DEEP LEARNING-BASED PLANT DISEASE DETECTION

Sweta Gupta
Computer Science and Engineering
REVA University
Bangalore, India
swetaguptarph99@gmail.com

V Chandana
Computer Science and Engineering
REVA University
Bangalore, India
chandanachandu3514@gmail.com

Chaitra A C
Computer Science and Engineering
REVA University
Bangalore, India
chaitraacgowda2710@gmail.com

Ambati Vamsidhar Reddy
Computer Science and Engineering
REVA University
Bangalore, India
ambativamsi200@gmail.com

Meenakshi Sundaram A
Computer Science and Engineering
REVA University
Bangalore, India
meenakshi.sa@reva.edu.in

Sneha N
Computer science and Application
REVA University
Bangalore, India
Sneha2442@ gmail.com

Abstract— The proportion of agricultural production plays a central role in the country's economic development. However, crop diseases are the most serious obstacle to food production and quality. Early detection of crop diseases is critical to global health and well-being. In a traditional diagnostic procedure, a pathologist visits the site to visually assess each plant. However, manual training on plant diseases is limited due to inaccuracies and low staff. To address these issues, we need to develop an automated approach that can efficiently detect and classify diseases in many plants. Advanced technology has made it possible to produce enough food to satisfy society's desires. However, food and crop protection are still invincible. Factors including climate change, reduced pollen counts, and crop diseases pose difficult situations for farmers. An important background of these factors must be achieved as a matter of preference.

Keywords—CNN, deep learning, disease detection

I. INTRODUCTION

As farmers cultivate a wide variety of crops, plants protection is closely related to sustainable farming and climate agriculture plays important role in the economic sector, Indian economy sector accounts for 18% of GDP and it also provides employment for 50% of the people of India but, due to the plant diseases the harvest will be significantly less and leads to drought. The traditional detection is a visual assessment by a pathologist but, it's not easy to detect all kinds of disease by site at the same time if the computation is for a large form it takes a huge time in computing. In this project a deep learning-based model that has the data collection of healthy and disease symptom leaf with the proper and best split of that data, through pre-processing and finally, the confidence calculation component is responsible for providing the confidence level during the resulting.

II. LITERATURE SURVEY

In this project, we developed the model where it detects the virus, fungal and bacterial diseases so that diseases can be easily detected by

SYSTEM	**AlGORITHMS USED**	**FEATURE EXTRACTION TECHNIQUE**	**ACCURACY**	**DRAWBACKS**
Yellow Vein Mosaic Virus Disease Detection System	Naïve Bayesian K-means	Dominated-Featureset-Selection	87%	The system only Supports Okra plant diseases.
Plant Disease Identification Mobile Application	CNN Clustering	Not mentioned in the paper	97.6%	The user should supply plant details manually to start the process.
Crop disease recognition System Using Machine Learning	Random forests	HOG feature extraction	70%	To get accuracy An immersive number of data should be fed.
Crop disease detection of Jute plant.	Support Vector Machine	Grey_x0002_level Cooccurrence Matrices	80%-86%	The system only supports Jute plant YVMV disease.

some marks and the color and predicted by processing. The disease can be further used to maintain the crop field. This model will help in detecting plant disease without harming the plant leaf.

Table 1. Various research on Plant leaf disease

III. REQUIREMENTS

- Hardware Components

a) PC

PC (personal computer) is for storing data and processing it in binary form, based on the instructions given to it. the requirement of the hardware is an i3 6th generation 6006U processor RAM of 6GB with a disk of 1TB.

b) Graphics Processing Unit (GPU)

GPU is an electric circuit that is commonly found in embedded systems, cell phones, and workstations as well. The graphics processing unit is used for fast mathematical calculation, personal computing, and accelerating machine learning. Our project is responsible for rendering images that are both 2d and 3d which helps in computing the images faster so the epoch will run faster than usual, modern graphics processing units are more efficient in processing images.

- Software components

a) Keras

Keras is a python library, used in neural networks. Keras is used to create graphical representations of models, which aids in understanding the model's structure. Auto Keras, a Keras-based library, has also gained popularity and can be used to improve the speed with which results are obtained.

b) TensorFlow

In TensorFlow most of the APIs come built-in such as Keras API that helps in loading dataset, splitting it into required ratios of 6:3:1 where 60% is for training 30% for testing, and 10% for evaluation, loading labels, and the image processing. TensorFlow is a open-source deep learning framework by the Google team.

c) Open CV

Open CV is an open-source python library which is used in computer vision projects. It can perform operations such as image processing and video capturing.

d) NumPy

Numpy is used when working with multi-dimensional arrays. It's the most important Python package for scientific computation. It also includes functions for manipulating linear algebra, Fourier transforms, and matrices. Travis Oliphant created NumPy in 2005. This is a free open-source project. NumPy is an abbreviation for Numerical Python.

e) Google Colab

Google colab is a jupyter notebook where it can easily integrate with drive and run on the cloud, where data can be easily accessible. The notebooks can be shared and edited concurrently by team members, even if they are in different locations. The notebooks can also be shared with the public by publishing them on Git Hub. Google spent many years developing TensorFlow, an AI framework, and Collaboratory, a development tool. TensorFlow is now open-source, and Google has made Collaboratory available to the public for free since 2017. Google Colab, or simply Colab, has replaced Collaboratory.

f) CNN

CNN is a complex neural network chain that works to extract image features from a trained data set and classify them to produce the desired output. It trains the neural networks by converting the data set images to numerical values. Without any intervention of human, CNN can detect features on its own. ConvNets outperform machine learning algorithms in terms of power and efficiency. These numerical values are then organized into numerical arrays based on their categorized properties. These arrays are then distributed to various nodes in the network and iterated multiple times based on the input provided.

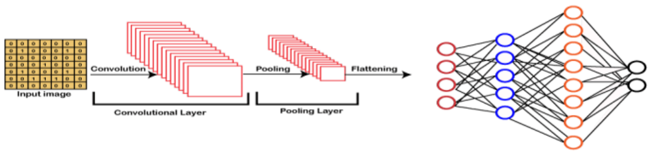


Fig. CNN

CNN models are used for geographical classification in a variety of companies that require data to be classified quickly and securely. It acts almost like a filter, removing dust and separating image features. Below are the layers in CNN that are depicted in the above image.

i. Convolutional Layer

In this layer, feature map is extracted where in the filter of specific size is used to perform convolution on input images with specific stride. The dot product is used in the operation of filtering. Then, the result of this layer i.e., feature map is used in next layers.

ii. Pooling layer

This layer is used to reduce the size of feature map by extracting the maximum value from the stride of specific size. So that, the computations to the computer will be greatly reduced and can be both time saving as well as cost saving. In this model we used max pooling.

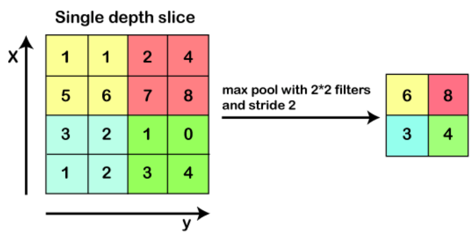


Fig. Max Pooling

iii. Activation Function

The activation of neural networks is a crucial aspect of deep learning. It's precision and consistency, as well as the computational efficiency of creating and interrupting massive neural networks. To put it another way, the activation function can be thought of as a mathematical condition that defines the neural network's output. The output of each neuron is usually justified by operational functions with an execution between 1 and 1 or between 1 and 1.

The activation function adds non-linearity to the model, allowing you to learn complex input-to-response function mappings. In this model we have used ReLU Activation function.

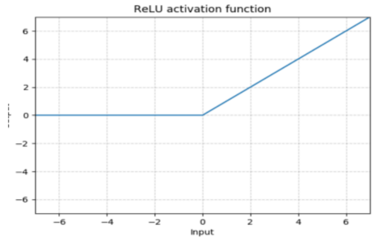


Fig. ReLU

iv. Fully-Connected Layer

This layer classifies based features extracted in previous layers and their filters. In convolutional and pooling layers tend to use Relu features, in fully connected layer basically leverages a SoftMax activation to calculate inputs, and gives a chance from zero to one at least. Usually, the suppression layer is used among all consecutive completely related layers to result in non-linearity and reduce overload, respectively.

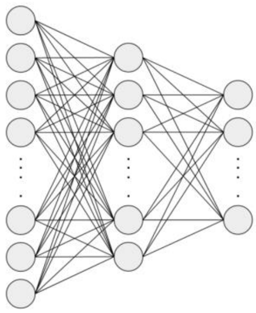


Fig. Fully Connected Layer

v. Dropout layer

Dropout layers are used in convolutional neural network models to avoid overfitting demonstrations. It acts as a regulator in the neural assembly.

IV.FLOW OF SYSTEM

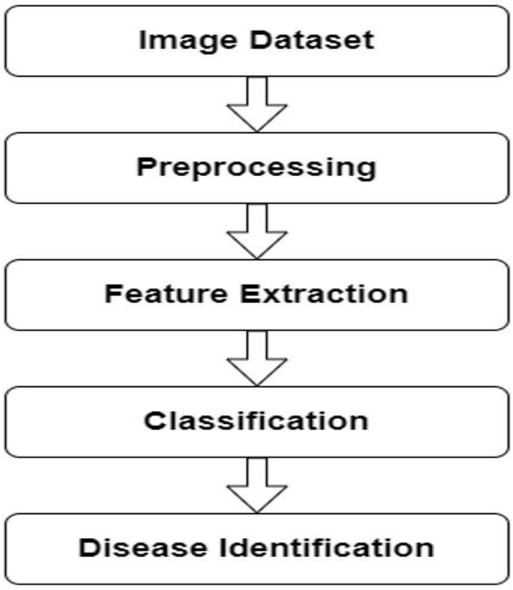


Fig. Model Flow

V. IMPLEMENTATION METHODOLOGY

The system is made up of a deep learning model that has been trained to recognize a specific plant disease. Deep learning is used in the system to detect crop diseases automatically. It is proposed to use Convolutional Neural Network as the deep learning algorithm.

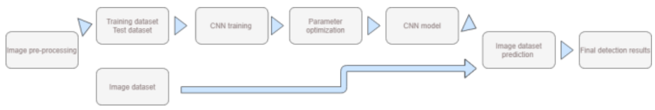


Fig. Methodology

Before sending the image to the network, the RGB image of the leaf, width, and height of 254 and 254 respectively, is scaled to 224 * 224 in the image preprocessing part and then this will an input layer of the CNN. Then this image is dispatched via a CNN which encodes this image into the array of numbers and classifies it with the opposite arrays in the version. The model is a tensor Flow version that is made into a tensor flow lite model due to the huge size of the normal tensor flow version. This model facilitates classifying the uploaded image numerical value to the data set values. when a numerical array fits it calculates the probability and shows the value which has the high probability which will be shown in the output.

a) DATASET:

With the intention to get a minimal loss and higher accuracy. we have used a big dataset for creating the model of CNN. The dataset of 10,000 leaf pictures of both healthy and diseased are used to train the CNN model that belong to 10 classes. The CNN model is tested against various kinds of data partitioning and the best partition of 6:3:1 is selected and the dataset is divided accordingly into three partitions. The dataset includes four forms of plant species and six sorts of plant diseases collectively 10 classes. The dataset incorporates both healthy and diseased crop images. The images cover species of crops, like Betel, cinnamon, coffee, and manioc. each of the pictures is re-sized and segmented for pre-processing and further classification.

b) PRE-PROCESSING OF THE IMAGES:

The picture of the plants are merged by pre-processing for best results. Pre-processing an image follows the steps of resizing the image, increasing the edges, and filtering the image. To use 8-way proximity measurement to isolate diseased plant areas and further examine the segmented sections, it is important to convert them to RGB format and restore the original colors. The image is processed for plant disease with the plant disease detection model. Using a pre-trained model. plant leaves are then analysed for the disease symptoms of the plant with the help of a dataset and CNN.

```
validation_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1/255)
validation_generator = validation_datagen.flow_from_directory(
    val_dir,
    shuffle=False,
    seed=42,
    color_mode="rgb",
    class_mode="categorical",
    target_size=IMAGE_SHAPE[:2],
    batch_size=BATCH_SIZE)
```

Snippet for Preprocessing

c) FEATURE EXTRACTION:

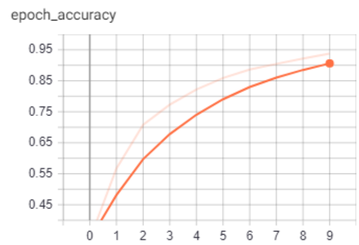
In the feature extraction, we take the picture and convert it into decreased variables. essentially each pixel of the images is taken and converted into a matrix for appearing convolutions. The manner runs throughout all the pixels in which the convolution matrix is extended with every pixel matrix. Also, we mention the variety of strides which refers to the shifting of the pixel matrix. once all the values are acquired by using multiplication, we then carry out Pooling at the matrix. Here we are the usage of Max pooling for our system for higher accuracy and extraction of capabilities. Each process i.e., Convolution and Pooling shapes an epoch. Now to improve the system accuracy we perform some of the epochs, but this could purpose to grow inside the range of parameters. As a result, following these steps we get to extract specific capabilities from the snapshots. these specific capabilities are then sent for similar approaches. Fig. 1 depicts pictorial information for the below piece of code.

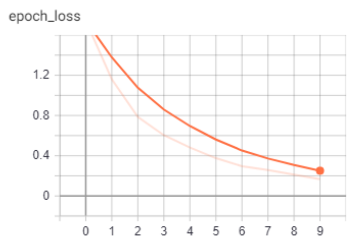
```
model = Sequential()


model.add(Conv2D(filters=32, kernel_size=(3,3),input_shape= IMAGE_SHAPE,padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(filters=64, kernel_size=(3,3), padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(filters=128, kernel_size=(3,3),padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(filters=256, kernel_size=(3,3),padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
results = model.fit(
                    train_generator,
                    epochs=10,
                    steps_per_epoch=train_generator.samples//train_generator.batch_size,
                    validation_data=validation_generator,
                    validation_steps=validation_generator.samples//validation_generator.batch_size,
                    callbacks=[tensorboard]
                   )
```

Snippet for Feature Extraction

VI. ACQUIRED RESULTS

We can capture plant diseases, show expected results, view snapshots, and use deep learning models that include plant names and a brief description of the disease. This is because the functional requirements considered in this task are successfully met with all useful necessities.





The Above graphs depicts training accuracy and loss respectively changing from each epoch. Accuracy is increasing from each epoch where as loss is reducing from each epoch. After 10 epochs, we get a training accuracy of 92.42% and loss of 19.86%. we chose 10 epochs because we found it best for the model as the accuracy and loss are good.

Below are some of the model results:

In the above result,

i. When we fed the model with Coffee_Healthy leaf picture then it has predicted the disease class as Coffee_Healthy, which is correct with an accuracy of 75.94%.

ii. When we fed the model with Cinnamon_Healthy leaf picture then it has predicted the disease class as Cinnamon_Healthy, which is correct with an accuracy of 85.93%.

In the above result,

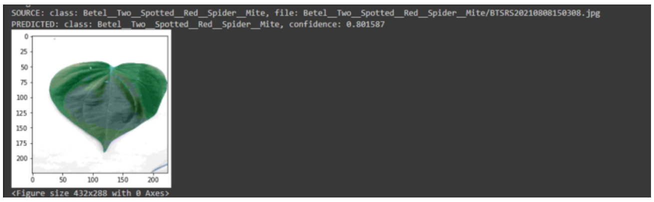
When we fed the model with Coffee_Cercospora_Leaf_Spot leaf picture then it has predicted the disease class as Coffee_Cercospora_Leaf_Spot, which is correct with an accuracy of 75.08%.

In the above result,

When we fed the model with Cinnamon_Gall_Forming_Mites leaf picture then it has predicted the disease class as Cinnamon_Gall_Forming_Mites, which is correct with an accuracy of 63.66%.

In the above result,

When we fed the model with Manioc_Healthy leaf picture then it has predicted the disease class as Manioc_Healthy, which is correct with an accuracy of 98.70%.

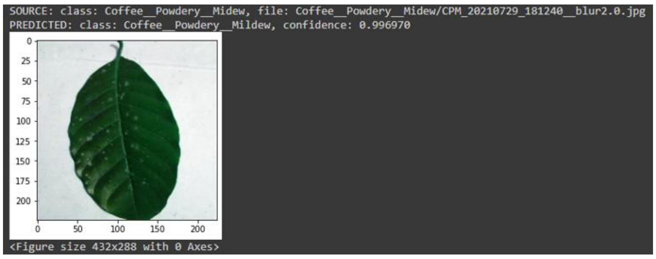


In the above result,

When we fed the model with Betel_Two_Spotted_Red_Spider_Mite leaf picture then it has predicted the disease class as Betel_Two_Spotted_Red_Spider_Mite, which is correct with an accuracy of 80.15%.

In the above result,

When we fed the model with Manioc_Mosaic_Disease leaf picture then it has predicted the disease class as Manioc_Mosaic_Disease, which is correct with an accuracy of 94.52%.



In the above result,

When we fed the model with Coffee_Powdery_Mildew picture then it has predicted the disease class as Coffee_Powdery_Mildew, which is correct with an accuracy of 99.69%.

In the above result,

When we fed the model with Cinnamon_Gall_Forming_Mites leaf picture then it has predicted the disease class as Cinnamon_Healthy, which is incorrect.

In the above result,

When we fed the model with Betel_Healthy leaf picture then it has predicted the disease class as Betel_Two_Spotted_Red_Spider_Mite, which is incorrect.

VII. CONCLUSION

Currently, the deep learning model has an accuracy of 82.5 percent. The validation and test datasets are used to inform the approach of this CNN architecture and the model's outcome. CNN is used to detect diseases. The model is essentially tested on specific types of plants and their few diseases specifically. As an expansion of the project, the number of plant classes and their diseases will be increased in future. The model will also be enhanced by expanding training and testing parameters.

This project displays the potential of deep learning techniques for crop disease identification. Their findings are genuinely promising for the development of the latest agricultural gear that could make contributions to an extra sustainable and cosy food manufacturing.

VIII. REFERENCES

1 Apsjournals.apsnet.org.2020. Plant Disease Detection by Imaging Sensors – Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping.

2 Descriptor, F., 2020. Feature Descriptor | Hog Descriptor Tutorial.

3 Learning, C. and Pai, A., 2020. ANN Vs CNN Vs RNN | Types of Neural Networks. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types of-neural-networks-in-deep-learning/> [Accessed 6 July 2020].

4 M. S. Arya, K. Anjali, and D. Unni, "Detection of unhealthy plant leaves using image processing and genetic algorithm with Arduino," 2020 International Conference on Power, Signals, Control and Computation (EPSCICON), Thrissur, 2020.

5 Z. N. Reza, F. Nuzhat, N. A. Mahsa and M. H. Ali, "Detecting jute plant disease using image processing and machine learning," 2019 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, 2019, pp. 1-6, DOI: 10.1109/CEEICT.2019.7873147.

6 P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," 2018 Conference on Advances in Signal Processing (CASP), Pune, 2018, pp. 175-179, DOI: 10.1109/CASP.2018.7746160.

7 T. Mehra, V. Kumar, and P. Gupta, "Maturity and disease detection in tomato using computer vision," 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, 2016, pp. 399-403, DOI: 10.1109/PDGC.2016.7913228.

8 R. P. Narmadha and G. Arulvadivu, "Detection and measurement of paddy leaf disease symptoms using image processing," 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2017, pp. 1-4, DOI: 10.1109/ICCCI.2017.8117730.

9 R. M. Prakash, G. P. Saraswathy, G. Ramalakshmi, K. H. Mangaleswari, and T. Kaviya, "Detection of leaf diseases and classification using digital image processing," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2017, pp. 1-4, DOI: 10.1109/ICIIECS.2017.8275915.

10 International Journal of Innovative Technology and Exploring Engineering, 2019. An Android Application for Plant Leaf Disease Detection using Convolution Neural Network. 8(12), pp.4133-4137.

11 Descriptor, F., 2020. Feature Descriptor | Hog Descriptor Tutorial. [online] Analytics Vidhya. Available at: [Accessed 6 July 2020].

12 Learning, C. and Pai, A., 2020. ANN Vs CNN Vs RNN | Types of Neural Networks. [online] Analytics Vidhya. Available at: [Accessed 6 July 2020]

13 OpenGenus IQ: Learn Computer Science. 2020. Convolutional Neural Networks (CNN). [online] Available at: [Accessed 7 July 2020].

14 Arunnehru, J., B. S. Vidhyasagar, and H. Anwar Basha. "Plant Leaf Diseases Recognition Using Convolutional Neural Network and Transfer Learning." In International Conference on Communication, Computing and Electronics Systems, pp. 221-229. Springer, Singapore, 2020.

15 Cs231n.github.io. 2020. Cs231n Convolutional Neural Networks for Visual Recognition. [online] Available at: [Accessed 7 July 2020].

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE