© November 2024 | IJIRT | Volume 11 Issue 6 | ISSN: 2349-6002

Deep Learning Based Plant Disease Detection with a Farmer-Friendly Interface

Dr. S.T. Patil[1], Hriday Badani[2], Aarya Gavaskar[3], Janhavi Awere[4], Ahmad Ali Sayyed[5]
[1,2,3,4,5]*Department of Computer Science Engineering Vishwakarma Institute of Technology Pune, India*

Abstract—The work temporarily drops metadata and instead focuses on a highly sophisticated model Convolutional Neural Network to predict potato leaf diseases. There are 3 categories of Potato leaves namly early blight, healthy and late blight , that PlantVillage dataset offers which is used for training the model. With the help of picture preprocessing methods, such as augmentation and rescaling allows better generalization to make precise diagnosis in different scenarios. Once a trained model has shown high accuracy during validation, it is provided to farmers using an API on Google Cloud where they can upload their images for disease prediction in real-time. This is then supplemented by a convenient user-friendly web interface which shows these predictions together with information regarding the detected condition, its etiology and potential therapies. The React, PHP-built web platform having MySQL and uses Leaflet for integration with HTML, CSS & JS. Using js for user location mapping, and Google Translate support to ensure that farmers are using the service in their native tongue. Apart from diagnosing diseases the platform also provides services like a blog, selling equipment and fertilizers, data-based insights about agricultural trends etc. In addition, the system stores disease outcomes and agricultural information of farmers in a MySQL database which can be used for further research on agriculture. Showcasing how deep learning and cloud technology driven by AI can disrupt precision agriculture through providing the marginal farming communities with access to them. Plant Pixel is intended to serve farmers as a comprehensive resource in addition to a disease detection tool. It provides opportunities for growth by forming alliances with governmental organizations, non-profits, and agricultural technology stakeholders.

Keywords— Potato leaf disease detection, Convolutional Neural Network, PlantVillage dataset, deep learning in agriculture, Google Cloud deployment, real-time disease prediction, multilingual farmer support, React web interface, precision agriculture, AI in farming, Leaflet.js mapping, agricultural research database, cloud-based disease detection, Plant Pixel platform, agricultural trends analysis.

I. INTRODUCTION

Plants, In addition to supporting ecosystems and the economy, plants also help to mitigate the consequences of climate change. Most agricultural plants are susceptible to a number of diseases.

[1]States that effective disease prevention and management strategies must be used to limit production losses and guarantee agricultural sustainability, highlighting the necessity of continuous crop monitoring. These are few of the practices recommended by the phytopathologists [1]. A plant's diseased leaves can frequently be identified by a noticeable change in the texture and colour of the leaves. Every disease typically results in unique alterations to the colour, texture, and shape of the leaf, stem, or root [2]. Therefore, it is possible to visually assess the quality, colour, and texture of leaves to ascertain the health of the plants [3, 4].

On the other hand, a wide range of diseases brought on by pathogens including bacteria, viruses, and fungi can affect plants.
Plant diseases have a significant economic impact, costing billions of dollars annually in lost food, fibre, and production systems. These diseases not only jeopardise agricultural yield but also pose major challenges to global food security.
Detecting and managing plant diseases in a timely manner is crucial to prevent widespread crop damage and mitigate economic losses. Traditional disease identification used to heavily depend on manual inspection by the agriculturalists which is labour intensive and error-prone. Some diseases that can spread, need to be detected and treated as early as possible. [5].

In the most recent years, on account of advances in technology — especially applied to deep learning-based strategies we are now marching toward automated plant disease detection systems. Deep learning, a subdomain of machine learning can overcome these issues as it does not require manual feature engineering and due to its fallibility in recognizing patterns the incredible development of object recognition and classification using Computer Vision and Artificial Intelligence (AI) approaches has been seen ([6]). These systems are powered by convolutional neural networks (CNNs) [8], thought to be ahead of human performance in certain large-scale reconnaissance tasks.

This research proposes to develop an automated system for finding diseases in potato leaves with deep learning. The proposed method categorize a leaf into healthy or unhealthy, and what type of disease & by using CNNs make use the powers to it. Images: Image Processing as one of the most important and leading methods to cure plant diseases In [8], [9] detection and identification are highlighted..

Using machine learning tools showcased a good response to the problem of early disease detection and management while also creating an image processing module for integration into mobile devices thereby en lighting African farmers. This will help loss of crops to be minimised; food security enhance, and improvement in sustainable agriculture practices.

This study aims to automate the identification process using advanced image recognition techniques, by utilizing convolutional neural networks (CNNs)and that will be a game changer in plant disease detection. CNNs are best for this task and they have the ability to derive advanced features from images, like color gradients, texture patterns or spatial relationships. Through the use of CNNs, this research attempts to address these prerequisites by allowing a grown leaf image with subtle visual cues that are hard to recognise even for human in manual inspection method to be rapidly and accurately classified into being healthy or diseased. These methods would shorten disease management protocols by speeding up diagnostics, accelerating crop loss mitigation and increasing the rate of progress to underpin global sustainability in food security while also safeguarding biodiversity.

II. METHODOLOGY

Deep learning has gone on to reduce this important challenge of feature engineering in machine learning. This development results in a significant reduction of the requirement for domain knowledge, allowing deep learning models to work efficiently with little human intervention. Deep learning is centered around the principles of Artificial Neural Networks (ANNs), or mathematical models that aim to mimic the basic functions of a brain using interconnected neurons and synapses. This can be done via a well-know neural network implementation package TensorFlow that provides everything you need for building and training these models. Because of this, TensorFlow can be used for image classification and text based classification as well making it a handy tool to have in the machine learning kit.The figure 1 shows the flowchart along with the methodology implemented at each step.

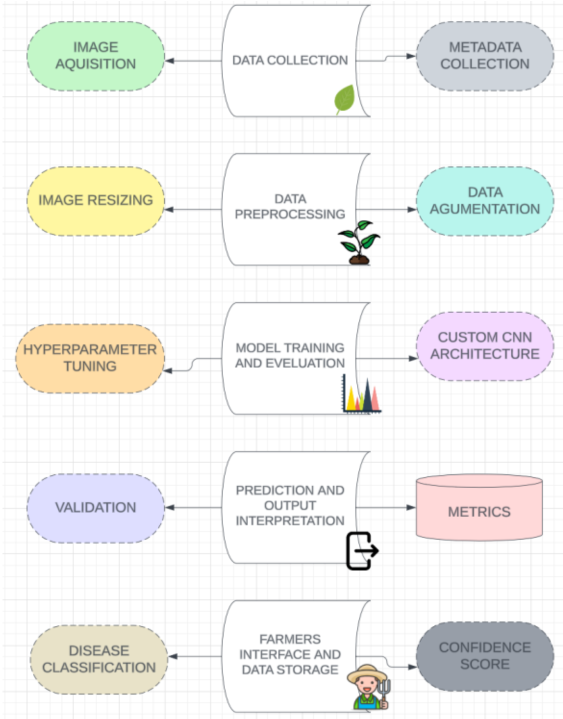

Fig 1 : Project Methodology

A. Convolutional Neural Networks

CNNs to identify the disease in a plant's leaves, convolution neural networks, or CNNs, are employed. CNN is an improved version of a basic ANN that produces superior visual results. because images—any image—contain recurring patterns of certain objects. Several essential layers make up a Convolutional Neural Network architecture, which is intended to process and learn from picture input. The first layer of the architecture is the input layer, which takes photographs in a predetermined format and size, frequently with RGB color channels included. Convolutional layers are the initial layers to apply many filters to an input image in order to extract characteristics like edges and textures.

Activation layers come after these and often employ the ReLU (Rectified Linear Unit) function to add non-linearity and improve feature representation. The most notable features are retained in the feature maps but their spatial dimensions are significantly decreased by pooling layers, which frequently use max pooling. The benefits of adding more convolutional and pooling layers in the algorithm is that it wakes each level to learn a variety patterns from simpler ones at lower layers;better model. So after a few convolutional and pooling layers, the network goes to flattening layer which takes those multi-dimensional feature maps back in shape of one dimensional vector. These vectors then get processed through a series of fully connected layers, or perceptron-based networks where every neurone in the layer above is connected to very other neurone below. The final output layer of a CNN (usually applies SoftMax function for classfication) used to classify the learned features and give predictions. These predictions are further elaborated into something meaningful (such as class labels, probabilities or diagnostic data) in the output interpretation phase and hence completes overall design structure of neural network model.

A1. Convolutional and Pooling Layer

One more important thing inside the convolutional neural networks (CNNs) is that, a convolational layer which automatically learns and detects features from the input image without any need to specify where these feature exist then in different layers they are learned hierarchically. It convolves(the convolution operation) the input by running a filter or kernel across the image to generate 3x32x32 and forming feature map.

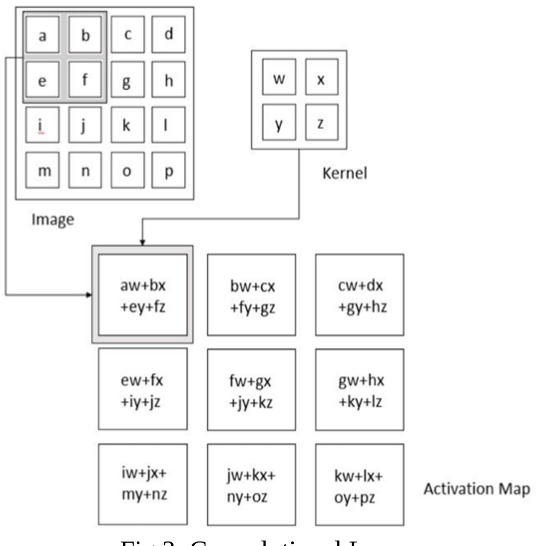

Fig 2: Convolutional Layer

We start with two of the essential operations in Convolution Neural Networks (CNNs), namely, Convolution and pooling. Pooling is used to reduce the size of a picture and convolution is performed in order to determine edges for patterns in an image. A note on Tensorflow Keras: End-to-End Convolutional Operation — The convolution operation applies a filter to the input image. It can be written using math as follows:

$$(I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n)$$

III is the input image matrix.

KKK is the convolutional kernel (filter).

(i,j)(i, j)(i,j) represents the position of the output pixel in the feature map.

Feature Map — output of filtering. This involves applying the filter across its input image, each location reflecting some features of edges etc., and creating a new map for global summary statistics like those you requested.

Stride: The number of pixels we shift our filter by is called its stride. The filter moves one pixel at a time when its stride value is 1, and two pixels at a time when its stride value is 2.

Padding — Additional pixels were added to the input image's boundaries in order to control the output's size. This is due to the padding, which keeps the feature map's dimensions constant or regulates their decline.

Activation Function: Then an Activation function is applied (like ReLU — Rectified Linear Unit) after the convolution. The ReLU function is represented as:

$$\mathrm{ReLU}(x) = \max(0, x)$$

This function introduces non-linearity, which allows the network to learn more complex patterns.

1) Example Calculation

For a given input image matrix III and a 3x3 filter KKK, the value of a pixel (i,j)(i, j)(i,j) in the output feature map is calculated by:

$$\mathrm{Output}(i,j) = \sum_{m=0}^{2} \sum_{n=0}^{2} I(i+m, j+n) \cdot K(m,n)$$

The convolution operation helps to extract essential features from the input image and these further extracted features are being utilized in more layers of CNN for categorization or classification.

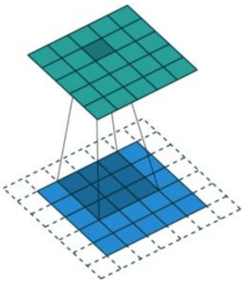

Fig 3: Applying Filter

API and a Jupyter notebook are used for the training of these models. TensorFlow's high-level API for creating and refining deep learning models is called Keras.

A2. Fully Connected Layer

The FC layer is essential in the structure of neural networks. Most commonly, it is located on the end part of the CNN's stacks. It is described as a layer that connects each neuron with every neuron in the previous layers. This layer "flattens" and turns feature maps into a single-dimensioned vector and performs attachments or regression at the last layer.

Dense Connections: The entire neuron in the fully connected layer gets input from the neuron which is situated on the previous layer. In terms of scope, the network may use all features that were available on the previous layers due to its high connectivity.

Mathematical Representation:

Let x be the input vector to the FC layer with dimensions n×1.

- Let W be the weight matrix of dimensions m×n, where m is the number of neurons in the FC layer.
- Let b be the bias vector of dimensions m×l.

The output vector y of the FC layer is computed as:

y=W·x+b

2. Activation Function: To add non-linearity, an activation function is added after the output has been computed.

Common activation functions include:

- Sigmoid Function:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

This function outputs values between 0 and 1 and is often used in binary classification tasks.

- SoftMax Function:

$$\mathrm{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

This function converts the output into probabilities,where xi represents the output of the i-th neuron, in the neural network.

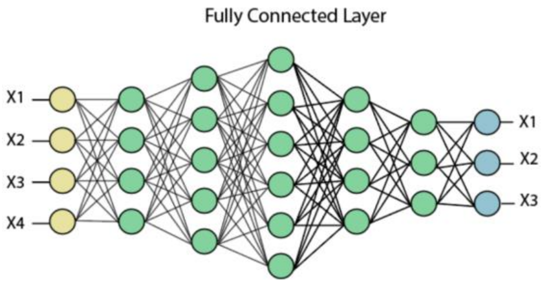

Fig 4: Fully connected layer

The fully connected layer is crucial for interpreting the features extracted by preceding layers and for producing the final output of the network. It consolidates information from the entire network and applies a dense, linear transformation to produce predictions or classifications.

B. Discussion of Dataset

The dataset employed in this study is the PlantVillage dataset, obtained from Kaggle, which includes a vast collection of plant leaf images. To ensure a robust model, the dataset is diverse in terms of plant species, disease variations, lighting conditions, and backgrounds. Two databases are used to identify plant diseases. The first dataset contains 15 classes, whereas the second contains 38 classes. There are a set number of photographs for each plant in both databases. The 38 classifications of various plants that can be found in the PlantVillage dataset serve as the foundation for the project's final outcomes. It is also freely available on the internet. These classes and the dataset are described in Table I. This table can be used to find the total number of photographs in each class. This collection contains fourteen different species of plants. For each plant, images of both healthy and diseased leaves are available. Apple and tomato plants are shown in most of the pictures. The classifications with the fewest are raspberries, soybeans, and squash. In this particular paper we'll be focusing only on the potato dataset as its one of the mainly grown crops in India. The photos of several leaves that are part of the dataset are displayed below. The dataset is split into three sections: one for testing one for training and one for validation. The dataset is split at random into 80/20 ratios. 20% is used for testing and validation and the remaining 80% is for the training dataset. 2152 photos were used for the model's training.

Table- I (a): Dataset Description

Class	Plant Name	Healthy or Diseased	Disease Name	Images (Number)
C_0	Apple	Diseased	Apple_scab	2016
C_1	Apple	Diseased	Black_rot	1987
C_2	Apple	Diseased	Cedar_apple_rust	1760
C_3	Apple	Healthy	-	2008
C_4	Blueberry	Diseased	-	1816
C_5	Cherry_(including_sour)	Diseased	Powdery_mildew	1683
C_6	Cherry_(including_sour)	Healthy	-	1826
C_7	Corn_(maize)	Diseased	Cercospora_leaf_spotGray_leaf_spot	1642
C_8	Corn_(maize)	Diseased	Common_rust	1907
C_9	Corn_(maize)	Diseased	Northern_Leaf_Blight	1908
C_10	Corn_(maize)	Healthy	-	1859
C_11	Grape	Diseased	Black_rot	1888
C_12	Grape	Diseased	Esca_(Black_Measles)	1920
C_13	Grape	Diseased	Leaf_blight (Isariopsis_Leaf_Spot)	1722
C_14	Grape	Healthy	-	1692
C_15	Orange	Diseased	Haunglongbing (Citrus_greening)	2010
C_16	Peach	Diseased	Bacterial_spot	1838
C_17	Peach	Healthy	-	1728
C_18	Pepper_bell	Diseased	Bacterial_spot	1913
C_19	Pepper_bell	Healthy	-	1988
C_20	Potato	Diseased	Early_blight	1939
C_21	Potato	Diseased	Late_blight	1939
C_22	Potato	Healthy	-	1824
C_23	Raspberry	Healthy	-	1781
C_24	Soybean	Healthy	-	2022
C_25	Squash	Diseased	Powdery_mildew	1736
C_26	Strawberry	Diseased	Leaf_scorch	1774
C_27	Strawberry	Healthy	-	1824
C_28	Tomato	Diseased	Bacterial_spot	1702

Fig 5: Kaggle PlantVillage Dataset

C. Model Description

The preset of convolution layers is a feature that the Convolutional Neural Network model has when ensuring the automated identification of the plant diseases. These other layers are relevant therefore integrating them in the study of the input images is imperative because each of them has a different filter to focus on local features such as edges, textures and patterns in the images. Every convolutional operation is complemented with a ReLU activation function which embeds a nonlinear part onto the network. This is attributable to the fact that the non linearity is necessary so that the network can comprehend and represent intricate structures and relationships inherent in the image data.

Typically encountered in between the convolutional layers, there exists a pooling layer that is max-pooling layer. Such layers help in down-sampling the feature maps developed during the convolutional layers and hence reduce the spatial dimensions of the maps. Increasing convolutional and pooling layers to the algorithms helps to add progressively more complicated patterns. Subsequent to many rounds of convolution and pooling, the network then outputs the fully connected layers which flatten the tape that contains multi-dimensional feature maps into a vector. Following this, the vector is passed on to the fully connected layers where all neurons within one layer are connected to all neurons within the next layer in order to achieve high-level feature representation throughout the layers. Additionally, the last output layer utilizes functions such as softmax in order to classify parameters within the succeeding layers. The last stage of the overall structure is the output interpretation phase which provides useful outputs for every prediction e.g. class labels, probabilities or more complex diagnostic information.

The final layer of a is the output layer, which typically has a softmax activation function. This layer generates a probability distribution. By using this softmax function, the model ensures that all probabilities have a summation of one, making the predictions easy to interpret for classification tasks.

In addition to predicting the disease, the model also provides valuable information such as confidence scores, potential causes of the disease, and suggested treatments. This architecture is specifically tailored for image classification, making it particularly effective for detecting plant diseases. Its capacity to automatically learn and extract features from raw images enables it to deliver practical insights for agricultural use, ultimately supporting farmers with crucial information for managing crop health.

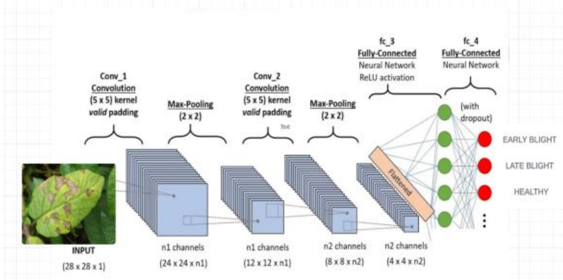

Fig 6: Custom CNN Model

Fig7: grid layout of the input image for layers to be applied

III. RESULTS AND DISCUSSIONS

The test dataset used in developing the Convolutional Neural Network model achieved an accuracy of [insert accuracy percentage]. This accuracy was reached after training for [insert number] epochs. For the training stages, a combination of both training validation accuracy and loss was performed and recorded at each epoch. Several graphs are drawn showing the training and validation accuracy and loss over time which explains the model associated with the data generalization and its learning process.

In terms of prediction outputs, the model is able to detect diseases in plants effectively. This model not only possesses the capability of diagnosing the disease for every inputed image but also other features like the confidence level, the disease agent, and treatment measures. This output is supportive as it enhances the accuracy of the prediction and enhances the quality of description for farmers on how to handle the problem of the plants' health.

The use of Google Cloud simplified model deployment and ensured its prediction capabilities were available in a reliable and scalable environment. Moreover, the model was deployed over Google Cloud Run to provide a robust web interface experience.The deployment setup encompassed creation of the right Cloud Storage resources for the models' storage and management of the prediction handling API endpoints.

The intention for building the application was to make it easy for farmers to login, and upload images of the leaf blades to determine the diseases affecting their crops. The frontend part of the application uses React and incorporates several features such as language settings and location context which are provided through the use of Leaflet.js. The backend develops model in PHP for performing API queries and storing objects in a MySQL relational database. This configuration means that the user requests, the results of predictions and the performance of the system can be mounted for intensive performance analysis.

There are however several capabilities of the site. On the Disease Detection page, farmers can predict and enter data of plant diseases with the help of geolocation and maps provided by Leaflet and databases. To eliminate language difference problems, Google Translate has been installed for many regional languages. The site provides a Blog section based on news and agricultural measures and updates, an About Us area that tells who is the site team, and a contact us area for collaboration possibilities. The let's support us section allows donations to the non afforded PLANT PIXEL project to be made. Other functionalities include a Fertilizer page where buyers can purchase locally available products, a Trends page that collates graphs and statistics on the soil and quality of plant vegetation and an Equipment section which contains donated used agricultural tools by various people.

In conclusion, the incorporation of the CNN model into a fully developed web application which is backed with a reliable cloud services makes it possible to detect any plant disease. The fact that the system can provide accurate forecasts as well as intricate details makes it possible for farmers to have access to the information required in the management of plant health as well as climatic conditions.

Fig 8: Prediction of the model

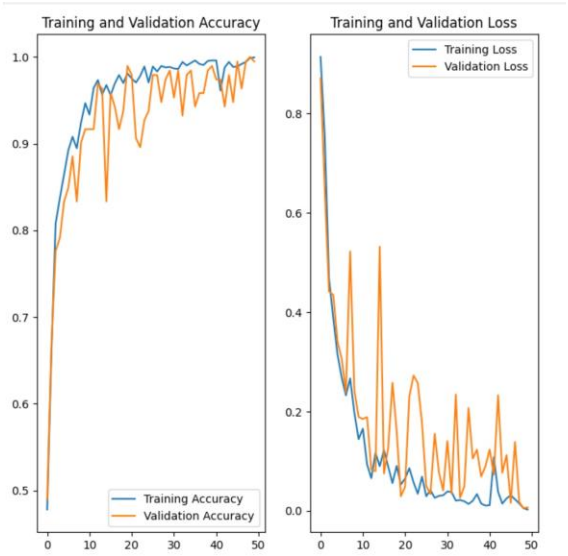

Fig 9: Accuracy and loss

Among the other Patel plant pixels websites, the Disease Detection page allows the users to upload images of the leaves of the plants and have the images analyzed. Users have to fill in their name, language (real time language through Google Translate) select a state which shows what crop is cultivated there mostly. There is an image upload button which also has a thumbnail of a sample image and preview, a button marked ' Predict' which allows the user to analyze the picture focusing on the image and some pages which show the confidence in the disease being present, the cause of it and recommended cures for it.

The website has a Leaflet.js integration for the purpose of locating the places of the farmers and Google Maps for searching fertilizer shops in the vicinity of the farms. Additional pages include About Us, Contact Us, Blog, Support Us, Fertilizers, Trends, and Equipment, providing detailed information and resources for farmers.

Fig 10: Farmer Friendly Disease detection page

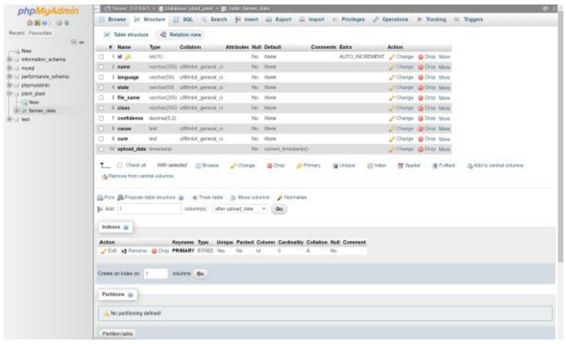

Fig 11: Database to store the farmers information

IV. CONCLUSIN AND FUTURE WORK

The Plant Pixel project represents a good combination of deep learning with web technologies in order to develop a sophisticated plant disease detection solution. The system trained with a custom Convolutional Neural Network model achieves reliable disease recognition when images of the diseased plant leaf are processed, which ensures that the farmers can most importantly carry out timely diagnosis and management of the crop health. Deployment on Google Cloud guarantees reliable and scalable service while the easy-to-use website improves user satisfaction. This system is not limited to assisting in rapid disease detection only but also helps farmers with information that enhances their agricultural practices. For any new product or service, it is standard business practice to include some constructive criticism. One of the possibilities already offered is the enlargement of the existing model, which would include more varieties of plants and diseases.

The website could be further improved through real-time chat support and connection to agricultural consulting services for the users' purpose and grow their base further. Similar considerations for growth directions include changing the visual organization of the interface for the sake of diversity and adding new languages to attract a greater number of users. Farming technology providers, healthcare and agriculture venture capitalists, government departments that deal with agriculture, food security focus NGOs, and universities may all be potential sources of key funding and could maximize its impact.

This technology, when put in mini-drones, can help in detecting plant diseases in cultivated areas in real time. Given that the Plant Village dataset comprises only 38 classes, this algorithm can be able to tell if a plant is sick or healthy as in many species of plants share some symptoms. In order to improve the performance of classifying real healthy leaves taken in natural settings and a greater diversity of plants and diseases and their photographs, more pictures of plants taken in natural settings can be used. It may be the case that this system will gradually develop a three-layer approach, where the first layer verifies the presence of a plant in an image, the second layer identifies the plant in question, and the last layer checks for the existence of the disease, provides its type if present and accordingly the type of treatment for that particular disease.

REFERENCES

[1] Mumtahana Mou, M. Mahedi Hasan, S. Chakraborty, and M. Abdur Razzak, "Recognition and detection of tea leaf's diseases using support vector machine," in *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, pp. 150–154, IEEE, 2018.

[2] C. Figueiredo, J. G. Barroso, L. G. Pedro, and J. J. C. Scheffer, "Factors affecting secondary metabolite production in plants: volatile components and essential oils," *Flavour and Fragrance Journal*, vol. 23, no. 4, pp. 213–226, 2008.

[3] R. H. A. F. K. Marshall, "Environmental factors affecting plant productivity," in *Fort Keogh Research Symposium*, vol. 1, pp. 27–32, 1988.

[4] M. S. Mustafa, Z. Husin, W. K. Tan, M. F. Mavi, and R. S. M. Farook, "Development of automated hybrid intelligent system for herbs plant classification and early herbs plant disease detection," *Neural Computing and Applications*, pp. 1-23.

[5] J. G. A. Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases," *SpringerPlus*, vol. 2, no. 1, pp. 660, Dec. 2013.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[7] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," *Computers and Electronics in Agriculture*, vol. 72, no. 1, pp. 1–13, 2010.

[8] F. Martinelli, R. Scalenghe, S. Davino, S. Panno, G. Scuderi, P. Ruisi, P. Villa, D. Stroppiana, M. Boschetti, L. R. Goulart, C. E. Davis, and A. M. Dandekar, "Advanced methods of plant disease detection: a review," *Agronomy for Sustainable Development*, vol. 35, no. 1, pp. 1–25, Jan. 2015.

[9] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311-318, 2018.

[10] S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," in *2015 International Conference on Computing Communication Control and Automation*, Pune, India, pp. 768-771, 2015, doi: 10.1109/ICCUBEA.2015.153.

[11] N. Shelar, S. Shinde, S. Sawant, S. Dhumal, and K. Fakir, "Plant Disease Detection Using CNN," *TM Web of Conferences*, vol. 44, p. 03049, 2022, ICACC-2022.