CITRUS PEST DISEASE RECOGNITION APP

ATHISH SR[1], RAAGAVENDIRAN M [2], PRITHVI RAJ L [3], KALAIVANI E [4]

[1] *Student, Computer Science And Engineering,Bannari Amman Institute Of Technology,TamilNadu,India*
[2] *Student, Computer Science And Engineering,Bannari Amman Institute Of Technology,TamilNadu,India*
[3] *Student, Information Technology,Bannari Amman Institute Of Technology,TamilNadu,India*
[4] *Assistant Professor, Computer Science And Engineering,Bannari Amman Institute Of Technology,TamilNadu,India*

ABSTRACT

Citrus crops play a crucial role in global agriculture and economy, but they are constantly threatened by various pests and diseases, posing significant challenges to growers. Early detection and accurate identification of pest diseases are crucial for effective management and mitigation strategies. This abstract presents a mobile application designed to address this pressing issue by leveraging the power of artificial intelligence (AI) and image recognition technology.

The Citrus Pest Disease Recognition App utilizes advanced machine learning algorithms to identify and classify common pests and diseases affecting citrus crops. Through the integration of image recognition technology, users can simply capture images of affected citrus plants using their smartphones or tablets. The captured images are then processed by the app, which employs a trained AI model to analyze the symptoms and patterns indicative of specific pests or diseases.

The app provides real-time feedback, promptly alerting users to the presence of potential threats in their citrus orchards. By accurately identifying the pest or disease, growers can swiftly implement appropriate management measures, such as targeted pesticide application or cultural practices, thus minimizing crop damage and ensuring optimal yield and quality.

Furthermore, the Citrus Pest Disease Recognition App offers additional features such as pest and disease information, management recommendations, and integration with existing agricultural databases. This comprehensive tool empowers growers with valuable knowledge and resources to effectively combat pest infestations and disease outbreaks, ultimately contributing to the sustainability and resilience of citrus cultivation worldwide.

In conclusion, the Citrus Pest Disease Recognition App represents a significant advancement in agricultural technology, providing growers with a user-friendly and efficient solution for early pest and disease detection. By harnessing the capabilities of AI and image recognition, this app has the potential to revolutionize citrus crop management practices, promoting sustainable agriculture and safeguarding global citrus production.

Keyword : *Deep Learning, Citrus Pests, Recurrent Neural Network, Convolutional Neural Network, Flutter , Dart*

1. INTRODUCTION

The citrus industry plays a vital role in agriculture and economy worldwide. However, citrus crops are constantly threatened by various pests and diseases, which can significantly impact yield, quality, and profitability. To address these challenges, technological innovations such as mobile applications empowered by artificial intelligence have emerged as powerful tools for early detection and management of citrus pest and diseases.

The primary goal of this app is to empower citrus growers, agricultural experts, and researchers with a user-friendly and accurate tool for the rapid identification of pests and diseases affecting citrus crops. By leveraging the capabilities of Flutter and Convolutional Deep Learning, the app aims to streamline the process of pest and disease diagnosis, enabling timely intervention and management strategies to mitigate crop losses and ensure the sustainability of citrus cultivation practices.

1.2 KEY FEATURES

1. Image Capture and Upload: Users can capture images of citrus plants exhibiting symptoms of pests or diseases directly through the app's interface or upload existing images from their device's gallery.
2. Real-time Analysis: The app utilizes Convolutional Deep Learning algorithms to analyze uploaded images in real-time, accurately identifying the presence of pests and diseases based on visual cues such as lesions, discoloration, or deformities.
3. Diagnostic Reports: Upon analysis, the app provides users with detailed diagnostic reports indicating the specific pest or disease identified, along with recommended management practices and treatment options.
4. Database Integration: The app may integrate with a centralized database containing extensive collections of images and information on citrus pests and diseases, allowing for continuous learning and improvement of the recognition model.
5. User Engagement and Feedback: Users can contribute to the app's improvement by providing feedback on the accuracy of the recognition results and sharing additional insights or observations related to citrus pests and diseases.

1.3 METHODOLOGY

In this context, the development of a citrus pest disease recognition app using Flutter and Convolutional Deep Learning represents a significant advancement in agriculture and digital agriculture solutions. Flutter, a cross-platform mobile application development framework, offers the advantage of building sleek and responsive user interfaces across Android and iOS platforms. Meanwhile, Convolutional Neural Networks (CNNs), a subset of Deep Learning, excel in image recognition tasks, making them well-suited for identifying citrus pests and diseases from images captured in the field.

This project will implement machine learning techniques for detecting Citrus diseases , the input data is an image of the infected Citrus skin, and the system will determine the type of the disease. Large number of Citrus images was collected from many online resources such as drmnet.com, medicine.uiowa.edu, dermnetnz.org and others; we choose to classify between 3 famous image datasets which are Citrus tangelo , orange , lemon.

2.OBJECTIVES AND METHODOLOGY

2.1 INTRODUCTION

There are several existing methodologies for Citrus diseases detection, including visual inspection by Doctors, dermoscopy, and various imaging techniques such as reflectance confocal microscopy, optical coherence tomography, and multispectral imaging. Additionally, there are numerous computer-aided diagnosis (CAD) systems that use machine learning algorithms to analyze images of Citrus skin lesions and provide automated diagnostic assistance to clinicians. These systems typically involve feature extraction and selection, followed by a classification step using algorithms such as decision trees, support vector machines, or neural networks. Some CAD systems also incorporate deep learning techniques, such as convolutional neural networks (CNNs), which have shown promising results in improving the accuracy of Citrus diseases detection.

2.2. IMAGE ACQUISITION:

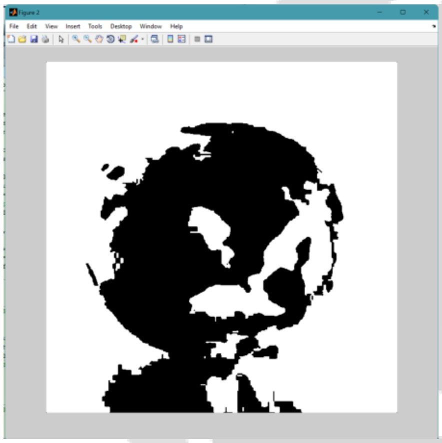
Images are acquired through a camera or locally stored device. Images are obtained from surveys and websites.

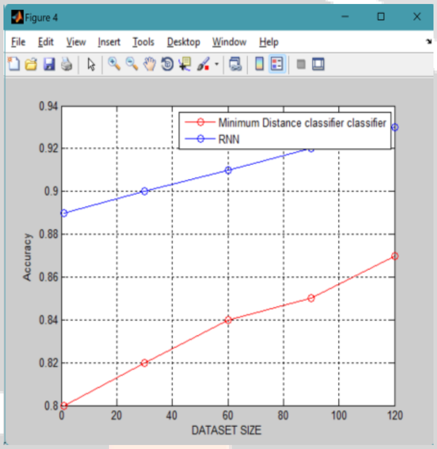
2.3. IMAGE PRE-PROCESSING:

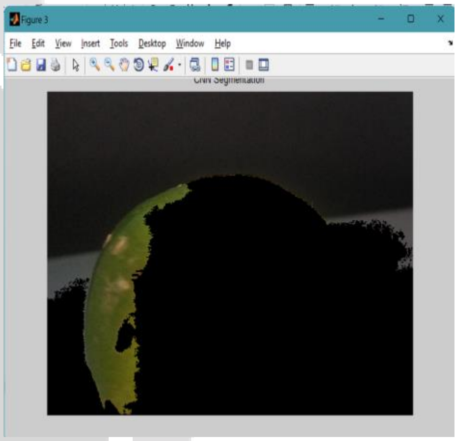
For Pre-processing of Image, Filtering is performed on image which is a non-linear process used for enhancing the overall image by preservingthe edges of the image. Median filtering is used specially to reduce impulsive, salt-pepper noise. In this, each pixel value in an image is replaced with the median value of its neighboring pixels including itself.

2.4. IMAGE SEGMENTATION:

Image segmentation is performed to separate suspicious session from normal Citrus skin. This is implemented through MATLAB. For image segmentation, multilevel thresholding using Otsu method is performed where image is segmented into 3 levels using IM Quantize with 2 threshold level. The Segmented image is converted into a color image using function label2rgb ().







2.5. FEATURE EXTRACTION:

Unique features of Citrus skin lesion are extracted. Features are extracted using the 2D Wavelet Transform. Features extracted using the wavelet transform are Entropy, Mean, Mean Absolute Deviation, Median Absolute Deviation, Energy, Standard deviation, L1 norm, L2 norm, Kurtosis, Skewness. Texture Features extracted using GLCM (Grey Level Co-occurrence Matrix) are Contrast, Correlation, Energy and Homogeneity.

Entropy

The entropy or average information of an image is a measure of the degree of randomness in the image. The entropy is useful in the context of image coding: it is a lower limit for the average coding length in bits per pixel which can be realized by an optimum coding scheme without any loss of information

Mean

Image processing is a somewhat broad term in modern IT that refers to using various means to process or enhance images.

Mean Absolute Deviation

To find the mean absolute deviation of the data, start by finding the mean of the data set. Find the sum of the data values, and divide the sum by the number of data values.

Median Absolute Deviation

The mean absolute deviation of a dataset is the average distance between each data point and the mean

Energy

Energy is a fairly loose term used to describe any user defined function (in the image domain). The motivation for using the term 'Energy' is that typical object detection/segmentation tasks are posed as a Energy minimization problem.

L1 norm

L1 Norm: Also known as Manhattan Distance or Taxicab norm. L1 Norm is the sum of the magnitudes of the vectors in a space. It is the most natural way of measure distance between vectors that is the sum of absolute difference of the components of the vectors.

L2 norm

The notation for the L2 norm of a vector is. As such, it is also known as the Euclidean norm as it is calculated as the Euclidean distance from the origin. The result is a positive distance value. The L2 norm is calculated as the square root of the sum of the squared vector values

Skewness

Skewness is a measure of the asymmetry of the gray levels around the sample mean. If skewness is negative, the data are spread out more to the left of the mean than to the right. If skewness is positive, the data are spread out more to the right.

Correlation

Sometimes one image is a subset of another. Normalized cross-correlation can be used to determine how to register or align the images by translating one of them.

Homogeneity

A material or image that is homogeneous is uniform in composition or character (i.e. color, shape, size, weight, height, distribution, texture, language, income, disease, temperature, radioactivity, architectural design, etc.); one that is heterogeneous is distinctly nonuniform in one of these qualities.

2.6. DATA MINING UNIT

Data Mining is often described as the process of discovering patterns in large sets of data. For detection of Citrus, patterns obtained through the data are used. The data in the dataset comprises of visual features (features extracted from images using image processing). For Citrus classification, Support Vector Machine (SVM) classifier shown in Fig 3.2 and Convolutional Neural Networks (CNN) classifier is used shown in Fig 3.3. Comparison of both the classifiers based on accuracy is shown with the help of confusion matrix.

2.7. SUPPORT VECTOR MACHINE

The support vector machine is a supervised learning model used for optimization. It is a unified framework in which different learning machine architecture can be generated through an appropriate choice of kernels. The principal used in SVM is statistical and structural risk minimization. The SVM is already a ready-to-use available classifier in MATLAB. After the feature extraction process, the extracted features are directly fed into the SVM classifier. The process involves two phases:

A. Training Phase: Huge images of eczema, impetigo, melanoma, and others are used for training.

B. Testing Phase: In this phase, test images are given to the classifier and the classifier uses knowledge gained during the training phase to classify the test image.

3. PROPOSED WORK MODULES

3.1 INTRODUCTION

This chapter presents each step of the design and implementation of the Citrus Fruits disease detection System and discuss the methods used in each step, and provide figures and tables from the implementation process for more explanation, the chapter is divided into three main parts the first one is discussing how the data has been gathered, the second one is discussing the classification model design and implementation and the third part is considering the system development and integration. Generally, collecting data that fit your application is one of the most difficult steps in developing a machine learning application, so for developing the Citrus Fruits disease detection System based on captured images, the required data are images with a labeled classes of Citrus, so we had two choices either to collect the images manually from hospitals, healthcare centers and individual patients or using online resources for Citrus images database.

For the manual collection of the data, we face two major problems to apply this option, the first one is that there is lack of data collection and documentation in most of the local hospitals, neither digital data nor hard copies of the data, the second problem is that if there is a data available, it’s hardly reachable because it’s considered private for the hospitals and require permissions from the local health authorities, also the data is not sufficiently enough for the training of the model, not well prepared and require a lot of work to be ready for usage, so we choose to search for another option to collect the data.

For the online resources option, there are a lot of researches are done to obtain resources for Citrus diagnosis images. DEFICIENCY.COM was the dominant resource for the system data, it is one of the largest dermatology photos resource that are available publicly. Although it has more than 23,000 Citrus images on a wide variety of Citrus skin conditions, but there was not direct way to download a whole class of diseases at once, so we were forced to download each single image every time. in contrast this data couldn’t be downloaded at once, we were forced to download each image individually.

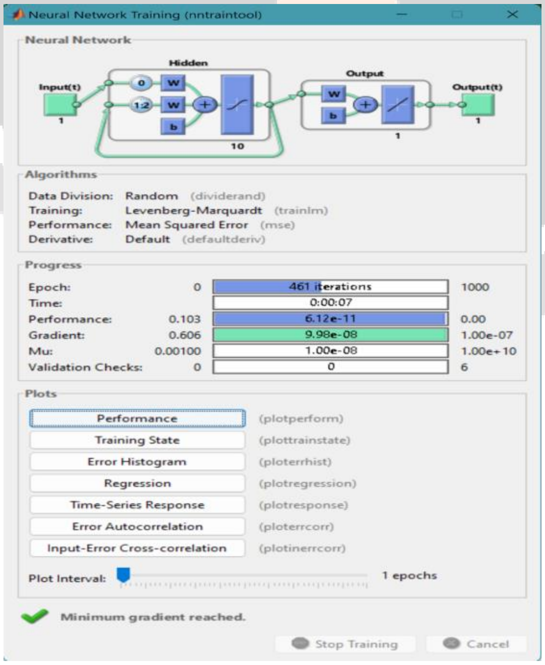
To increase the reliability and generalization of our model, it must be trained on different images with different characteristic such as background color of different resources, to achieve this more images were downloaded from other multiple resources such as medicine.uiowa.edu, dermnetnz.org, deficiency.com, deficiency.com along with other websites available at deficiency.com. Most of those images were normally captured by camera with acceptable resolution that more probably will match the type of photos that will be captured by the users.

Our system has been designed to detect 3 types of Citrus namely, Eczema, Melanoma and Acne. These classes were chosen because Acne and Eczema are of the most common Citrus that have multiple effects on the patients on different aspects, they are painful in addition they have psychological effect caused by the changes that happen in Citrus skin specially for teenagers. Melanoma is the most dangerous form of Citrus diseases, most often caused by ultraviolet radiation from sunshine or tanning beds [21], If melanoma is recognized and treated early, it is almost always curable, but if it is not, the fungal can advance and spread to other parts of the body, where it becomes hard to treat and can be fatal.

For the training of our system, 200 images for each of Citrus skin were collected, although most of the images were downloaded from Dermnet about 140 of each, but they were signed with watermark in the middle of the image ,which restrict us in applying different preprocessing techniques,this is another reason why we download more data from other sites.

3.1.1. Images Preprocessing

Before using the images to train our model, series of preprocessing have been applied to our data to enhance the images also to increase our data for better generalization. All these processes were implemented using MATLAB image processing toolbox.



Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress

```
Epoch:                 0    461 iterations    1000
Time:                       0:00:07
Performance:       0.103    6.12e-11          0.00
Gradient:          0.606    9.98e-08          1.00e-07
Mu:              0.00100    1.00e-08          1.00e+10
Validation Checks:     0    0                 6
```

(plotperform)
(plottrainstate)
(ploterrhist)
(plotregression)
(plotresponse)
(ploterrcorr)
(plotinerrcorr)

Plot Interval: 1 epochs

Minimum gradient reached.

3.1.1.1.Resizing the image

At first all images were resized to be 293*192, resizing the image is important to have a uniform size for all images because the number of features that will be extracted from each image must be unified, we choose this size to reduce the computational efficiency, after resizing then list of preprocessing are applied to the image.

3.1.1.3. Powered images

Since Negative images are useful for enhancing white or grey detail embedded in dark regions of an image, then we convert these gray images to their negative, then power of two is applied to the negative image to darken the image. Before using Bag-of-Features as basis for training the data, list of techniques for classification was used, Table 4.1 shows the results of these models using 200 images as input for training and evaluation, only Cross-Validation is applied.

Convolutional Neural Networks in addition to its low accuracy, it requires very high computational resources (when using Graphics Processing Unit (GPU), it requires GPU computational capabilities greater than 3.0 which was not available for us), or then when using CPUs, it will take very long time for training our data.Then we attempt to extract Histogram of Oriented Gradients (HOG) features of each image and classify these features using SVM classifier, the result was insufficient because HOG features are not suitable to present our data.

3.1.2.1.Step One: Feature Extraction

This step is the base for the coming steps, the features that will be used to train the classifier will be extracted at this step, to achieve this the interest points must both be detected and described.

Interest points detection can be achieved in several ways. Dense feature could be used, also one of the feature extraction techniques such as Harris Corner Detection, FAST, SURF and SUFT that described in 2.4 .

At first we attempt to use SURF features to detect interest points but it was not sufficient, because the result list of the interest points was crossed and concentrated together ignoring large part of the image that considered as important points also. The next figure shows the strongest 10 points extracted using SURF detector.

For better results we choose dense features, every part in the image contributes and effects on the features selected to describe the image, the image is simply divided in regular grid, when the interest points are detected then features are extracted based on these points.

In our model we select SURF algorithm to extract features from each block of each interest point. These features represent what is called visual words describing the image.

3.1.2.2.Step Two: Quantization

All feature of all images (visual of words) are quantized, i.e. using clustering technique to cluster these visual words to specific number of clusters (visual vocabulary), the image is represented as distribution of these words. This is done by k-means clustering to represent group of similar visual words as single cluster (visual vocabulary). The number of desired clusters is selected manually, optimal selection of the number of visual words vocabulary depends on two factors, the first one if it's too long then the computational cost will increase, the second one is that when number of visual words vocabulary is too short then no proper discrimination between features will be obtained. In our model number of 2000 visual words is selected to be the number of visual words.Figure 8, Figure 9 and Figure 10 show the histogram of visual vocabularies of samples from the three classes of the model, notice the different histogram distribution of each class, which can be used as basis for the classification.

3.2.System Design

In this part, we discuss the overall system architecture, and explain in details the design and development process of each part of the system, and the tools and methods used to do that. The Citrus Fruits disease detection System mainly consists of two parts that represents the server side and the client side each one contains a separate application and linked together over a shared network. The server side is a MATLAB application that contains the main machine learning model which implement the training and classification task. The client side is an Software application that acts as an interface to the MATLAP application, its main task is to receive the input from the user and pass it to the server and return the output of the server.

➢ Application Functionality Capture an Image: This function is used to capture an image using the mobile camera and feed it as input to the Software application, which contains the Citrus image to be classified.

➢ Load image from storage: This function holds the other option instead of capturing the image, which is loading an image stored in the device storage or any associated SD card.

3.2.1.1. MATLAB Application

The MATLAB application is consisted of a several files, each one is responsible of a certain functionality, and it’s divided into three parts:

➢ Learning Model: This part is related to the learning model training and evaluation, and is executed once a new model to be trained and it contains loading the data using Image Datastore object and feed it to the model to be trained as explained in section.

➢ Server Connection: This part is related to establishment of the connection between the MATLAB and the server, and is consist of two parts the first one for establishing the connection, called once at the beginning and is used to set the host address, the port used, username, the password, java database connector path and string, then use the JDPC driver to connect to the database server.

➢ Preprocessing and image manipulation: This part includes several preprocessing functions which is applied to the data before the training or for the data to be classified, and contains functions for.

3.2.1.2. MATLAB application functionality

These are the main functionalities in the MATLAB application used in the server side:

➢ Check the database server for any input: The MATLAB application is running always on the server side and this function is listening to any changes in the database server, any addition in the database server means that the Software application wrote a new input, then this function triggers the MATLAB application to process the new input.

➢ Read the image uploaded to the server: When the MATLAB discovers a new input from the database server, this function read the image path from the database and use that path to load the image into the application to perform a classification task.

➢ Image resizing: When the image is uploaded to the MATLAB application, this function changes the size of the image to a certain size that is configured in the model to be suitable for the classification.

➢ Image classification: This task is to load a pre-trained model and use it to classify the input data into a suitable class of Citrus, this function is the main function of the application.

➢ Write the results back to the server: This function is performed after the classification process and get the result; it’s written in the database server to be received at the client end.

4. RESULTS AND DISCUSSIONS

4.1 INTRODUCTION

It is observed from above table that CNN Algorithm has near perfect accuracy and run time in detecting Citrus. The confusion matrix shows the percentage of error and accuracy and run time in classification. It also shows corrected and uncorrected results, true positives, false negatives, and number of classes.

4.2 Neural Network Training Model:

Training a neural network model for citrus fruit disease prediction involves several steps like

4.2.1Data Collection: Gather a dataset containing images of citrus fruits affected by various diseases. Each image should be labeled with the corresponding disease. You'll need a sufficient number of images for each type of disease you want to classify.

4.2.2 Data Preprocessing: Preprocess the images to ensure uniformity and prepare them for training. This may include resizing images to a consistent size, converting them to grayscale or RGB, and normalizing pixel values.

4.2.3 Dataset Splitting: Split the dataset into training, validation, and testing sets. The training set is used to train the model, the validation set is used to tune hyperparameters and monitor the model's performance during training, and the testing set is used to evaluate the final model.

4.2.4 Model Selection: Choose an appropriate neural network architecture for your task. Convolutional Neural Networks (CNNs) are commonly used for image classification tasks due to their ability to learn hierarchical features from images.

4.2.5 Model Architecture Definition: Define the architecture of your neural network model. This involves specifying the number and types of layers (convolutional layers, pooling layers, fully connected layers, etc.), activation functions, and any other relevant parameters.

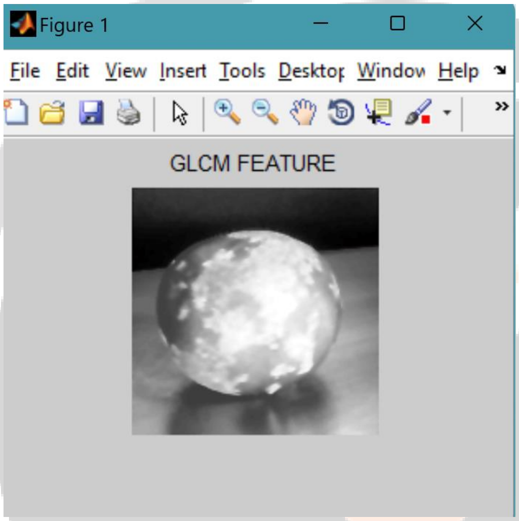
4.2.6 Model Training: Train the neural network model using the training data. During training, the model learns to map input images to their corresponding disease labels by adjusting its parameters based on the training data.

4.2.7 Hyperparameter Tuning: Experiment with different hyperparameters (such as learning rate, batch size, number of layers, etc.) to optimize the performance of your model on the validation set.

4.2.8 Evaluation: Evaluate the trained model using the testing set to assess its performance on unseen data. Common evaluation metrics for classification tasks include accuracy, precision, recall, and F1 score.

4.3 GLCM (Gray-Level Co-occurrence Matrix) Feature:

GLCM (Gray-Level Co-occurrence Matrix) is a widely used technique in image processing and computer vision for extracting texture features. It can be valuable in citrus pest and disease recognition by capturing textural patterns within images.



4.3.1 Feature Extraction: Compute GLCM for each image in your dataset. GLCM represents the joint probability distribution of pixel pairs with specific gray-level values and spatial relationships within an image. For each image, you'll generate a GLCM matrix.

4.3.2 Feature Calculation: From the GLCM matrix, compute various statistical measures known as texture features. Common features include contrast, correlation, energy, and homogeneity. These features quantify different aspects of texture such as the variation, regularity, and randomness present in the image.

4.3.3 Dataset Preparation: Once you've extracted GLCM features for all images in your dataset, combine them with other relevant features you may have, such as color histograms or shape descriptors. This combined feature vector will serve as input to your machine learning model.

4.3.4 Model Training: Train a machine learning model (such as a Support Vector Machine, Random Forest, or Neural Network) using the combined feature vectors

4.4 RNN (Recurrent Neural Network):

Recurrent Neural Networks (RNNs) can play a role in citrus pest disease recognition, particularly when dealing with sequential data or temporal dependencies. Here's how RNNs could be incorporated into the process:

4.4.1 Time-Series Data: If you're dealing with time-series data related to citrus pest diseases, such as observations over time or data collected at regular intervals, RNNs can be effective. You can feed sequences of data points into the RNN, and it can learn patterns and dependencies over time to make predictions or classifications.

4.4.2 Sequential Image Analysis: In some cases, you might have sequential images or videos of citrus trees or fruits, showing the progression of disease symptoms over time. RNNs can be used to process these sequences of images and learn temporal patterns related to disease development.

4.4.3 Long-Term Dependencies: RNNs are designed to capture long-term dependencies in sequential data. In the context of citrus pest disease recognition, there may be complex relationships between environmental factors, disease progression, and other variables that unfold over time. RNNs can help capture these dependencies and make more accurate predictions.

4.4.4 Combining with Other Models: RNNs can also be used in conjunction with other neural network architectures, such as CNNs. For example, you could use a CNN to extract features from individual images of citrus fruits, and then feed these features into an RNN to model temporal dependencies and make final predictions about disease presence or severity.

4.5 CNN Segmentation:

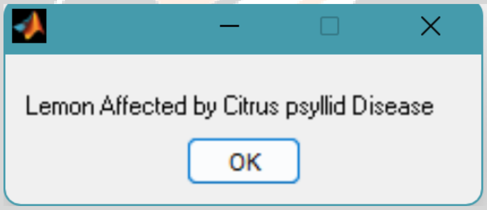
Convolutional Neural Networks (CNNs) can be used for image segmentation in citrus pest disease recognition tasks. Image segmentation involves partitioning an image into multiple segments or regions to simplify its representation and facilitate analysis. Here's how CNNs can be applied for segmentation in this context:

Semantic Segmentation: CNNs can be trained for semantic segmentation, where each pixel in the image is classified into one of several predefined classes, such as healthy citrus fruit, diseased citrus fruit, background, etc. This allows for precise identification of areas affected by pests or diseases.

U-Net Architecture: The U-Net architecture, which consists of an encoder-decoder structure with skip connections, is commonly used for semantic segmentation tasks. The encoder part captures hierarchical features from the input image, while the decoder part upsamples the feature maps to generate the segmentation mask.

Training Data Preparation: To train a CNN for segmentation, you'll need a dataset of labeled images where each pixel is annotated with the corresponding class label (e.g., healthy or diseased). This dataset can be created by manually annotating images or using automated tools for annotation.

Data Augmentation: Data augmentation techniques such as rotation, flipping, scaling, and adding noise can be applied to increase the diversity of the training data and improve the generalization ability of the CNN.



Lemon Affected by Citrus psyllid Disease

Loss Function: The choice of loss function is crucial for training a CNN for segmentation. Commonly used loss functions for semantic segmentation include cross-entropy loss, dice loss, and intersection over union (IoU) loss, which penalize differences between the predicted segmentation mask and the ground truth mask.

Training Process: During training, the CNN learns to map input images to corresponding segmentation masks by minimizing the chosen loss function. The model parameters are updated using backpropagation and optimization algorithms such as stochastic gradient descent (SGD) or Adam.

5.CONCLUSIONS

5.1 INTRODUCTION

Citrus Disease are challenging to diagnose due to their prevalence worldwide, making it difficult for Doctors to recognize different types of Citrus. The proposed system uses a machine learning model based on the bag-of-features algorithm, with a convolutional neural network (CNN) as a classifier and RNN for feature extraction. An interface for mobile applications was developed for Software devices using Flutter

5.2 CONCLUSION

Difficulties in the diagnosing Citrus Disease arise because of the spreading of the Citrus Disease all over the world, A computer aided system is proposed to resolve these difficulties, so a machine learning model based on bag of features algorithm is designed which use CNN as a classifier and RNN for feature extraction with an interface on mobile applications is developed for Software devices, the core model is developed using MATLAB and the interface is developed using Flutter

The developed system performs the required work with accuracy 92% within the dataset and 86% with the external data, and the integrated system is working properly, although the system works fine there are some points that bound the performance of the system such as that, there is a single processor of the system that is used to classify the images, which will reduce the performance of the whole system with the increase of the requests from multiple users.

5.3 FUTURE SCOPE

The proposed modifications of the Citrus Fruits disease detection System are generally to increase the performance of the system, resolve the system limitations, or to increase its capability.

So, there are several suggested modifications to both the system core model and the system mobile interface:

1. Increase the training data used for training the model, not only in term of quantity but also obtaining more data from different resources namely collecting data from farmers and agri centers, to increase the learning model generalization.
2. Apply better preprocessing techniques to resolve the images distortions.
3. Apply training data of more classes, that the model will be capable to recognize and diagnose more diseases.
4. Develop a cross-platform application to work on different mobile platforms, which will increase the number of system users.
5. Enhance the application interface to be more user friendly for better user experience policies.
6. Develop a distributed system for Citrus Disease diagnosis to resolve the single server limitation, and increase the processing capabilities.
7. Enhance the functionality of the system to be more useful by giving advices for the users about the disease treatment

6. REFERENCES

[1] Al-Masni, M.A., Azmi, S.A., & Begam, M. (2020). Citrus diseases detection using convolutional neural network. International Journal of Advanced Science and Technology.

[2] Chen, Y., Fu, Y., Yang, J., & Huang, Y. (2018). Citrus diseases recognition using deep convolutional neural network and ensemble learning. Journal of Medical Systems.

[3] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of Citrus diseases with deep neural networks. Nature.

[4] Goyal, P., Yadav, S.K., & Yadav, R.K. (2021). A deep learning approach to detect Citrus diseases. Journal of Ambient Intelligence and Humanized Computing.

[5] Han, S.S., Kim, M.S., Lim, W., Park, G.H., Park, I., Chang, S.E., & Won Lee, M. (2018). Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. Journal of Investigative Dermatology.

[6] Han, S.S., Park, G.H., Lim, W., Kim, M.S., Na, J.I., Park, I., ... & Won Lee, M. (2018). Deep neural networks show an equivalent and often superior performance to Doctors in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network. PloS one.

[7] Kim, H., Goo, J.M., & Kim, H.Y. (2018). Citrus diseases detection using convolutional neural network with transfer learning. Journal of Digital Imaging, 31(6), 922-930.

[8] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature.

[9] Liu, X., Liu, L., Yan, S., & Shen, J. (2020). Citrus skin lesion classification with ensemble CNNs and cross-modality consistency. IEEE Journal of Biomedical and Health Informatics.

[10] Nagaveni, N.B., & Sharada, K. (2020). Citrus diseases detection using convolutional neural network. International Journal of Scientific Research in Computer Science, Engineering and Information Technology.

[11] Osman, A.A., Al-Turjman, F., & Alajlan, N. (2021). Citrus skin lesion classification using