Krashi Prabhandak (Agricultural Manager)

Prafful Mundra[1], A V Pavan Krishna[2], Swarnalatha P[3], Venkata Sumanth Kakollu[4]

[1]Student of SCOPE Vellore Institute of Technology Vellore, India
prafful.mundra2018[at]vitstudent. ac.in

[2]Student of SCOPE Vellore Institute of Technology Vellore, India
krishnapavanayitha[at]gmail.com

[3]Associate Professor, SCOPE Vellore Institute of Technology Vellore, India
swarnaa2k2[at]gmail.com

[4]Student of SCOPE Vellore Institute of Technology Vellore, India
k.v.sumanth135@gmail.com

Abstract: ***Plant diseases are generally caused by pests, insects, and pathogens which may affect productivity on a large scale if not controlled in due time. Losses are incurred in the agricultural sector due to various crop diseases. It is difficult to monitor acres of fields for farmers. The proposed prototype "Krashi Prabandhak", provides the solution for effective management of the cultivated area by recognizing plant species and detecting diseases just by scanning the image of the uploaded leaf of the same. It also links the user to concerned awareness blogs regarding the species found or the disease detected. These blogs, separate from the news feed on the home page of the website, provide information worth reading about the recognized species such as favorable conditions to grow, time of the year, or environmental conditions best suitable for growth. They also provide relevant causes and remedies for the disease detected, resulting in an efficient "Agricultural Management System".***

Keywords: machine learning, species recognition, disease detection, random forest, sequential model, deep learning model

1. Introduction

The Web-App Krashi Prabandhak (Hindi for Agricultural Manager helps in care for plants in various phases of their life cycle. The user can upload an image of the leaf of a plant be it a crop or a garden plant and identify its species. Then blogs related to the species identified will be displayed. If a plant or a crop is suffering from a disease, the user can upload the image of the leaf and the disease will be identified. Further, blogs related to treatment and precautions related to the disease identified will be displayed. This is the first version of the product. As India depends primarily on agriculture, the scope of the web app is large.

After various layers of preprocessing and feature extraction for minimizing the discrepancies, images can be converted to arrays using the OpenCV library. Once converted, they can be used either with machine learning or deep learning models to obtain results. Machine learning models take less time to implement but the deep learning models are superior in accuracy. After image processing, we have used both machine learning (for species recognition) and deep learning (for disease detection).

2. Literature Survey

Machine learning has arisen with big data technologies and better processing to set out new open doors for data concentrated science in the multi-disciplinary agri-technologies area. In this paper, we present an extensive audit of examination devoted to applications of machine learning in agricultural creation frameworks. The works examined were classified in

a) Crop management, including applications on yield prediction, disease detection, weed detection, crop quality, and species acknowledgment;
b) Domesticated animals the executives, remembering applications for creature government assistance and domesticated animals production;
c) Water management;
d) Soil management.

The separating, what's more, arrangement of the introduced articles show how farming will profit from machine learning technologies. By applying machine learning to sensor data, farm management systems are evolving into real-time artificial intelligence-enabled programs that give rich suggestions and insights for farmer decision support and action.

Plant diseases are by and large brought about by pests, bugs, microbes and diminishing the usefulness to enormous scope if not controlled inside time. Agriculturists are confronting loss due to different harvest diseases. It gets drawn out to the cultivators to screen the yields routinely when the developed region is immense, that is in sections of land. The proposed framework gives the answer for routinely observing the developed region and gives the robotized disease detection utilizing far-off detecting pictures. The proposed framework insinuates the agriculturist about the yield diseases to make further moves. The goal of the proposed framework is to early detection of diseases when it begins spreading on the external layer of the leaves. The proposed framework works in two stages: the first stage manages preparing informational collections. This incorporates, preparing both healthy and diseased information (datasets). The second stage manages observing the harvest and distinguishing the disease utilizing Canny's edge detection calculation.

From the need of computerization of plant species acknowledgment and accessibility of advanced data sets of plants, we propose a picture based distinguishing proof of species of plant. These pictures may have a place with various organs of the plants like leaf, stem or bark, blossom and natural product. Various techniques for acknowledgment of the species are utilized by the piece of the plant to which the picture has a place with. For bloom class, a combination of shape, shading and surface highlights are utilized. For different classifications like stem, natural product, leaf and leaf scan, Sparsely coded SIFT highlights pooled with Spatial pyramid coordinating with approach is utilized. To cater the occasional and geological impacts on the presence of the plant, our framework additionally utilizes metadata for example content, date, time, scope, longitude related with pictures to help the distinguishing proof measure and acquire more exact outcomes. For a given picture of plant and related metadata, the framework perceives the species of the given plant picture and delivers a yield that contains the Family, Genus, and Species name. The proposed structure is executed and tried on ImageClef information with 50 unique classes of species. Most extreme exactness of 98% is accomplished in leaf filter sub-classification though least precision is accomplished in organic product subclass which is 67.3 %.

Plants are perceived however fundamental as they seem to be the essential wellspring of mankind's energy creation since they are having nutritious, restorative, and so on qualities. Whenever between crop cultivating, plant diseases can influence the leaf, bringing about gigantic yield creation harms and monetary market esteem. Hence, in the cultivating business, distinguishing proof of leaf disease assumes an urgent part. It needs, in any case, tremendous work, more noteworthy planning time, and far reaching plant microbe information. For the ID of plant disease detection different machine learning (ML) just as Deep learning (DL) strategies are created and analyzed by different analysts, and on large numbers of the occasions they additionally got critical outcomes in the two cases. Spurred by those current works, here in this article we are looking at the presentation of ML (Support Vector Machine (SVM), Random Forest (RF), Stochastic Gradient Descent (SGD)) and DL (Inception-v3, VGG-16, VGG-19) as far as citrus plant disease detection. The disease order precision (CA) we got by experimentation is very noteworthy as DL techniques perform better compared to that of ML strategies in the event of disease detection as follows: RF-76.8% > SGD-86.5% > SVM87% > VGG-19–87.4% > Inception-v3– 89% > VGG-16–89.5%. From the outcome, we can tell that RF is giving the least CA though VGG-16 is giving the awesome terms of CA.

Grape establishes quite possibly the most broadly developed organic product crops in India. Usefulness of grape diminishes because of contaminations brought about by different kinds of diseases on its natural product, stem and leaf. Leaf diseases are essentially brought about by microbes, growths, infection and so forth. Diseases are a main consideration restricting natural product creation and diseases are frequently hard to control. Without precise disease conclusions, legitimate control activities can't be utilized at the proper time. Picture Processing is one of the generally utilized procedures for the plant leaf diseases detection and characterization. This paper is proposed to help in the detection and arrangement of leaf diseases of grapes utilizing SVM order procedure. First the diseased area is discovered utilizing division by K-implies grouping, at that point both shading and surface highlights are separated. The last order method is utilized to identify the kind of leaf disease. The proposed framework can effectively recognize and arrange the inspected disease with precision of 88.89%.

Plants have a significant part in agricultural, mechanical, medication, natural and biological assurance. As of late, with a dangerous atmospheric deviation, biodiversity misfortune, quick metropolitan turn of events and ecological harm, individuals have been genuinely annihilating the regular habitats, which brings about that countless plant species continually kicking the bucket and surprisingly ceasing to exist each year. It is fundamental to ensure plant species. The initial step of ensuring plants is to remember them and comprehend what they are and where they come from. In any case, there are an enormous number of plant species that have been named on Earth, and many are at this point unclear yet, it is hard to identify every species. To deal with such tremendous data, fostering a fast and productive grouping technique has become a huge exploration. Plant species can be perceived by its leaf, blossom, skin, products of the soil, and so forth Generally talking, utilizing leaf to perceive plant species is straightforward and helpful, and many leaf based plant species acknowledgment strategies have been proposed. In this paper, we basically sum up the current leaf based plant species recognizable proof strategies, including plant leaf trademark, public information bases, highlight extraction based techniques, subspace learning based techniques, inadequate portrayal based techniques, and profound learning based techniques. The point is to underscore the significance of plant species ID, train individuals to think about plant species, and give direction and extensive investigation to the fledglings in this field, thus, to prize and secure plant species.

3. Methodology

A. Dataset description

1) Plant species dataset

The leaves were placed on a white background and then photographed. The pictures were taken in broad daylight to ensure optimum light intensity. Folio Leaf Dataset [2] from UCI is used.It contains 32 different species, with each containing 20 images.RGB images with dimensions 2322*4128 pix are used.

Beaumier du perou

- Eggplant
- Fruit Citere
- Guava
- Betel
- Rose
- Chrysanthemum
- Ficus
- Ashanti blood
- Bitter Orange

- Coeur Demoiselle
- Jackfruit
- Pimento
- Pomme Jacquot
- Star Apple
- Barbados Cherry
- Croton
- Thevetia
- Vieux Garcon
- Chocolate tree
- Coffee
- Ketembilla
- Chinese guava
- Lychee
- Sweet potato
- Hibiscus
- Duranta gold
- Geranium
- Mulberry Leaf
- Sweet Olive
- Caricature plant
- Papaya

B. Plant disease dataset

The plant disease dataset is obtained from kaggle which contains 15 different types of diseases. These diseases were mainly seen in pepper bell, potato, tomato plants

- Pepper__bell___________Bacterial_spot
- Pepper__bell___________healthy
- Potato__Early_blight
- Potato__Late_blight
- Potato__healthy
- Tomato_Bacterial_spot
- Tomato_Early_blight
- Tomato_Late_blight
- Tomato_Leaf_Mold
- Tomato_Septoria_leaf_spot
- Tomato_Spider_mites_Two_spotted_spider_mite
- Tomato_Target_Spot
- Tomato_Tomato_YellowLeaf__Curl_Virus
- Tomato_Tomato_mosaic_virus
- Tomato_healthy

Species Recognition: Using image processing, the uploaded image will be converted to numerical parameters after color, gradient, and shape feature extraction. Then the image converted as an array of values will be used as a test case for the trained model to identify it. As the image is converted to an array of values, we are using machine learning to predict the species. The model used is random forest for classification along with k-fold cross-validation to provide a higher degree of accuracy score. Once found, the recognized species will be displayed with relevant information like favorable growth conditions.

Disease detection: The image will be converted to an array using OpenCV. Then using Keras sequential model, we will identify the disease, if any in the concerned leaf image uploaded by the user. Once found, the recognized disease will be displayed with remedial measures and possible causes to help prevention in the future. The user will then be directed to the home page for registered users again.

Website home page: The website opens with selected blogs worth a read in the newsfeed, after describing the functionalities of the project, providing an option for the user to register for accessing the features provided by "Krashi Prabandhak".

Website species recognition page: This page opens with the details of species recognition and explains how to use this page to detect any type of species. Once an image is uploaded in this page the output will be the name of the species, its usage, favourable conditions for its growth and many more things about species.

Website disease detection page: This page opens with the details of disease detection and explains how to use this page to detect any type of plant disease. Once an image is uploaded in this page the output will be the name of the disease, the reasons why the disease occurs, solutions for getting rid of that disease and many more things about disease and species.

4. Implementation

4.1 Species Recognition

A. Preprocessing

i) Resizing: Resizing of input picture is fundamental to adjust the tradeoff between the speed and definite highlights on the train and test data. The pictures are resized to 512*512 pix pictures.

Figure : (512*512) Resized Leaf Image

ii) Gray Scaling: It is done to diminish the quantity of shading channels to chip away at. Thus, the hued image (RGB) is changed over to dark and white (GRAY SCALE).

Figure : Grayscaled Leaf Image

iii) Thresholding: It is utilized to change over the grayscale pictures to twofold pictures. As paired pictures are simpler to work upon when preparing includes edge and form recognition.

Figure : Leaf Image after OTSU Thresholding

Figure : Leaf Image with extracted Contours(in red)

iv) Opening and Inverse Thresholding: The yield of the initial activity is given as a contribution to the backwards Threshold work. It is utilized to make the inverse of the binary picture.

Figure : Leaf Image after Opening and Inverse Thresholding operations

v) Feature Extraction: A total of 810 features based on shape, color, intensity and gradient oriented are calculated for training the machine learning model change.

vi) Extracting Contours: Shapes can be clarified just as a bend joining every one of the ceaseless focuses (along the boundary), having a similar intensity.

Area and Perimeter: Using Moments (obtained from contours) the area of the leaf is calculated. Using arclength() from openCV, the perimeter is calculated.

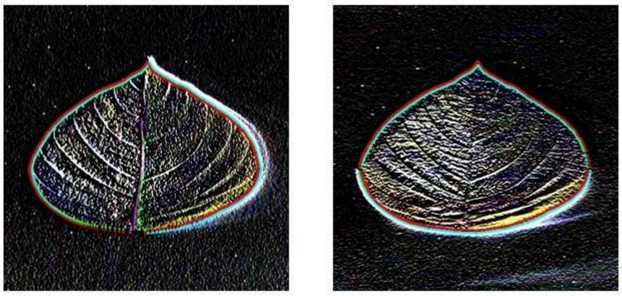
Convex Hull: It is the smallest area bounding the entire leaf.

Other Shape features: Aspect ratio, White area ratio(WR), Perimeter to area ratio(PtoA), Area of hull to Area of leaf ratio(ahtal), Perimeter of hull to perimeter of leaf(PHtP) are also calculated.

Aspect Ratio	White Area Ratio	Perimeter to Area	Perimeter to Hull	Hull Area Ratio
Width/Length	Area of leaf/(Length * Width)	Perimeter of leaf/Area of leaf	Perimeter of hull/Perimeter of leaf	Area of leaf/Area of hull

vii) Histogram of Oriented Gradient(HOG): The HOF technique counts frequencies. of gradient orientation in the localized portion of an image, using direction of gradients as features.

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803



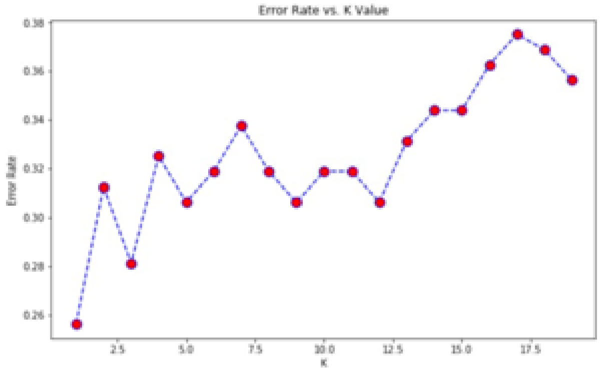
(a) Gradient of the Leaf Image along X-Axis(SobelX Operator)

(b) Gradient of the Leaf Image along Y-Axis(SobelY Operator)

Training on dataset: We trained the dataset with a testing size of 25%.

Applying various models:

i) KNN:

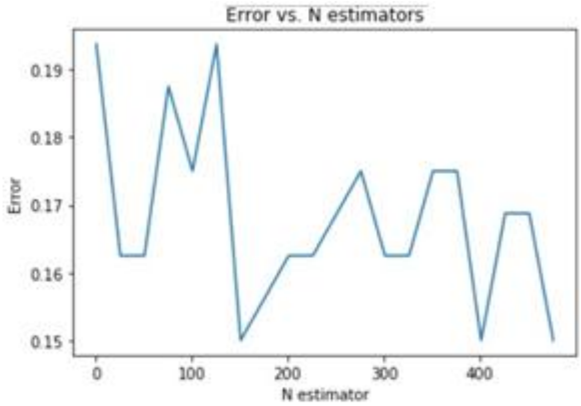


kNN is a straightforward method that works when there is practically no earlier information about the distribution of data. k in the calculation refers to the number of neighbors the calculation considers for characterization. The arrangement works in the style that a circle is drawn and k neighbors are recorded in it. Then dependent on the greater part of voting the classification is done. (Note: k should consistently be an odd number to stay away from a tie the votes). Usually, euclidean distance is utilized for kNN, but different distances like hamming distance, manhattan distance, and so forth can likewise be utilized. The benefits of this calculation are that it can learn complex models decently effectively and it is powerful in nature. On the other hand disservices of this calculation is that we need to decide the worth of 'K'.And in instances of high dimensional information low computational effectiveness and false intuition can be there. After selecting the n value as one with minimum error as 3, K - nearest neighbors was applied. An average accuracy of 70% was found.

ii) Decision Tree

It was found to give an average accuracy of 68%. A decision tree is a flowchart-like construction where each inward hub indicates a test on quality, each branch addresses a result of a test and each leaf or terminal hub is a class label. The highest hub in a tree is a root node. It utilizes a tree-like model of decisions. So the benefit this methodology holds is that it is simple to comprehend, decipher and envision. It can deal with both mathematical just as straight out data. The fundamental inconvenience this methodology holds is that little variety in information can prompt a totally unique tree, this is called Variance. Additionally over complex trees can be made that doesn't sum up the information well and can prompt overfitting. This parting can stop when a client characterized models are met. The parting can bring about completely developed trees until halting measures are met, but completely developed trees are probably going to overfit information, and consequently Pruning should be possible.

iii) Random Forest



Random forest calculation is perhaps the most famous and amazing AI calculation that is fit for performing both relapse and order assignments. As the name recommends the calculation works by making a large number of decision trees at preparing time and yielding the class that is the method of the classes(in case of grouping) or mean prediction(in the instance of relapse) of the individual trees. In general more the quantity of trees in the model more hearty is the forecast and consequently higher is the precision.

Benefit of this technique is that this strategy can deal with enormous dataset and will not overfit the information when there are more number of decision trees.Disadvantage of this strategy are that this technique is useful for order issue yet not very great for relapse issues and we have a little control on what the calculation does.

After applying a random forest classifier, with a number of estimators as 400 which was determined after calculating error with different n_estimators, an average accuracy of 83% was found. After analysing time complexity as well, RFC was chosen finally. For recognizing the species of an image, all the image processing tasks in step 1 were again done on the image and then it was predicted using the RFC.

Disease Detection

Preprocessing: The images are resized to 256*256 pix images and then converted to an array using the openCV library. Training on dataset: We trained the dataset with a testing size of 20%.

Sequential model

A Sequential model is proper for a plain heap of layers where each layer has precisely one information tensor and one yield o/p tensor. A Sequential model isn't suitable when: Your model has different information sources or various outputs. Sequence Modeling is the undertaking of foreseeing what word/letter comes straightaway. In contrast to the FNN and CNN, in sequence modeling, the current yield is reliant upon the past input and the length of the info isn't fixed.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 256, 256, 32)      896
_________________________________________________________________
activation (Activation)      (None, 256, 256, 32)      0
_________________________________________________________________
batch_normalization (BatchNo (None, 256, 256, 32)      128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 85, 85, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 85, 85, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 85, 85, 64)        18496
_________________________________________________________________
activation_1 (Activation)    (None, 85, 85, 64)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 85, 85, 64)        256
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 85, 85, 64)        36928
_________________________________________________________________
activation_2 (Activation)    (None, 85, 85, 64)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 85, 85, 64)        256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 42, 42, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 42, 42, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 42, 42, 128)       73856
_________________________________________________________________
activation_3 (Activation)    (None, 42, 42, 128)       0
_________________________________________________________________
batch_normalization_3 (Batch (None, 42, 42, 128)       512
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 42, 42, 128)       147584
_________________________________________________________________
activation_4 (Activation)    (None, 42, 42, 128)       0
_________________________________________________________________
batch_normalization_4 (Batch (None, 42, 42, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 21, 21, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 21, 21, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 56448)             0
_________________________________________________________________
dense (Dense)                (None, 1024)              57803776
_________________________________________________________________
activation_5 (Activation)    (None, 1024)              0
_________________________________________________________________
batch_normalization_5 (Batch (None, 1024)              4096
_________________________________________________________________
dropout_3 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 15)                15375
_________________________________________________________________
activation_6 (Activation)    (None, 15)                0
=================================================================
Total params: 58,102,671
Trainable params: 58,099,791
Non-trainable params: 2,880
_________________________________________________________________
```

Applying deep learning model: We then applied the sequential model with 25 epochs and verbose set to auto. The accuracy increased with each epoch, finally reaching an average of 88%.

For recognizing the disease of an image, all the image processing tasks in step 1 were again done on the image and then it was predicted using the sequential model, according to the highest probability amongst the given diseases in the dataset.

For website:

Initially we start with an app.py file where we will specify all the routing of pages and create a folder for templates and add all the required html files here. Here we specify which decorated function takes to which page of the website.

Folder structure:

```
> env
v static
  v css
  v images
      images.jfif
      my_photo.jfif
      neem.jpg
      pavan.jpg
      prafful.jpg
  > js
v templates
    disease_detection.html
    help.html
    index.html
    species_recognition.html
  app.py
  disease_detection.py
  krishi.zip
  species_recognition.py
```

We now take all the saved model files(.h5 files), .pkl files and .sav files and integrate these files with flask.

Disease detection integration flask

We use this model.h5 file for leaf disease detection. Next, is to import all the tensorflow and keras packages to use the model.h5 file. We then load the model and repeat the preprocessing steps for the test image (we take the image and resize it and now convert it to array and send it to model). We take the image from the disease detection page and save it as the testing image and send this image to disease_detection.py file where the integrated model is present. After completion of the entire processing this file returns the name of the disease to app.py. Here we search for this disease in the database and get all the information about the disease. Now the routing goes to the output page where all the information about the disease is rendered. This page includes the name of disease, causes and solutions to get rid of the disease.

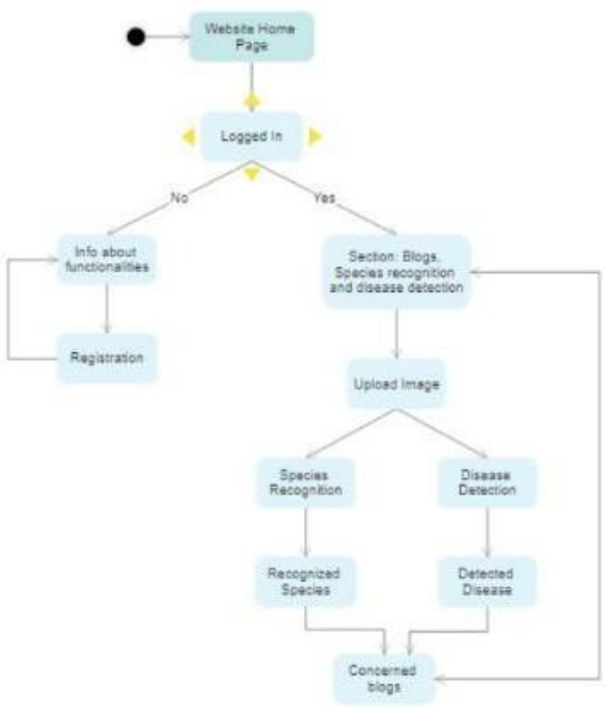
Species recognition integration flask:

For implementing the species recognition part we use cnn_model.pkl, Forest_Regression_Classifier.sav and label_transform.pkl files. Initially we load all the pkl files and .sav files in species_recognition.py file. Now we start preprocessing of the image in the following order Resizing the image to 512x512px, GrayScaling the image, giving the a threshold value and applying threshold to image, now applying Opening and Inverse Thresholding to the image and extraction Several features such as shape, color, intensity and gradient of the image and finally after applying hog transform the preprocessed image is sent to the pkl files to give the output number which we will match with the items in the array and finally return the leaf name to the app.py file.

Now in app.py file we get all the data about the species name from the database and this data is rendered in a clean way in the output page.

Some other small parts of implementations are making help pages for users so that they can have a clear understanding of how the website works.

5. Flow Chart

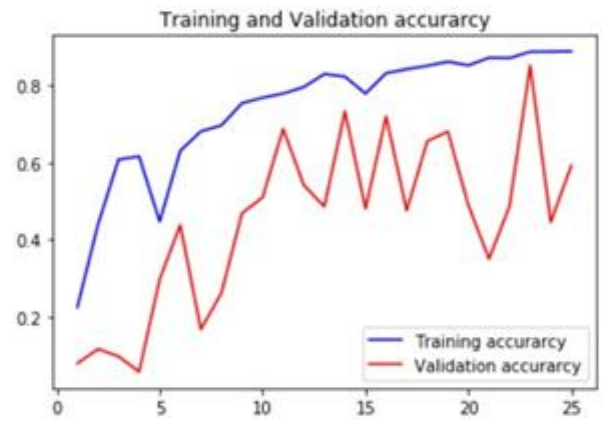


6. Results and Discussion

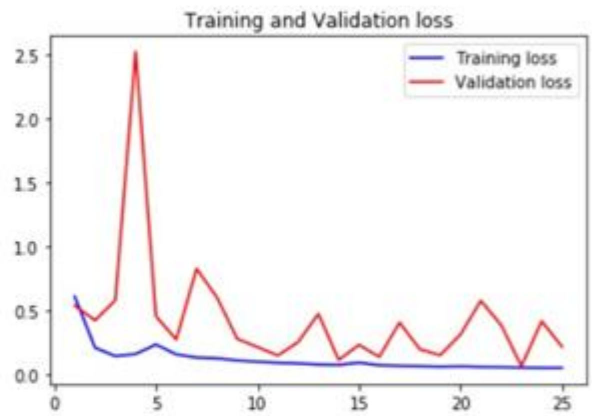
Classification report of species recognition using RFC: The precision, recall and f1 score can be seen from the following classification report of the 32 species in the dataset. An average accuracy of 83% was obtained.

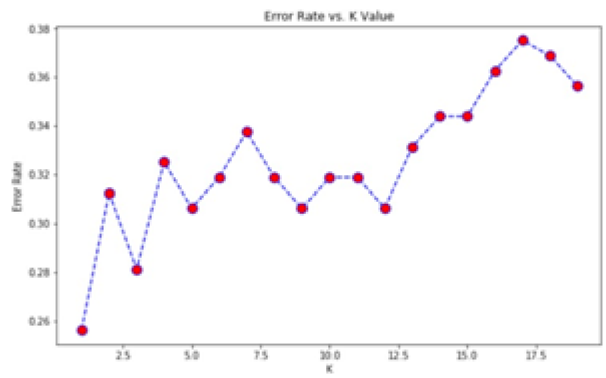
	precision	recall	f1-score	support
0	0.75	0.86	0.80	7
1	1.00	0.50	0.67	4
2	0.60	1.00	0.75	3
3	1.00	0.83	0.91	6
4	0.57	0.50	0.53	8
5	1.00	1.00	1.00	2
6	1.00	1.00	1.00	2
7	1.00	1.00	1.00	8
8	1.00	1.00	1.00	4
9	0.75	1.00	0.86	3
10	1.00	0.67	0.80	9
11	1.00	1.00	1.00	6
12	0.80	1.00	0.89	4
13	1.00	0.33	0.50	3
14	1.00	1.00	1.00	4
15	1.00	0.83	0.91	6
16	0.75	1.00	0.86	3
17	0.71	1.00	0.83	5
18	0.50	0.67	0.57	6
19	0.89	0.89	0.89	9
20	0.60	1.00	0.75	3
21	0.75	1.00	0.86	3
22	1.00	0.57	0.73	7
23	1.00	1.00	1.00	8
24	1.00	1.00	1.00	4
25	1.00	0.75	0.86	4
26	1.00	0.25	0.40	4
27	0.40	0.67	0.50	3
28	0.60	0.75	0.67	4
29	1.00	0.86	0.92	7
30	0.88	1.00	0.93	7
31	0.80	1.00	0.89	4
accuracy			0.83	160
macro avg	0.85	0.84	0.82	160
weighted avg	0.87	0.83	0.83	160

Training and validation accuracy for disease detection: It can be seen how the accuracy, for both training and validation, increases as the number of epochs is increased.



Training and validation loss for disease detection: It can be seen how the loss, for training and validation, decreases as the number of epochs is increased and is minimum at the 23rd epoch.





Statistics during training of the max validation accuracy, i.e. 85.11% (23rd epoch, can be determined from the above figure):

```
Epoch 23/25
73/73 [==============================] - 309s 4s/step - loss: 0.0506 - accuracy: 0.9059 - val_loss: 0.0659 - val_accuracy:
```

7. Conclusion and Future Work

The 32 species available in the dataset were recognized, with an average accuracy of 83% by the random forest classifier model of machine learning. The diseases were detected too, with 16 types of diseases in total, with the sequential model from keras, with a validation accuracy of 85%. The response time for preprocessing and predicting a test image was of the order 0.02 seconds.

However even with 4GB GPU, SSD, 16GB RAM and i5 processor specifications, the time taken to preprocess all the species in species recognition was significant. For disease detection too, each epoch took around 300 seconds to process. In future, while integrating with a larger dataset, distributed and cluster computing should be used to reduce the processing time. The website will also have a subscription based model for revenue generation. It will include the basic functionalities of the project as intended for normal registered users, but specialist consultation and onsite inspection for premium users. The blogs will also be catered specifically for different scenarios.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

References

[1] C.H. Arun, W.R. Sam Emmanuel, D. D. Folio dataset. UCI Machine Learning Repository.
[2] Machine Learning in Agriculture: A Review Konstantinos G. Liakos 1 , Patrizia Busato 2 , Dimitrios Moshou 1,3, Simon Pearson 4 ID and Dionysis Bochtis
[3] Crop Disease Detection using Machine Learning: Indian Agriculture Anuradha Badage
[4] Automatic Plant Species Recognition Technique using Machine learning approaches Suchit Purohit, Ronak Viroja, Savita Gandhi Naina Chaudhary
[5] Jyotismita Chaki, R. P. Plant leaf recognition using shape based features and neural network classifiers.
[6] Opencv python tutorials.
[7] V. Satti, A. Satya, S. S. International journal of engineering science and technology(ijest). 874–879. An automatic leaf recognition system for plant identification using machine vision technology
[8] .Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.
[9] Badage, A. (2018). Crop disease detection using machine learning: Indian agriculture. Int. Res. J. Eng. Technol, 5(09).
[10] Purohit, S., Viroja, R., Gandhi, S., & Chaudhary, N. (2015, December). Automatic plant species recognition technique using machine learning approaches. In 2015 International Conference on Computing and Network Communications (CoCoNet) (pp. 710-719). IEEE.
[11] Sujatha, R., Chatterjee, J. M., Jhanjhi, N. Z., & Brohi, S.
[12] N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection. Microprocessors and Microsystems, 80, 103615.
[13] Padol, P. B., & Yadav, A. A. (2016, June). SVM classifier based grape leaf disease detection. In 2016 Conference on advances in signal processing (CASP) (pp. 175-179). IEEE.
[14] Zhang, S., Huang, W., Huang, Y. A., & Zhang, C. (2020). Plant species recognition methods using leaf image: Overview. Neurocomputing, 408, 246-272.