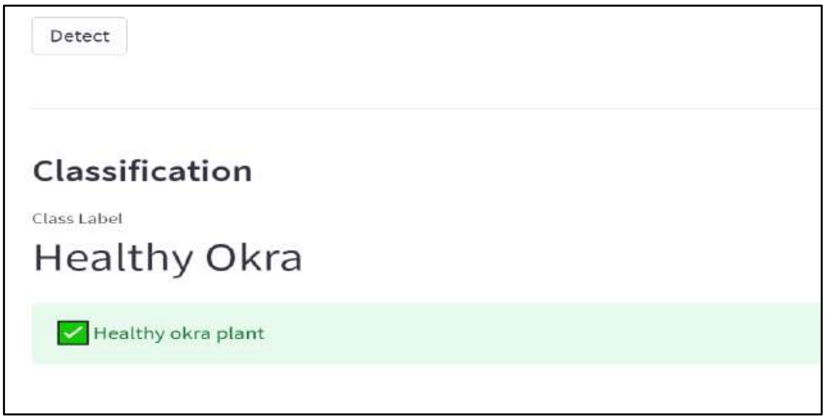


Fig. 22 Selected healthy image display

The selected image appears on the screen with the image number to the user.



Classification

Class Label

Healthy Okra

✓ Healthy okra plant

Fig. 23 No disease-detected detection

On clicking the detect button the classification of the image will be done. Over here it is detected as a healthy plant.

IV. CONCLUSION

In summary, the Okra Disease Detection Model investigates the issue of early-stage bacterial wilt infections, which can lead to early or late blight, flea beetles, insect activity, and other issues. We also contribute a dataset of early-stage bacterial wilt infections in Abelmoschus Esculentus that we self-compiled. For the creation of ground truth images for the U-net model modified as part of the semantic segmentation module, a static HSV color thresholding approach is contributed. Our analysis used the classification module which also proves the robustness of the segmentation module. Though an accuracy of 98.8% accuracy is achieved towards training datasets, there exists a scope for future work to improve the accuracy of test sets. Additionally, with early-stage datasets, performing feature engineering to reduce the intra-class variations of various infected samples with healthy samples is highly challenging.

As a next step, we propose to develop a system that is user-friendly and can work with low to high-resolution images photographed using handheld devices such as mobile phones. Through this application, one can automatically detect the type of disease the plant is infected with and also learn the ways to treat those diseases. The proposed contribution may enable to development of a handheld system that is user-friendly to use for farmers directly without any middleman support. We propose to build an Android application that will be easier and more convenient to use for the users.

REFERENCES

[1]. www.google.com

[2]. Mondal, A. Chakraborty, D. K. Kole, and D. D. Majumder, "Detection and classification technique of Yellow Vein

DOI: 10.17148/IJARCCE.2023.12630

[3]. Mosaic Virus disease in okra leaf images using leaf vein extraction and Naive Bayesian classifier," 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), 2015, pp. 166-171.
[4]. Norhalina Senan, Muhammad Aamir, Rosziati Ibrahim, N. S. A. M Taujuddin and W.H.N Wan Muda, "An Efficient Convolutional Neural Network for Paddy Leaf Disease and Pest Classification," International Journal of Advanced Computer Science and Applications(IJACSA), 2020, 11(7).
[5]. Aravind Krishnaswamy Rangarajan, Raja Purushothaman, and Anirudh Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," Procedia Computer Science, Volume 133, 2018, pp. 1040-1047.
[6]. Detection of cucumber leaf diseases using the newly generated techniques of the Haralik
[7]. features and the technique of the local tri-directional pattern by Abdul Rehman, and Zain Tariq (2021)
[8]. Nutrient Deficiencies and the detection of disease using image processing(2021) “Ankit Kumar Ghorai,S Mukhopadhyay”
[9]. Leaf Disease Detection using Digital Image Processing with SVM Classifier(2019)"Sagar Gaikwad, Sagar Shinde"
[10]. Plant Disease Detection Using Image Processing and Machine Learning "Pranesh Kulkarni, Atharva Karwande, Tejas Kolhe, Soham Kamble, Akshay Joshi, Medha Wyawahare"
[11]. Paddy Leaf Disease Detection Using Image Processing and Machine Learning(2019) "Dr. Neha Mangla, Priyanka B Raj, Soumya G Hegde, Pooja R"
[12]. Detection and Classification of Tomato Crop Disease Using Convolutional Neural Network (2022)by "Gnanavel Sakkarvarthi,Godfrey Winster Sathianesan ,Vetri Selvan Murugan ,Avulapalli Jayaram Reddy ,Prabhu Jayagopal 3 and Mahmoud Elsisi"
[13]. Literature Review of Diagnosis Rice Leaves Diseases Using Image Processing(2021) "R. I. L. Jayasooriya, Samantha Mathara Arachc