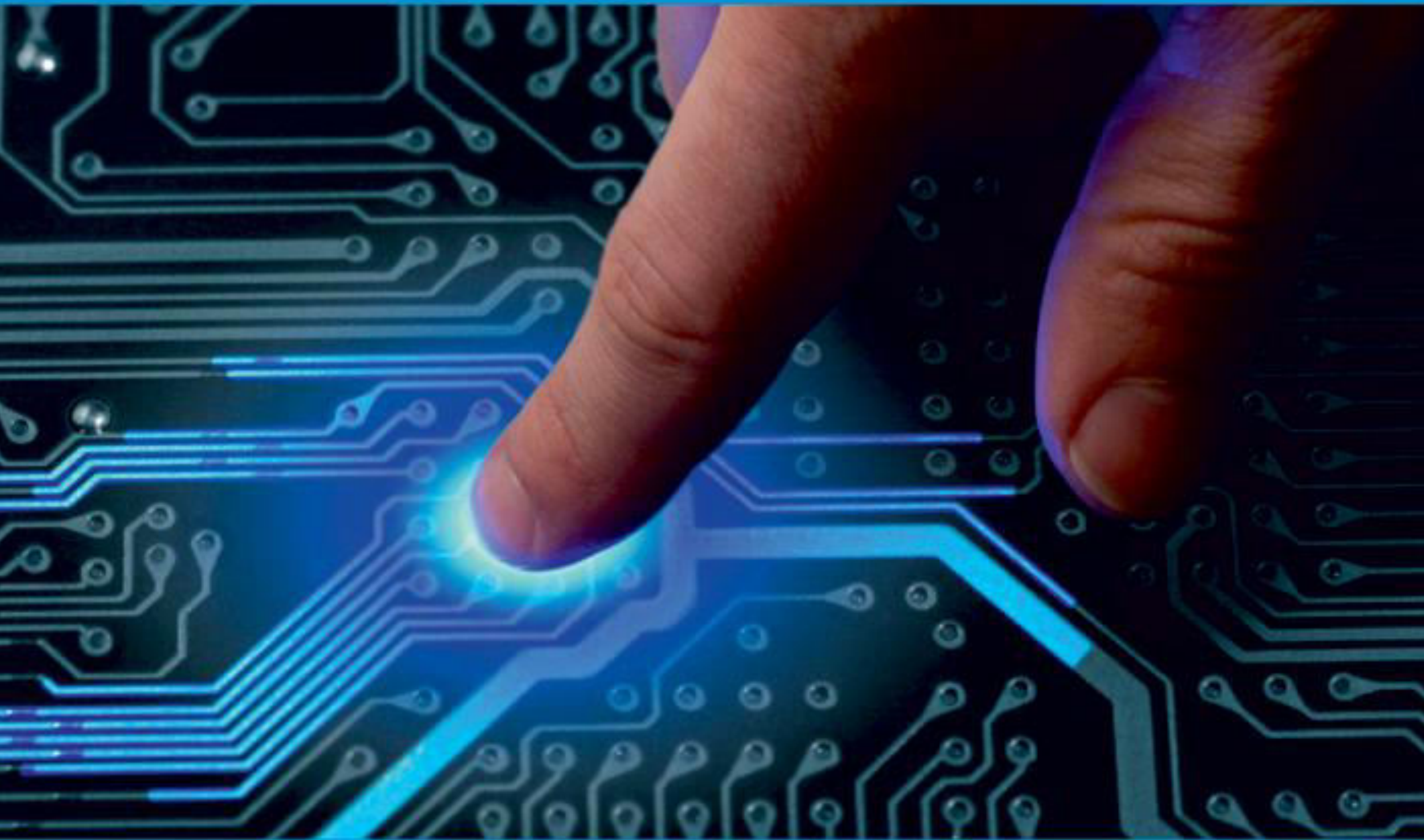
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204363 |

Flutter App for Identification and Remediation of Plant Disease

Hari Charan R, Induja J, Jaganath R, Nitish P

UG Student, Dept. of CSE., KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India

UG Student, Dept. of CSE., KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India

UG Student, Dept. of CSE., KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India

UG Student, Dept. of CSE., KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India

ABSTRACT: The application users can simply upload plant photos to the program, and the model will process them to detect any diseases. Along with suggestions for prevention and treatment, the application provides information on the diagnosed illness. The proposed algorithm has a lot of potential as a useful tool for gardeners and farmers to track and identify plant health. Plant disease identification and management become simpler for users when machine learning algorithms are integrated with the flutter. A preprocessed dataset of images of both healthy and sick plants is used to train the suggested model. To improve the model's performance, a variety of training techniques are applied, such as data augmentation, transfer learning, and hyperparameter tuning. The proposed algorithm is integrated with an application that provides an easy-to-use interface for capturing images of plants and getting instantaneous feedback on how healthy they are. With additional advancements, this technology may have a big impact on agriculture, increasing food security and agricultural productivity. The application is built using Flutter which is cross-platform and may run on both iOS and Android devices, which minimizes development expenses and time while optimizing the user base.

KEYWORDS: Plant disease, Deep Learning, Flutter framework, Sequential model

I. INTRODUCTION

As a source of food, fuel, and raw materials for several industries, agriculture is a vital part of the global economy and human civilization. Agriculture and the environment may be significantly impacted if a plant had the disease. They may raise the cost of production, decrease crop yields, and degrade the quality of agricultural products. Plant diseases can, in extreme circumstances, even kill entire plants or harvests, resulting in a shortage of food and financial losses. A new area of study that aims to increase the precision and efficiency of plant disease identification is machine learning for plant disease detection.

Large datasets containing photos of both healthy and unhealthy plants can be used to train machine learning algorithms, which can then be used to accurately categorize and diagnose plant disease. Machine learning has the potential to drastically cut off the time and cost required to diagnose plant diseases, enabling more prompt and efficient crop treatment. Developers can create applications that function flawlessly on both the iOS and Android platforms by utilizing a single code base with Flutter. This lowers the time and effort needed to design and manage. Google's Flutter is a feature-rich cross-platform framework that gives developers a strong toolset for making visually appealing and highly efficient mobile apps. In this introduction, we will go through the idea of utilizing the Flutter framework to create a plant disease detection and remediation application, as well as the benefits it brings to the development process.

II. LITERATURE SURVEY

With the development of mobile app frameworks like flutter and algorithms for machine learning like convolutional neural network models (CNNs), the usage of technology in agriculture has been increasing in recent years. Plant disease detection applications that accurately detect diseases have been developed as a result of these improvements, enabling farmers to reduce losses and take preventative measures. The proposed mobile application consists of two primary components: the first component takes images of the plant leaves, and the second component processes the images to find out whether any plant diseases are present. The mobile device's camera is used in the initial part of the application to take a picture of the plant leaf. The OpenCV library is then used to process the acquired image in order to

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 4, April 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1204363 |

improve its quality and eliminate noise. Using the CNN Architecture, the second part of the mobile application is in charge of detecting plant diseases.

The CNN architecture was created for mobile devices with constrained processing power in mind. The architecture reduces the number of parameters and compute needed for the model by using a sequence of depth-wise separable convolution layers. Using a dataset of 38,000 photos of both healthy and diseased plant leaves, the authors trained the CNN network. Ten distinct plant species and illnesses are included in the dataset. To expand the quantity of the dataset and raise the accuracy of the model, the authors employed a data augmentation technique. The accuracy of the proposed mobile application in identifying plant diseases was 88.93%. All things considered, the suggested mobile application appears to have the potential to make plant disease detection on mobile devices simple and accessible. On mobile devices with constrained processing power, the CNN architecture and the flutter framework enable the deployment of the application in an effective and lightweight manner.

III. METHODOLOGY

A case study analysis and a survey of the literature are combined in this research article's qualitative research strategy. The following objectives are what the technique aims to achieve: Flutter is an open-source machine learning framework that was used to construct and train the models for the plant disease detection app. With the help of an app, farmers may take a picture of a plant, which is then used to identify the plant and determine whether it shows any symptoms of illness. The software was trained using a dataset of pictures of plants and the diseases that are linked to them that was gathered from several sources.

Module 1: Selecting dataset for the model.
Module 2: Building a model suitable model for the dataset.
Module 3: Training the model.
Module 4: Developing an application and a user interface.
Module 5: Deploying the model in the application.

Selecting dataset:

The dataset that we have chosen for our classification-focused model comprises four distinct plant species, each of which can be classified into a total of fifteen classes. The classes in the collection contain labels like "healthy," "multiple disease," and others and cover a variety of plant attributes. Our model has been fine-tuned to accurately categorize fresh instances of the plants, and the dataset is divided into distinct sets of photos for training and evaluation.

Building Model:

For this project, we decided against transfer learning in favor of creating a unique model. After experimenting with different architectures, we decided on a sequential model that extracts information from our photos using a number of convolutional and pooling layers. Our model comprises of two fully connected layers with a dropout layer in between, and four convolutional layers, each followed by a max-pooling layer. Using our dataset, we trained our model, assessed its performance, and made necessary adjustments and optimizations to obtain high accuracy in classifying the various plant traits. Given the distinctive characteristics of our pictures, transfer learning did not produce results that were sufficient, our customized model has shown to be extremely useful for our purposes.

Training model:

Initially, the required libraries were imported, and a dataset containing photos of plants with various illnesses and leaves in good condition was downloaded. A mapping of plant diseases to their respective class labels was also included in the dataset. After preprocessing the data, we constructed a model. Using the ADAM optimizer, the model was trained for 50 epochs, generating 815 batches (the default value). We used early halting with a patience of 3 and a minimum delta of 0.01 to prevent overfitting. Additionally, a model checkpoint was employed to track validation accuracy and store the optimal model for later training. Lastly, we employed the ReduceLROnPlateau callback with three patience levels, a factor of 0.2, and a minimum learning rate of 0.00001. The training and validation data, as well as all of the callbacks, were supplied to the fit method in a list format.

Application development:

The software application uses machine learning algorithm to accurately identify and categorize plant illnesses from input images of their leaves. A variety of platforms, such as desktop, mobile, or web-based apps, can be used to construct the application. Poor user interfaces can cause users to get frustrated and confused, while a well-designed user interface can help an application become more intuitive, user-friendly, and easy to use.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

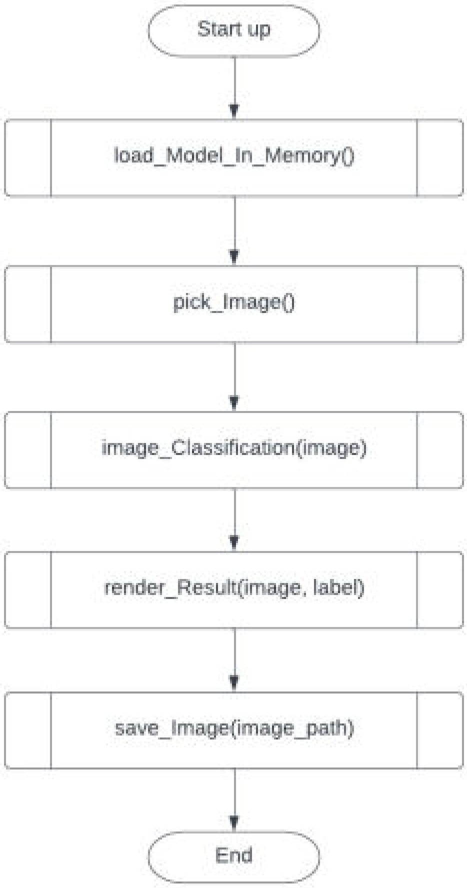
|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204363 |

Deploying model:

A vital component of developing an application is integrating the machine learning model into the application and thoroughly testing it to make sure it works as planned. Users are able to use the application after it has been tested and found to function as expected. To ensure that the application keeps working properly and remains current with machine learning innovations, regular maintenance and updates are required.

IV. APPLICATION WORKFLOW



V. RESULTS

As a result, we imported the dataset and generated multiple models with various neural networks utilizing transfer learning. Table1 summarizes the varying accuracies of these models. The model using the Sequential Model has the highest accuracy among these networks since it aids in maintaining a low error rate considerably deeper in the network. As a result, it has proven to perform exceptionally well in applications requiring deep neural networks, such as feature extraction, semantic segmentation, and various Generative Adversarial Network architectures.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 4, April 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1204363 |

Model	Accuracy (%)
MobileNet	53.72
DenseNet201	68.15
ResNet152	77.76
VGG19	87.24
Sequential API	95.02

Table1. Accuracy Table

The User have to install the developed Flutter application and open the application.

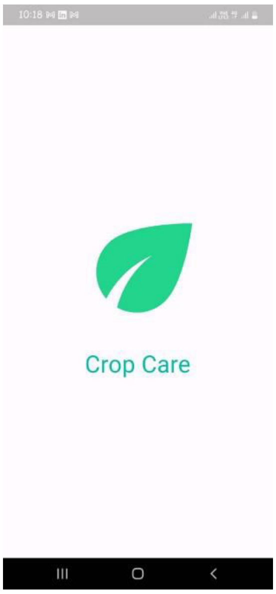


Fig1.Logo of the application

Initially no images is selected at the beginning. To select the images of the affected plant, click either "Pick image from Gallery" or "Pick image from Camera".

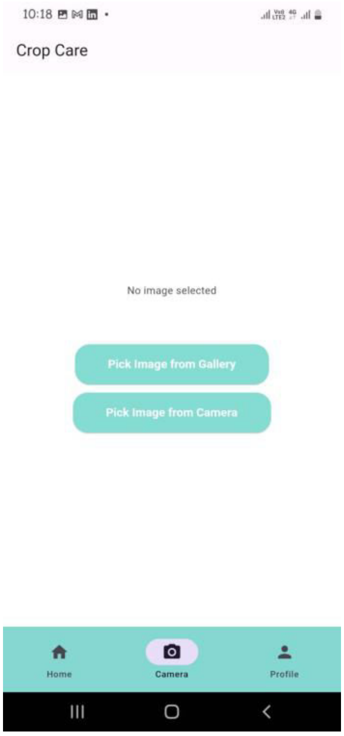


Fig2. Hero page of the application.

After the image is selected the identified name of the plant and the disease name is displayed.

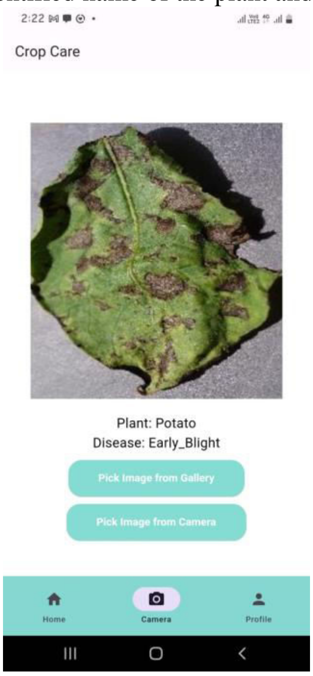


Fig3. Plant disease identification.

The remediations will be the displayed in the home page.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204363 |

Fig4. Symptoms and remediation for the plant disease.

VI. CONCLUSION AND FUTURE WORK

The plant disease detection app developed using flutter and convolutional neural networks is an effective and convenient tool for farmers. With its user-friendly interface and high accuracy rate, the app has the potential to significantly impact the agriculture industry by reducing the spread of plant diseases and minimizing damage. Further work could include expanding the dataset used for training the CNN model and improving the app's performance and accuracy.

Increased accuracy: while CNN algorithms have demonstrated high accuracy in identifying plant diseases, there is still room for improvement. Future work can focus on improving the accuracy of the algorithm by exploring different architectures, optimizing hyperparameters, and using more diverse datasets.

Real-time detection: currently, the app requires users to capture an image and wait for the algorithm to process it. Future work can explore real-time detection using video streams or continuous image capture to provide faster results.

Expansion to more crops: while the initial focus may be on specific crops, future work can expand the app to cover a wider range of crops to address the needs of farmers who grow different types of crops.

Integration with other agricultural technologies: the app can be integrated with other agricultural technologies, such as sensors and weather forecasting, to provide farmers with a comprehensive view of their crops and help them make informed decisions.

Accessibility and localization: future work can focus on making the app more accessible to farmers in different regions and languages. This can be achieved by incorporating localization features, providing translated versions of the app, and ensuring that the app is compatible with low-end devices.

User feedback and engagement: user feedback is crucial for improving the app's accuracy and usability. Future work can focus on engaging with farmers and soliciting feedback to identify areas for improvement and ensure that the app meets their needs

REFERENCES

1. S. Saha, S. Saha, And A. Mandal, "Plant Disease Detection And Classification Using Convolutional Neural Network: A Review," Journal Of Agriculture And Food Technology, Vol. 7, No. 2, Pp. 1-12, 2021.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204363 |

2. B. Yadav And A. Bisht, "Plant Disease Detection Using Convolutional Neural Network," International Journal Of Computer Applications, Vol. 171, No. 9, Pp. 1-5, 2021.
3. A. T. Pham, T. M. Truong, And H. Q. Nguyen, "Development Of A Mobile Application For Plant Disease Diagnosis Using Deep Learning And Flutter Framework," In 2020 Ieee 10th International Conference On Knowledge And Systems Engineering (Kse), Pp. 57-62.
4. S. S. Kiran And N. Y. Bhat, "Deep Learning Based Mobile Application For Plant Disease Detection And Diagnosis," In 2018 Ieee Global Humanitarian Technology Conference (Ghtc), Pp. 1-6.
5. Wang P., Chen K., Yao L., Hu B., Wu X., Zhang J., Et Al. (2016).” Multimodal Classification Of Mild Cognitive Impairment Based On Partialleast Squares”.
6. J. G. A. Barbedo, “Digital Image Processing Techniques For Detecting, Quantifying And Classifying Plant Diseases,” Springer Plus, Vol. 2, No.660, Pp. 1–12, 2013. [8] Caglayan, A., Guclu, O., & Can, A. B. (2013, September).
7. A Plant Recognition Approach Using Shape And Color Features In Leaf Images.” In International Conference On Image Analysis And Processing (Pp. 161-170). Springer, Berlin, Heidelberg.
8. Zhen, X., Wang, Z., Islam, A., Chan, I., Li, S., 2014d. “Direct Estimation Of Cardiac Bi-Ventricular Volumes With Regression Forests.” In: Accepted By Medical Image Com- Puting And ComputerAssisted Intervention–Miccai 2014.
9. S. Yun, W. Xianfeng, Z. Shanwen, And Z. Chuanlei, “Pnn Based Crop Disease Recognition With Leaf Image Features And Meteorological Data,” International Journal Of Agricultural And Biological Engineering, Vol. 8, No. 4, P. 60, 2015.
10. Flutter Documentation: Https://Flutter.Dev/Docs