Agriculture Field Monitoring and Plant Leaf Disease Detection

1st Rohit Nalawade
Dept. of Information Technology
Xavier's Institute of Engineering, Mumbai
Maharashtra, India
apoorv.nagap99@gmail.com

2nd Apoorv Nagap
Dept. of Information Technology
Xavier's Institute of Engineering, Mumbai
Maharashtra, India
apoorv.nagap@gmail.com

3rd Lakhan Jindam
Dept. of Information Technology
Xavier's Institute of Engineering, Mumbai
Maharashtra, India
lakhanj569@gmail.com

4th Prof. Meena Ugale
Dept. of Information Technology
Xavier's Institute of Engineering, Mumbai
Maharashtra, India
meena.u@xavierengg.com

***Abstract*—Terrace gardens are becoming a common feature in today's urban environment. It is difficult to keep a track on the plant's growth and health. Various pathogens are present in the environment which severely affects the plants and the soil in which it is planted, thereby affecting the production. Current systems cannot help people about the disease with which the plants leaves are affected by and what steps should be taken in order to prevent it from being damaged. The proposed system provides the leaf disease detection along with complete surveillance of the field with real-time values of field factors like temperature, humidity, moisture, etc. i.e. real-time monitoring. User can automatically control the flow of water if not physically present via app, also the real-time values can be tracked.**

***Index Terms*—leaf disease detection, IoT, API, field monitoring, dataset, neural network, sensor.**

I. Introduction

India is the largest freshwater user in the world, and the country's total water use is greater than any other continent [18]. The agricultural sector is the biggest user of water, followed by the domestic sector and the industrial sector. A small scale practice of doing the farming is Gardening, which is done at the backyards of houses or in balconies. Gardening is the practice of growing and cultivating plants as part of horticulture [16]. In such congested environment, rooftops and terraces of buildings remains as a valuable sources for urban horticulture. Crop diseases are a major threat to food security, but their rapid identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. In field of agriculture, detection of disease in plants plays an important role. The existing method for plant disease detection is simply naked eye observation by experts through which identification and detection of plant diseases is done [17]. To detect a plant disease in very initial stage, use of automatic disease detection technique is beneficial.

II. KEY FEATURES

- Recognize the disease of the plant by feeding the images to the machine learning model.
- Machine learning model is flexible and can be trained for any type of diseases and different kinds of plant.
- Real-time images are provided to the app which in turn is integrated with Machine learning model and output disease is predicted.
- Information about each plant is available on the app such as description, fertilizers, pesticides, etc.
- Real-time values from the various sensor through Arduino is retrieved from the real-time database of firebase.
- Model which is built for detecting disease can be integrated with any app because the model is converted to API and requests can be made to the endpoint URL of the API.
- It is hosted globally so anyone can access the API and integrate it with the app.

III. DESIGN

A. Details of hardware and software

a) Software Requirements:

- Python (3.5 or above)
- Text editor(Atom, sublime text, etc)
- Firebase
- Jupyter Notebook
- Flutter framework and dart language
- Ngrok localhost webhook development tool
- Arduino IDE

b) Hardware Requirements: The Table 1 contains the different sensors like DH for sensing temperature and humidity, LDR for sensing light, soil moisture sensor for reading moisture value from soil, NodeMCU for setting up WIFI and so on.

978-1-7281-2339-4/20/$31.00 ©2020 IEEE

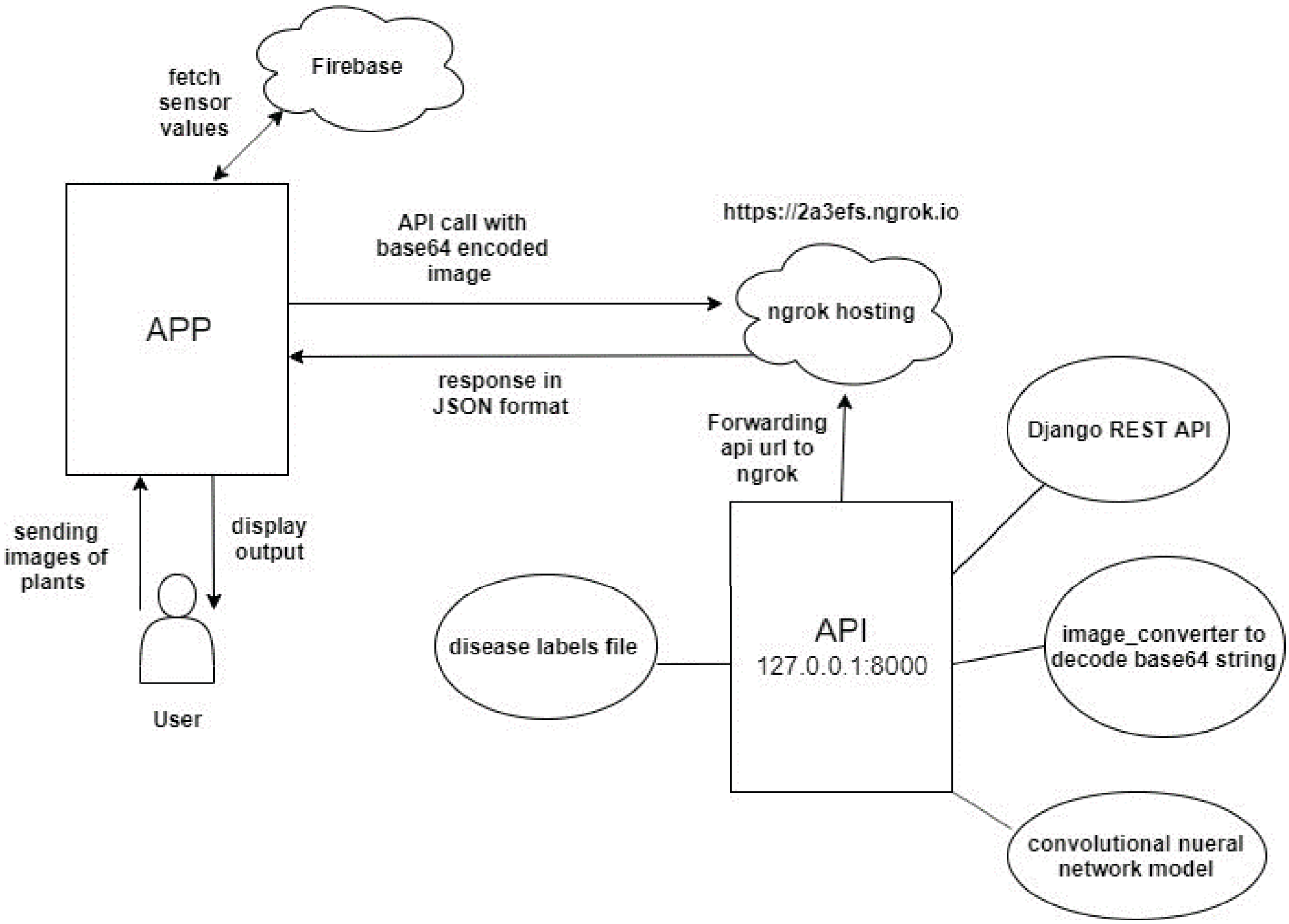

Fig. 1. Design of the System

TABLE I
HARDWARE REQUIREMENTS

Items	Model
NodeMCU	Wi-Fi module ESP-12
DHT sensor	DHT11
Soil moisture sensor	FC-28
5V External power supply	3.3V/5V MB102 PowerSupply Module
Jumper wires	N.A
Breadboard	N.A
Water motor	N.A
Relay module	5v JQC-3FF-S-Z
LDR	Ktc Cons

c) System Requirements:

- Processors: Intel R CoreTM i5 processor 8300H at 2.60 GHz or 2.80GHz(1 socket, 4 cores, 2 threads per core), 8 GB of DDR4 RAM
- Graphics Card : Intel HD graphics 630 or Nvidia Geforce card
- Disk space: 4 to 8 GB
- Operating systems: Windows 10, macOS, and Linux.

B. Block diagram

The block diagram of the system is shown in Fig. 1. Here, the connection of App, firebase and API is represented. The API URL is forwarded to ngrok. From there the response is given in the JSON format to the app. The API is attached to the disease label files, image convertor, CNN model. User directly communicates with the app.

IV. IMPLEMENTATION

In order to build and deploy the model, implementation is divided into two parts. First part is creating the REST API of the CNN (convolutional neural network) model and second part is integrating it to the application.

A. REST API of the model

a) Dataset of the plant leaves:

i The images of diseased plant are acquired from the Kaggle platform for testing and training the model.

ii Dataset has approximately 9000+ images for 10 plants leaves for 13 categories.

iii First 200 images of each category are used for training of the model and subsequently converted each image to an array with pre- defined default dimensions (256x256) using Cv2 module

iv To detect and identify the disease, the labels are created for each disease associated with the plant.

b) Defining and creating a CNN model:

i Create a sequential model with input shape pre- defined.

ii In the model, there is a default to “channel_last” architecture but also creating a switch for backends that support “channel_first”.

iii Takes place creation of the first CONV⇒RELU⇒POOL. CONV layer has 32 filters with a 3 x 3 kernel and RELU activation (Rectified Linear Unit).

iv Batch normalization, max pooling, and 25% (0.25) dropout is applied.

v Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data.

vi Next is the creation of two sets of (CONV⇒RELU) * 2⇒POOL blocks. Then only one set of FC (Fully Connected Layer⇒RELUlayer.

vii Here Keras Adam Optimizer is used for the model and compiled it using binary_crossentropy with metrics asaccuracy.

viii Training the network is initiated where model.fit_generator is called, supplying the data augmentation object, training/testing data, and used an epochs value of 30 for this project.

ix Then the model file is saved using pickle module of python in .pkl format.

c) Django REST framework for building API:

i First a Django app is created using command “django-admin.py startproject disease_api”, where disease_api is name of the project.

ii Command to create app within project folder “python manage.py startapp appname”.

iii Set the output URL of the API in urls.py file of the django app and define the path as/predict.

iv Creation of a function called image_converter in image.py file which decode the base64 encoded image into original image using PIL module.

v API returns the output in JSON format and accepts the request in base64 encoded image.

B. Flutter to build the application and integrating with custom REST API

a) Design of the app:

i It includes three sections namely Plants info, prediction of disease, Sensor data. The app has a camera which captures the image of leaf, also user can add the image from the storage of the mobile. Fig.2 and Fig. 3 shows the app design.

Fig. 2. Information page for the Apple Plant

ii Plant’s info includes the description, conditions and requirements, fertilizers and pesticides, and Disease associated with each plant, as shown in Fig. 2.

iii Prediction of disease consist of capturing or selecting an image of a diseased plant and making a API call which then predicts and categorize the disease.

iv Sensor data involves monitoring of the values fetched from sensors placed on the field with help of firebase and it also helps to control the water pump on the field.

b) Integrating API endpoint to app:

i Using ngrok for providing custom and permanent domain to the API to handle requests.

ii It provides the API endpoint on which request can be sent and endpoint is used in flutter to send a API request of base64 encoded image.

iii Response from API is in JSON format, so first decoding the JSON into plain text format using flutter functions.

iv Result is displayed along with the solutions to cure or prevent the disease.

C. Field monitoring

The hardware required for field monitoring is shown in Fig. 4. Here, the data from the sensors are sent to the app. This

Fig. 3. Field values obtained on app

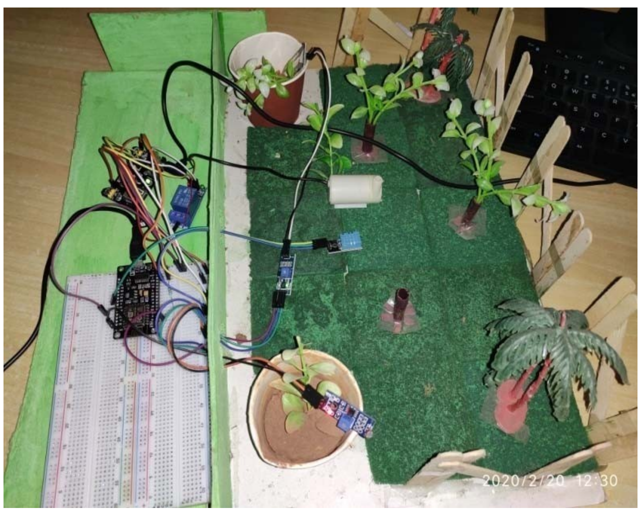

Fig. 4. Hardware Connection for Field Monitoring

data includes light intensity, soil moisture value, temperature and humidity. Also the water motor which is present on the field, can be controlled via app where the status of on or off is displayed. Hence, the overall field report is reflected on the app. The values of the sensors go to the firebase and from there these values are fetched and displayed on app

V. RESULTS & DISCUSSIONS

Dataset has approximately 9000+ images for 10 plants leaves for 13 categories [20]. The 200 images for individual disease are used for training purpose and the remaining images approximately 700+ are used for testing purpose. The Fig. 7 shows some sample images of leaf diseases. The Table. II shows the leaf diseases, the samples taken, the result obtained

```
[INFO] Calculating model accuracy
440/440 [==============================] - 23s 53ms/step
Test Accuracy: 98.07851271195845
```

Fig. 5. Model Accuracy

and the accuracy obtained for each disease for different 10 categories of plants. After the processing of the CNN model, with the help of REST API, the output is directly shown on the app. In Fig. 6, Training accuracy graph calculates accuracy of model at each epoch or iteration along with the accuracy while validating the data. Accuracy loses while training and validating and second graph depicts the loss at each epoch. In Fig. 5, for predicting the model accuracy, evaluate function of model calculates it on the basis of "x_test" and "y_test" parameters which is data subset used for testing.

Accuracy for grapes leaves is less because the disease patterns of grape leaves along with the shape interfere with the strawberry leaves and is facing difficulties in prediction. The accuracy of pepper bell plant is 100% due to the fact that the pepper bell has clear patches with shape of the leaves being different from other leaves which makes it easier to recognize. The average accuracy (testing with real time diseases) obtained for leaf disease prediction is 87.43 %.

TABLE II
PLANT'S LEAF DISEASES AND THEIR RESPECTIVE ACCURACIES OBTAINED

Sr.No	Plant name	Disease name	Total sample	Accuracy
1	Apple	Apple black rot	10	80
2	Grapes	Grape leaf blight (Isariopsis leaf spot)	15	86.67
		Grape Black Measles	10	80
3	Corn	Corn (maize) common rust	15	93.33
4	Peach	Peach bacterial spot	10	90
5	Potato	Potato early blight	10	80
		Potato late blight	10	90
6	Tomato	Tomato mosaic virus	10	90
		Tomato yellow Leaf curl virus	15	80
7	Strawberry	Strawberry Leaf scorch	20	90
8	Pepper	Pepper bell Bacterial spot	10	100
9	Cherry	Cherry (including sour) healthy	15	86.67
10	Squash	Squash powdery mildew	10	90
Accuracy				87.43%

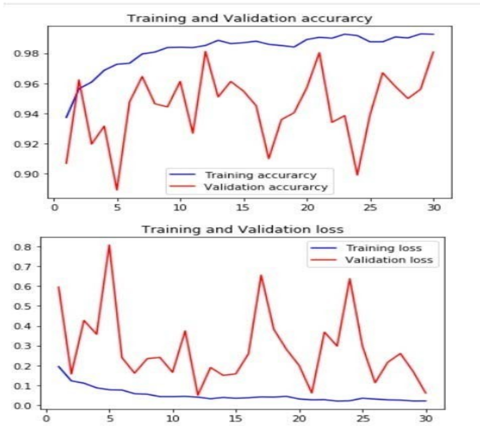

Fig. 6. Training and Validation graph

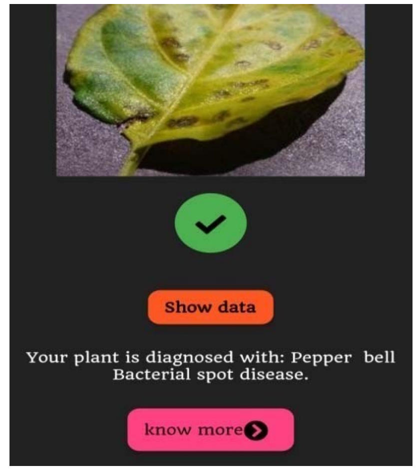

Fig. 8. Prediction of disease from the app

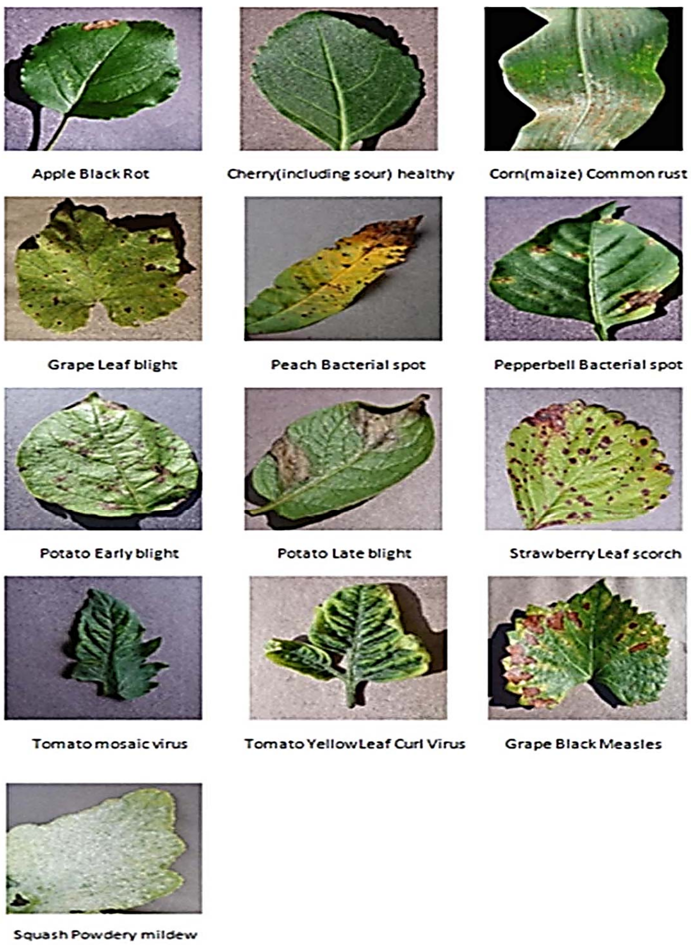

Fig. 7. Leaf Disease Samples

VI. CONCLUSION AND FUTURE SCOPE

The proposed system periodically monitors the cultivated field and successfully shows the status on the application developed. The water pump can be controlled through the application. Application shows the information about the plant, fertilizers required, soil factors and pesticides to treat diseases. The average accuracy of the model is 98.07%. The user can observe both the outputs i.e. the field report and the disease detection on the app.

Leaf diseases are detected in early stage by using edge detection and Machine learning. Machine learning techniques are used to train the model which helps to take a proper decision regarding the diseases. The pesticide as a remedy is suggested to the user for infected diseases to control it. The proposed system can be further extended by adding extra functionalities like location of stores present nearby user, list of pesticides and fertilizers, real-time interaction with agricultural experts via chatting or video call, etc.

REFERENCES

[1] https://www.hackster.io/evive-an-opensource-embedded-platoform/plant-monitoring-and-watering-system-using-evive- d48677, 01st August 20:02
[2] https://www.youtube.com/watch?v=nUHizmtyt74, 20th September 20:02
[3] http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity- sensor-on-an-arduino, 5th August 22:02
[4] https://www.hackster.io/evive-an-opensource-embedded-platoform/plant-monitoring-and-watering-system-using-evive-d48677, 20th September 20:02
[5] http://web.csulb.edu/ bpenzens/se4s/smart.html, 11th August 18:02
[6] https://www.packtpub.com/mapt/book/hardware_and_creative/9781787286429/1/ch01lvl 1sec10/sensor-devices-for-a-smart-gardening-system , 19th October 17:02
[7] https://www.cnet.com/news/smart-garden-buying-guide/,18th November 23:02
[8] https://github.com/salathegroup/plantvillage_deep learning_paper_dataset, 20th March 20:45
[9] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems, eds F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc.), 1097–1105 , 15th April 19:51
[10] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. arXiv:1512.03385, 19th February 1:02
[11] Zeiler, M. D., and Fergus, R.(2014). “Visualizing and understanding convolutional networks,” in Computer Vision–ECCV 2014, eds D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (Springer), 818–833, 4th September 2:02

[12] Anand.H.Kulkarni, Ashwin Patil R. K. Applying image processing technique to detect plant diseases. International Journal of Modern Engineering Research. Vol.2, Issue.5, Sep-Oct. 2012 pp-3661-3664, 13th February 20:22
[13] B. Bhanu, J. Peng. Adaptive integrated image segmentation and object recognition. In IEEE Transactions on Systems, Man and Cybernetics, Part C. volume 30, pages 427–441, November 2000, 30th September 20:33
[14] S. Arivazhagan, R. Newlin Shebiah, S. Ananthi, S. Vishnu Varthini. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture, 23rd January 1:02
[15] Agric Eng Int: CIGR Journal, 15(1): 211217 2013, 16th August 20:02
[16] https://www.wisegeek.com/whatisaterracegarden.htm,19th April 12:30
[17] https://www.sciencedirect.com/science/article/pii/S221431731630015 4, 25th September 15:45
[18] https://ourworldindata.org/waterusestress, 8th October 18:15
[19] https://robu.in/product/mb102breadboardpowersupplymodule3 3v5varduinosolderlessbreadboard/, 19th August 14:02
[20] https://github.com/spMohanty/PlantVillageDataset, 22nd March 20:50