International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

LoCoFarm

Adharsh.S[1], Susmitha.S[2], Indhumathi.G.V[3], G.Mervin George[4]

[1 2 3] **BCA(Artifical Intelligence) Rathinam college of Arts and Science Coimbatore,tamil Nadu**

[4] **M.Tech Senior Faculty, Inurture Education Solutions Rathinam group of institutions, Coimbatore mervin.g@inurture.co.in**

ABSTRACT:

Our smartphone application uses deep learning Convolutional Neural Network (CNN) algorithms to accurately diagnose plant illnesses. The program can swiftly evaluate an image of the diseased plant that is taken with a smartphone camera and diagnose the condition. The application makes use of a pre-trained CNN model that was developed using a sizable dataset of pictures of both healthy and ill plants. The algorithm can effectively identify the type of illness afflicting the plant by identifying patterns and characteristics in the picture that are suggestive of a certain disease.

With the help of our app, farmers and gardeners can quickly and easily identify plant illnesses, enabling them to take the necessary steps to stop future crop loss and damage. With a straightforward and intuitive UI that anybody can use, the app is user-friendly and accessible regardless of their degree of technological skill.

Overall, our software offers a creative and practical answer to the diagnosis of plant disease, assisting in increasing crop yields and promoting sustainable agricultural methods. Farmers and gardeners may swiftly find answers using a chatbot and the weather, supporting sustainable agriculture and boosting crop yields.

Keywords— Convolutional Neural Network, Activation function, Android Studio, Jupyter notebook, Yolov7,Detectron2,openAI API(Application Interface), OpenWeatherMap API,HTTP(Hypertext transfer protocol)

Introduction

Plant diseases pose a substantial resulting in severe production losses and economic harm. Early identification and diagnosis of plant diseases is critical for successful disease management, crop loss reduction, and food security. According to the United Nations Food and Agricultural Organization (FAO), the world's population will reach 9.1 billion by 2050. To meet the nutrient demands of such a large population, the food growth rate need be boosted to 70% by 2050**[2].** Conventional illness detection and diagnosis methods rely on visual inspection by human specialists, which can be time-consuming and error-prone. As a result, the development of automated plant disease detection systems is critical for enhancing disease diagnostic speed and accuracy.

Deep learning is a machine learning discipline that has made tremendous success in a variety of computer vision applications, including picture categorization. Convolutional neural networks (*CNN*) are a type of deep learning model that has found widespread use in image categorization applications. *CNN* have been demonstrated to achieve high accuracy in the identification of various objects, including plants and plant diseases.

Accurate disease management systems are required to identify and segment maize disease lesions and quantify their severity under complicated field circumstances. Although they proposed deep learning approaches are becoming more popular for recognizing single illnesses, there is still a lack of effective models for identifying numerous diseases and segmenting lesion regions for severity assessment under field settings. A bespoke dataset of handheld photos of maize leaves diseased with Gray Leaf Spot (GLS), Northern Leaf Blight (NLB), and Northern Leaf Spot (NLS) diseases was utilized in this work to build a novel two-stage semantic segmentation strategy for recognizing corn illnesses and estimating their severity [**4**].

In this research, *they* present a deep learning-based technique for identifying plant diseases using a smartphone application. The suggested method employs a CNN to learn and extract characteristics from plant photos recorded by a mobile device, followed by a classification algorithm to forecast illness based on the recovered features. The suggested technique performs well on a *publicly* accessible dataset, attaining high accuracy in the detection of plant diseases. Machine Learning and Computer Vision are proving beneficial in practically every subject since they may provide more promising results at cheaper prices. Further study on the uses of these technologies is being undertaken as time passes the agriculture business is beginning to rely on Deep Learning-based solutions to improve productivity [**6**].

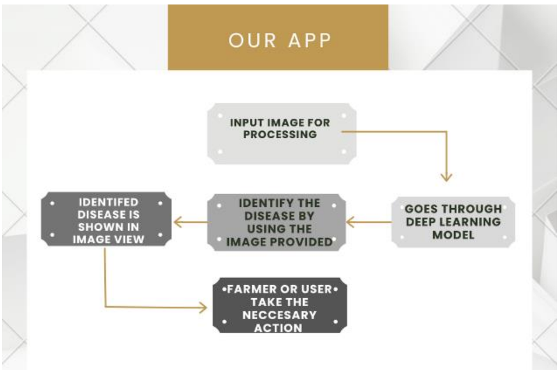

Fig 1.0 Proposed Method

CNN Algorithm

The CNN method uses backpropagation and gradient descent during training to modify the weights of the neurons in the fully connected layers and the filters in the convolutional layers in order to reduce the loss function. The objective is to minimize the difference between the projected output and the actual output, which is measured by the loss function.

In a range of computer vision applications, such as picture classification, object identification, and semantic segmentation, CNN have shown state-of-the-art performance. They have also been effectively used in other fields, such speech recognition and natural language processing. There are numerous levels in the CNN algorithm, and each one applies a different operation on the incoming data. Usually, the layers are as follows:

Convolutional Layer:

In order to extract features like edges, corners, and patterns from the input picture, this layer applies a collection of filters (also known as kernels) to the image. This layer produces a collection of feature maps, each of which captures a different element of the input image.

The primary component of convolutional neural networks (cnns), the convolutional layer, is in charge of removing pertinent elements from the input data.

Convolution Layer equation

$$O_{i,j} = \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{m=1}^{M} I_{i+k-1,j+l-1,m} * F_{k,l,m} + b$$

The Convolutional Layer processes input data, which is often an image represented as a 3D array of pixels, by applying a collection of learnable filters (also known as kernels or feature detectors) (height x width x channels). The filters are tiny matrices that go over the input image one pixel at a time, multiplying elements one by one, and adding the results to get a single output value. Convolution is the procedure in question.

Pooling Layer:

In order to lower the input's dimensionality and improve the network's computational efficiency, the pooling layer down samples the feature maps acquired by the convolutional layer. Max-pooling and average-pooling are two common pooling techniques.

Max Pooling equation

$$O_{i,j} = \max_{k,l} I_{i\times S+k,j\times S+l}$$

Average Pooling equation

$$O_{i,j} = \frac{1}{K \times L}\sum_{k=0}^{K-1}\sum_{l=0}^{L-1} I_{i\times S+k,j\times S+l}$$

Convolutional neural networks (CNN) have a layer called the pooling layer that down samples the output of the convolutional layers. The major goal of the pooling layer is to keep the key characteristics of the input while reducing the spatial dimensionality of the feature maps and improving computational efficiency.

For CNN, the Pooling Layer offers a number of advantages. The model's first benefit is that it has fewer parameters, which improves its computational efficiency and lowers the possibility of overfitting. Second, because the output values are less susceptible to minute changes in the input, the model becomes more resistant to translations and tiny variations in the input. By encouraging spatial invariance and lessening the effect of minor local input fluctuations, it also helps to minimize overfitting.

Activation Layer:

The pooling layer's output is applied to the activation layer, which uses a non-linear activation function (like ReLU) to add non-linearity and increase the model's expressiveness.

Relu:

An activation function frequently employed in artificial neural networks is the Rectified Linear Unit (ReLU). It is a straightforward activation function that has been demonstrated to be efficient in a variety of deep learning applications.

ReLU activation function equation:

$$\boldsymbol{O} = \mathbf{max}(\mathbf{0}, \boldsymbol{I})$$

Where f(x) is the activation function's output and x is its input. In essence, the ReLU function is a linear function with a zero clip. The output is equal to the input when the input is positive, and it is zero when the input is negative.

The relu function's primary benefit is that it is computationally efficient, which is crucial when working with big neural networks. Furthermore, it makes biological sense because genuine neurons' firing rates are frequently correlated with corrected functions.

It has been demonstrated that the ReLU function performs well in a wide range of deep learning applications, including object identification, picture classification, and natural language processing. It frequently serves as the activation function in a neural network's hidden layers and occasionally in the output layer as well.

Tanh:

An activation function frequently employed in artificial neural networks is the hyperbolic tangent (tanh). It is a nonlinear function that produces values in the range of and is symmetric about zero (-1, 1).

Tanh activation function equation:

$$\boldsymbol{O} = \boldsymbol{tanh}(\boldsymbol{I})$$

When the function's input is x and its output is f(x).

Similar to the sigmoid function, the tanh function is scaled and shifted such that its output values span from -1 to 1 rather than 0 to 1. Tanh function, which is likewise S-shaped like the sigmoid function, is frequently applied as an activation function in neural networks' hidden layers.

Sigmoid:

A mathematical function frequently utilized in artificial neural networks and machine learning is the sigmoid activation function. Any input value is converted to a value between 0 and 1 using a distinctive S-shaped curve.

Sigmoid function equation:

$$\boldsymbol{O} = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{I}}}$$

Where x is the function's input and exp(-x) represents the exponential function with base e (Euler's number) raised to power -x.

The sigmoid function is helpful for cases where the result must be understood as a probability since its output is always a number between 0 and 1. For instance, the output of a sigmoid function may be seen as the likelihood that the input belongs to one of the classes in binary classification issues, where the objective is to predict a binary output (for instance, 0 or 1).

The sigmoid function, however, has certain drawbacks. One of the key problems is that when the input values are far from zero, the output saturates and becomes extremely little or very big, which can slow down learning and lead to the vanishing gradient problem.

Fully Connected Layer:

Similar to a conventional feedforward neural network, the fully connected layer links every neuron in the previous layer to every neuron in the current layer. The final classification operation is carried out by this layer by mapping the retrieved features to the output classes.

$$\boldsymbol{O} = \boldsymbol{Wx} + \boldsymbol{b}$$

Each neuron in a fully connected layer receives the output from every neuron in the layer below as input and gives each input a weight. These weights are developed throughout the training process and are modified to reduce the discrepancy between the output that was anticipated and the output that was actually produced. The model then includes non-linearity due to the activation function that is applied to each neuron's output.

Related Work:

Because of the large differences in the size, shape, color, and location of leaves, efficient and effective automated localization and categorization of crop disease remains a difficult challenge. Also, fluctuations in brightness during the picture capture process of the leaves confound the detecting procedure. We attempted to address the aforementioned issues in this study by integrating a modified Centernet framework with densenet-77 at the *key points* calculation level to calculate the deep key points of input data and to localize and classify various plant diseases. Our results demonstrate that the provided approach is resistant to changes in size, rotation, color, brightness, contrast, lightning conditions, blurring, and large numbers of noisy input samples [**1**]. The cellular neural network (CNN) universal machine and supercomputer is presented by them as a novel innovation. This is the world's first algorithmically programmable analogue array computer, with real-time and supercomputer capabilities on a single chip. The CNN universal machine is detailed, with an emphasis on its programmability as well as global and distributed analogue memory and logic, high throughput through electromagnetic waves, and sophisticated cells that may also be used to simulate a wide range of PDE's [**3**].

For each step, They Used three semantic segmentation models were trained using the SEGNET, UNET, and deeplabv3+ network architectures. Semantic segmentation was employed in stage one to recover leaves from complex field backdrops. Semantic segmentation was employed in stage two to detect, identify, and quantify area coverage for disease lesions. During training, the unet model had the greatest performance for stage one, with up to 0.9422 mean weighted intersection over union (MWIOU) and 0.8063 mean boundary F1-score (MBFSCORE). The deeplabv3+ model outperformed the others in stage two, identifying illness lesions with a MWIOU of 0.7379 and a MBFSCORE of 0.5351. Lastly, the severity of the illness was calculated by determining the proportion of leaf area covered by disease lesions. In the test set, an R2 value of 0.96 was obtained, indicating that the combined (unet-deeplabv3+) model predicted the severity of three illnesses extremely closely to the actual data [**4**].

They named their approach RFE-CNN, which includes a residual channel attention block (RCAB), a feedback block (FB), elliptic metric learning (EML), and a convolutional neural network (CNN). Then, we used two parallel CNNs to extract the main characteristics of healthy and damaged wheat leaves. Second, they optimized the fundamental characteristics using residual channel attention blocks. Lastly, they trained the preceding characteristics using feedback blocks. Lastly, we processed and classified these data using a CNN and elliptic metric learning [**5**].

Several approaches for detecting plant diseases have recently been developed under the influence of Deep Neural Networks in Computer Vision. Yet, the lack of openness in this sort of study makes its use in the actual world less appealing. We present rests (Residual Teacher/Student) architecture, which may be utilized as a visualization and classification tool for plant disease diagnostics. Rests is a tertiary adaption of the previously recommended Teacher/Student architecture. Rests is built on a Convolutional Neural Network (CNN) framework that includes two classifiers (ResTeacher and ResStudent) as well as a decoder [**6**].

Proposed Methodology

Currently we have Used deep learning CNN algorithms to identify plant diseases is a novel strategy for increasing crop yields and advancing sustainable agricultural practices. In order to guarantee that the photos are of the highest quality and are in a format that the model can utilize, the preprocessing step in the proposed technique entails gathering a sizable dataset of photographs of both healthy and ill plants.

Jupyter Notebook

An open-source online tool called Jupyter Notebook is used to create and share documents with real-time code, equations, visuals, and text. It supports several different programming languages, including Julia, Python, and R. The Notebook interface makes it simple to experiment with data and rapidly view findings by allowing users to write and run code in cells. It has grown in popularity as a tool for data science and scientific computing because it makes procedures for data analysis and research repeatable. Jupyter Notebook may be used for both teaching and learning, giving students an interactive setting in which to practice coding and conduct data experiments.

The preprocessed dataset is then used to train the deep learning CNN model, with an emphasis on accuracy and performance improvement strategies including data augmentation. When the model has been trained, its performance and accuracy are assessed using a different dataset of photos.

The next stage after creating a trained and tested model is to create the app itself, which should have an intuitive user interface that enables users to rapidly diagnose a plant's condition by taking a photo of it. The deep learning model, along with any additional required parts like image processing algorithms and user interface components, must all be integrated into the app as the last stage.

Android Studio

Popular Integrated Development Environment (IDE) Android Studio is used to create mobile apps for the Android operating system. A code editor, debugging tools, and a user interface designer are just a few of the many features and capabilities it offers developers to simplify the development process.

Android Studio can be utilized to construct the app's user interface, as well as to incorporate the deep learning CNN algorithm and image processing algorithms, in the instance of developing a plant disease diagnosis app. With the help of the Android Studio environment, developers can quickly test and debug their apps on a variety of Android devices to make sure they function properly and operate smoothly. In general, Android Studio is a crucial tool for creating functional and high-quality Android applications.

Therefore, irrespective of the amount of technical competence, the plant disease diagnosis app created using this technique should be reliable, accurate, and available to farmers and gardeners. The software can aid in preventing future crop loss and damage by giving users a quick and simple way to identify plant diseases, which will support sustainable agricultural practices.

Chatbot

The App implements a chatbot for Android that makes use of Firebase and the OpenAI API. Users may ask inquiries into the chatbot and get answers using the OpenAI API. The program initializes UI components including the ImageButton, RecyclerView, and EditText while also making API requests to OpenAI using OkHttp.

Messages are also saved to the Firebase database. When a user enters a query, OkHttp is used to send the query to the OpenAI API and retrieve the answer asynchronously. The answer is then shown in the RecyclerView and added to the messageList. The mechanism to display a typing indication while awaiting a response is also included in the implementation.

The code may be enhanced with new capabilities like natural language processing, sentiment analysis, and machine learning techniques and is helpful for creating chatbots on Android utilizing Firebase and the OpenAI API.

Weather Report

The Android application that accesses the OpenWeatherMap API using the user's current location and displays weather information on the device's screen.

The app initially gets the user's current location and then initializes location services. Once the location has been determined, the app uses latitude and longitude coordinates to submit an HTTP call to the OpenWeatherMap API to receive weather information.

The important information, such as temperature, humidity, wind speed, dawn and sunset times, and weather description, are extracted from the weather data, which is acquired in JSON format. The device's screen then shows the retrieved data.

The software shows an error message on the screen if a problem arises when processing or retrieving data. This program uses a variety of Android capabilities to show the weather information, including AsyncTask for background processing, LocationManager for location tracking, and several view widgets. Additionally, for the app to work correctly, the user must give location rights.

In conclusion, this app shows how location-based services and APIs can be utilized to give consumers access to real-time weather information on their devices.

Result & Discussion:

The ability to identify illnesses in plants and flowers is crucial for agriculture and horticulture. Thanks to deep learning algorithms, mobile apps can now swiftly diagnose and treat damaged plants by *analyzing* photos. These apps can help farmers and gardeners increase yields and reduce losses. However, there is a need for further study to enhance the accuracy and adaptability of these apps, and ensure they are user-friendly for a diverse range of users. While deep learning methods have revolutionized the field of plant and floral disease identification, there is still room for improvement in terms of precision and applicability to different regions and climates. To harness the full potential of these apps, continued research and development is required. Overall, deep learning in plant and floral disease identification offers promising opportunities for the future of agriculture and horticulture.

Deep learning algorithms can classify new images and identify diseases by using patterns learned from previous images. Convolutional Neural Networks (CNN) and Transfer Learning are two common deep learning architectures used for plant disease detection.

Further Work

YOLOV7

The computer vision and machine learning community are buzzing about the YOLOv7 algorithm. The fundamentals of YOLOv7's operation and what makes it the greatest object detecting algorithm currently in use will be covered in this article.[7]

The most recent YOLOv7 algorithm outperforms all earlier object detection algorithms and YOLO iterations in terms of speed and precision. It can be taught significantly quicker on tiny datasets without any pre-learned weights than other neural networks and requires technology that is several times less expensive.

Real-time object identification is a crucial job in computer vision and frequently a crucial part of computer vision systems. Video analytics, robotics, autonomous cars, multi-object tracking and object counting, medical image analysis, and other applications employ real-time object identification models. When performing image recognition tasks, an object detector uses an object detection method that predicts bounding boxes and class probabilities for each object in the input image (see the example image below).

The majority of methods extract characteristics from the picture to estimate the likelihood of the learnt classes using a convolutional neural network (CNN).

Fig 1.1 Yolov7 prediction

Detectron2

The Detectron object identification platform has grown to become one of Facebook AI Research's (FAIR) most frequently used open-source initiatives since its launch in 2018. We are now offering the second version of the library, which includes significant improvements for both study and production use, to build upon and expand this effort. It may be found here.[8]

A complete overhaul of the original Detectron that began with maskrcnn-benchmark. The technology has now been put into use in PyTorch. Detectron2 now has a new, more modular design that makes it adaptable, expandable, and capable of offering quick training on one or more GPU servers. High-quality implementations of cutting-edge object identification methods are included in Detectron2, including DensePose, panoptic feature pyramid networks, and several iterations of the ground-breaking Mask R-CNN model family, which was also created by FAIR. Modern research projects may easily be implemented using it without having to fork the entire software because to its expandable nature.

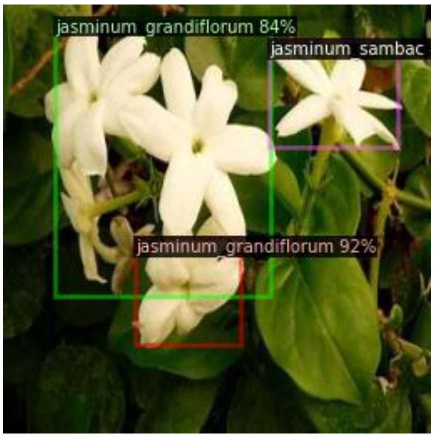

Fig 1.2 Detectron2 Prediction

Disadvantages:

Detectron2 Disadvantages:

Training complexity: Although Detectron2 offers a versatile and modular API for creating bespoke object detection models, training and optimizing these models might be more difficult than with other frameworks.

Hardware requirements: When compared to YOLOv7, Detectron2 requires more powerful hardware, particularly for training. It may become more challenging to operate on embedded or low-end computers as a result.

Runtime performance: While Detectron2 typically outperforms YOLOv7 in terms of accuracy, it may be slower in terms of real-time performance, especially on less powerful hardware.

YoloV7 Disadvantages:

Limited adaptability: YOLOv7 predicts bounding boxes and class probabilities from complete photos using a single neural network. It is quicker and lighter than some other frameworks because of this, but it may not be as adaptable for jobs requiring customized object identification.

Accuracy: Although YOLOv7 is quick, some alternative object detection frameworks may be more accurate. It could have trouble, in instance, identifying tiny items or objects with intricate forms.

Customization: Although YOLOv7 is open-source, customization and fine-tuning may be more challenging when compared to other frameworks like Detectron2.

Conclusion

In conclusion, the deep learning-based smartphone app that is being developed to diagnose illnesses in plants and flowers has the potential to have a big influence on the agricultural sector. The app can assist farmers and gardeners maintain their plants and harvests by increasing the effectiveness of disease control. The software may aid in the discovery of novel therapies and advancements in plant pathology research.
The conversion of the Detectron2 and Yolov7 trained model conversion to the android studio compatible file is the major disadvantage of the Yolov7 and Detectron2.

So for now we have finalized the current app with CNN algorithm.

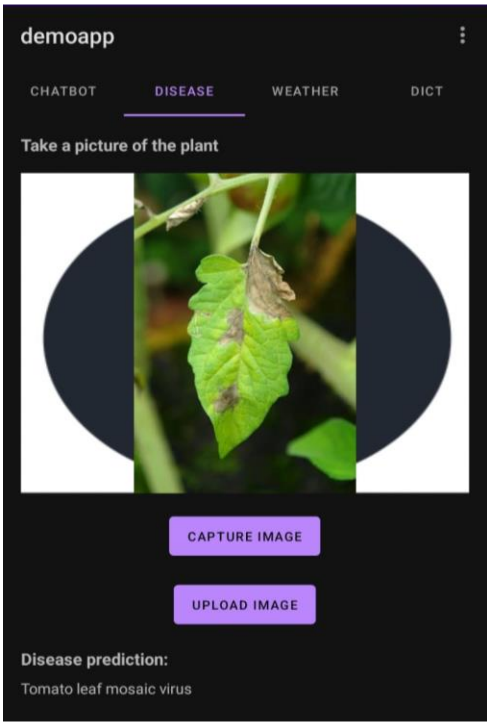

Fig 1.3 Demo screenshot

Mulitilingual support

With the use of a Spinner UI component, users of this Android app may choose between **English and Tamil as** their language. The app applies the user's language option to the app's settings using the Locale and settings classes after saving it in Shared Preferences.

The user interface (UI) is modified when the user chooses a language, and the language choice is stored for further use. A logout button that logs the user out and takes them back to the login screen is also included in the code. In general, the app aids in giving consumers a bilingual user experience and a seamless change between app modes.

REFERENCE:

1. Albattah, W., Nawaz, M., Javed, A., Masood, M., & Albahli, S. (2022). A novel deep learning method for detection and classification of plant diseases. Complex & Intelligent Systems, 1-18.
2. Bruinsma, J. (2009). The resource outlook to 2050: by how much do land, water and crop yields need to increase by 2050? Expert meeting on how to feed the world in 2050. Http://www. Fao. Org/wsfs/forum2050/wsfs-background-documents/wsfs-expert-papers/en/.
3. Roska, T., & Chua, L. O. (1993). The CNN universal machine: an analogic array computer. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 40(3), 163-173.
4. Divyanth, L. G., Ahmad, A., & Saraswat, D. (2023). A two-stage deep-learning based segmentation model for crop disease quantification based on corn field imagery. Smart Agricultural Technology, 3, 100108.
5. Xu, L., Cao, B., Zhao, F., Ning, S., Xu, P., Zhang, W., & Hou, X. (2023). Wheat leaf disease identification based on deep learning algorithms. Physiological and Molecular Plant Pathology, 123, 101940.
6. Shah, D., Trivedi, V., Sheth, V., Shah, A., & Chauhan, U. (2022). Rests: Residual deep interpretable architecture for plant disease detection. Information Processing in Agriculture, 9(2), 212-223.
7. Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.
8. Abhishek, A. V. S., & Kotni, S. (2021). Detectron2 object detection & manipulating images using cartoonization. Int. J. Eng. Res. Technol.(IJERT), 10.