International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Design of Domestic Plants Leaves Disease Detection Using Deep Learning

R. Augustian Isaac[1], P. Sundaravadivel[2*], D. Yuvaraj[3], P. Hemavathy[4], P. Janaki Ramal[5], P. Sankar[6]

Submitted: 15/01/2024 **Revised**: 23/02/2024 **Accepted**: 01/03/2024

Abstract: The proposed system presents a deep learning-based solution for detecting and classifying leaf diseases in domestic plants. The proposed system employs a convolutional neural network (CNN) to automatically extract features from leaf images and classify them into different disease categories. The dataset used in the study consists of images of healthy leaves and leaves affected by bacterial spots, early blight, and late blight. Using the dataset, the CNN algorithm is trained to identify the characteristics and trends that distinguish healthy leaves from diseased leaves. Next, new photos of leaves are classified using the template as either healthy or deteriorating, and if a leaf is deteriorating, the exact disease is identified. Experiments carried out to assess the suggested approach demonstrate that it provides a high degree of accuracy in terms of both leaf disease detection and classification. The application is designed to be user-friendly and easy to use, as it is implemented as a mobile application that can be installed on a smart phone or tablet. The user can take a picture of a leaf using the camera of the device, and the application automatically processes the image and displays the name of the disease, as well as methods to cure the disease. This feature can help gardeners and farmers identify the disease and take the necessary action to prevent the illness from spreading.

Keywords: *Edge Computing, Intelligent Shopping Cart, Internet of Things, Long Short-Term Memory (LSTM), Optimization, Reinforcement Learning, Smart Shopping*

1. Introduction

Plant diseases can significantly impair crop quality and output, causing financial losses for farmers and compromising food security. For prompt intervention and successful disease management, early detection and accurate diagnosis of plant diseases are essential. Plant disease detection has always been done by visual inspection, which can be labor-intensive and error prone. Additionally, not all farmers possess the knowledge required to correctly identify illnesses. Utilizing machine learning methods to identify plant diseases has gained popularity in recent years. Because it can automatically learn complicated features from huge datasets, deep learning has demonstrated promising outcomes.

[1]*Saveetha Engineering College, Thandalam, Chennai, India, 602 105.*
Email: sangam.naadu@gmail.com
ORCID: 0009-0004-6479-6885
[2]*Saveetha Engineering College, Thandalam, Chennai, India, 602 105.*
Email: sundar.me2009@gmail.com
ORCID: 0009-0003-1115-9490
**(Corresponding Author)*
[3]*Department of Computer Science,*
Cihan University-Duhok, Duhok, Iraq.
Email: d.yuvaraj@duhokcihan.edu.krd
ORCID: 0000-0002-2745-8092
[4]*Saveetha Engineering College, Thandalam, Chennai, India, 602 105.*
Email: hemavathyp@saveetha.ac.in
ORCID: 0000-0002-5781-0616
[5]*Saveetha Engineering College, Thandalam, Chennai, India, 602 105.*
Email: janakiramal424@gmail.com
ORCID: 0009-0006-9954-8296
[6]*Saveetha Engineering College, Thandalam, Chennai, India, 602 105.*
Email: sankarpalanisamy1983@gmail.com
ORCID: 0009-0008-0986-4493

1.1. Image processing

Image manipulation is the procedure of applying various adjustments to an image to improve it or extract relevant data. The output of signal processing on an image could be another image or a characteristic or set of features. Image processing is currently the technology that is evolving the fastest. It is also very important that engineering and computer science research fields. The three stages that make up the image processing process are as follows:

- To import the picture using tools for image capture.
- Examining and modifying the image.
- Products that either feature altered images or analyses of reports based on images.

The two types of image processing techniques used

- Analog
- Digital

1.1.1 Analog image processing

The alteration of images referred to as analogue image manipulation that is represented in continuous physical quantities, such as light intensity or voltage, without converting them into discrete digital values. Analog image processing techniques were commonly used in the past when digital computers were not yet available, and they are still used in some specialized applications where the conversion to digital format is not practical or desirable. This technique involves the use of various analog devices, such as filters, amplifiers, and analog-to-digital converters, to modify or analyses the properties of an image. For example, an analog filter can be used to remove noise from an image by selectively attenuating certain frequency components, while preserving others. Analog amplifiers can be used to enhance the contrast or brightness of an image by boosting the signal levels. Analog-to-digital converters can be used to convert analog images into digital format for further processing or storage.

1.1.2 Digital image processing

Digital image processing on the other hand, refers to the manipulation of images that are represented in discrete digital values, such as pixels. Digital image processing techniques are based on mathematical algorithms that operate on the digital representation of the image, and they are extensively employed in contemporary image processing uses, including digital photography, medical imaging, and machine vision.

Digital image processing techniques involve the use of various algorithms, such as filtering, enhancement, segmentation, and compression, to modify or analyze the properties of an image. For example, a digital filter can be used to remove noise from an image by applying a convolution operation to the pixel values. Digital enhancement techniques can be used to manipulate pixel values using mathematical equations to enhance an image's contrast or clarity.

Digital segmentation algorithms can be used to separate different objects or region in an image based on their properties, such as color or texture. Digital compression algorithms can be used to reduce the size of an image by removing redundant or irrelevant information while preserving the essential features.

1.2. Image Enhancement

Image enhancement is the process of applying different algorithms or approaches to an image in order to improve its visual quality. In deeper learning, convolution neural networks (CNNs), a kind of neural network frequently employed for processing image tasks, are frequently utilized to enhance images. CNNs are made to apply a number of filters to the input image in order to extract significant features from it. These filters pick up on features like corners, borders, and textures that are trends in the image information. By stacking multiple layers of these filters, a CNN can learn increasingly complex representations of the input image. One common approach to image enhancement using CNNs is to use a type of neural network called an auto encoder. An encoder network plus a decoder network makes up an auto coder. The encoder network into a lower-dimensional participation compresses the input picture, and the decoder network to reconstitute the original picture then uses the shortened version. By minimizing the reconstruction error through auto encoder retraining, it learns to remove noise, artifacts, and other imperfections from the input image.

Another approach to image enhancement in deep learning is to use a type of neural network called a generative adversarial network. The discriminate network and the generator network are the two networks that make up a GAN. The generator network generates new images that are comparable to the training data while the classification network tries to distinguish between the generated images and the real ones [36]. The generator network gains the ability to produce realistic and aesthetically beautiful images by training both networks in an atmosphere of competition.

1.3. Common Factors that affect in Leaves

Pest infestations: Insects and mites can feed on plant leaves, causing damage such as discoloration, holes, and distortion.

Fig. 1. Vegetable health

Fig. 2. Vegetable early blight

Fig. 3. Potato late blind

Fig. 4. Tomato healthy

Fig. 5. Tomato early blight

Fig. 6. Tomato late blight

Fungal diseases: Fungal infections can cause discoloration, wilting, and decay of leaves, as well as the growth of mold or mildew on the surface. Bacterial diseases: There are several signs associated with bacterial infections, including as areas, tumors, and leaf wilting. Viral diseases: A variety of indicators, including as flecks, yellowing, and deformation of leaves, can be attributed to viruses. Nutrient deficiencies: Lack of essential nutrients, such as nitrogen, phosphorus, and potassium, can cause leaves to turn yellow, brown, or develop dead spots. Environmental stress: Extreme temperatures, drought, or exposure to chemicals can cause leaves to wilt, turn brown or yellow, or develop spots. Physical damage: Mechanical damage such as tearing or cutting of leaves can cause them to wilt or develop brown edges.

2. Related Works

Martinelli, Federico, et al (2015). "Advanced methods of plant disease detection. A review." Agronomy for Sustainable Development 35.1. The introduction of the study discusses the significance of plant disease detection and how it affects agricultural productivity and food security. The conventional procedures for finding plant diseases are then discussed, including visual inspection, laboratory analysis, and remote sensing technologies. The authors then go over the most recent developments in plant disease detection technology, including imaging-based methods like hyperspectral imaging, thermal imaging, and fluorescence imaging, as well as molecular biology methods like PCR and next-generation sequencing (NGS). The potential for and difficulties in implementing the use of unmanned aerial vehicles or drones for plant disease detection are also covered in the article. The authors emphasize the significance of data integration and the requirement for creating integrated plant disease monitoring systems that integrate various detection methods.

P. Erentinos (2018), published in the journal Computers and Electronics in Agriculture in , provides a comprehensive review of the use of deep learning replicas for plant disease detection and diagnosis The relevance of plant disease identification in agriculture is covered in the opening section of the study, along with the drawbacks of more conventional techniques like human specialists' eye examination. The fundamentals of deep learning models—such as convolutional network models, which are frequently employed for image identification tasks—are next covered. The study examines a number of investigations that have employed deep learning models to identify plant diseases, including those that focus on detecting specific diseases such as potato late blight, apple scab, and tomato diseases. The studies generally involve training CNNs on large datasets of labeled images of healthy and diseased plants, and then evaluating the models' accuracy in detecting and diagnosing diseases.

Suruchi Jindal and Maninder Kaur (2018) proposed application base support of Wild Species for Wheat Improvement Using Genomic Approaches, in terms of area, productivity, and dietary value, wheat is the most prominent food crop in the world. This could grow somewhere around a few yards to higher than 3800m above sea level, across both tropical and temperate regions. Amidst being capable of the broad range of climatic adaptability, its yield stability would be restricted by an assortment of biotic and abiotic stresses. These primary, secondary, and tertiary gene streams within wild wheat hold untapped variances for wheat enhancement, such as tolerance to biotic and abiotic stressors that could be translated to cultivated wheat. So even though Tritium and Aegilops species, which share a genome in common with wheat, can transfer genetic variability by small alterations to the experiment.

Mamoon Rashid, Balwant Ram, and Ranbir Singh Batth (2019) using Python OpenCV, a Novel Image Processing Method for Feature Identification of Wheat Crops, with the goal of removing undesirable foliage from leaf pictures. The authors devised and employed OpenCV, an image processing technique, with the goal of separating the unhealthy portion of the overall from the leaf photographs. Wheat images undergo a series of filtering techniques, including color filtering, edge recognition, background extraction, and an amalgamation of edge detection and color filtering. This trimming approach can aid in the identification of illnesses in wheat plants. The module utilizes users to capture plant imagery in an organized sequence so that risks may be swiftly appraised. Besides helping humans in detecting plant issues, with the outcomes of this study, the detection process may indeed be done by machines without the need for human resources to supply the model with the dataset, which is a considerably higher reliable technique. In terms, this research will contribute to streamlining farming activities more swiftly and permit farmers to cultivate more territory in less time frames.

Mohanty, Sharada P., David P. Hughes, and Marcel Salathe (2016)."Using deep learning for image-based plant disease detection." Frontiers in plant science. The authors discussed the relevance of plant disease identification in agriculture is covered in the first section of the study, along with the possible benefits of automated detection systems for raising crop yields. Next, a deep learning model based on convolutional neural networks is demonstrated for the identification and diagnosis of plant illnesses from pictures. The authors used a sizable dataset of labelled photos of various plants and diseases, such as tomato yellow leaf curl virus, potato late blight, and cassava brown stripe disease, to train their model. The model's performance was then assessed using a different test dataset, and its results were contrasted with those of other machine learning techniques like support vector machines and random forests.

3. Proposed Work

A subset of deeper learning is also called machine learning. It functions similarly but has different properties, and it is classified as machine learning in theory. Machine learning models are more advanced than deep learning models, but they still need outside assistance to continue improving. When an automated learning framework predicts something incorrectly, programmers need to explicitly handle that. However, the model accomplishes that in deep learning. A self-driving automobile system is a good illustration of deep learning.
What happens if the user says, "The light is so faint I can't see anything," instead of saying, "It's dark," when they wish to switch on the torch? In this case, the machine learning model analyses different sentences people say, looking for the word "dark" and turning on a flashlight when it appears. In this respect, deep learning differs from machine learning. A deep learning model that can learn from its computational process uses Flashlight.

3.1 Reasons for choosing deep learning

Reasons for choosing deep learning: A subset of machine learning, deep learning uses algorithms to analyze data, create abstractions, and mimic thought processes. Deep learning (DL) uses multiple layers of algorithms to analyze data, understand spoken language, and visually identify elements. Each layer sends information to the layer below it, and the output of one layer serves as the input of another layer. The input and output layers are the first and last layers of the network. All layers between the two are called hidden layers. Each layer is usually a simple algorithm with a single type of activation function.
Another part of deep learning is featuring extraction. Feature extraction uses algorithms to automatically generate features of data that are useful for training and understanding. A data scientist or programmer is usually responsible for feature extraction. When deep learning was first made famous in 1943, When Walter Pitts and Warren McCulloch developed a computer model; they used the neural networks found in the human brain together with a collection of principles and techniques known as "thresholds logic" to simulate mental activities. Since then, he has only encountered two hurdles in the growth of deep learning. Both have something to do with the infamous AI Winter.

3.2 Working

We must first comprehend the issue at the outcomes of the finest solution. It also determines whether using deep learning makes sense from a practical standpoint. Relevant facts are found and addressed appropriately in the second step. Thirdly, pick a suitable deep learning method. Fourth, while training the data set, we must employ an algorithm. Fifth, the data set needs to be finalized for analysis.

3.3 Virtual assistants

Forbes reports that MIT is creating a new technology enabling autonomous cars to drive without a map because 3-D mapping is still only available in high-traffic parts of the world and is less effective at 24 preventing accidents. The reason this kind of "map-less" approach hasn't been explored before Fig 4.10 Virtual Assistants They learn to understand spoken language to learn how to obey your commands. Moreover, virtual assistants may take your notes, transform your speech into text, and set up appointments. Virtual assistants can do everything from running errands to automatically replying to your calls to organizing duties between you and your team members, so they are practically at your beck and call. Deep learning apps, such as document summaries and text generation, enable artificial intelligence to assist you in producing or delivering suitable email copy.

3.4 Visual benefactor

Take a moment to envisage yourself scrolling through a myriad of evocative old photos. While you would prefer to sort them first, you resolve that you will have some of them framed. In the lack of metadata, menial work was the only viable option. The most you could do was order them chronologically, but now and then downloaded shots lack the necessary metadata. Deep learning has made it possible to film based on things from events, individuals, locales, or dates that may be shown on it. To locate an individual photo inside a library, modern visual recognition methods with hidden stages from basic to expert are essential 25 (let's presume a dataset as vast as Google's image library). Humongous shot This same execution of visual identification utilizing deep neural networks, accelerating advancements in digital media management, uses convolutional neural networks, Tensor flow, and Python.

3.5 Convolution Neural Networks

Bionic convolutional neural networks have been put forward as an answer to this issue, by reducing the amount of factors and specifically tailoring the structured cabling to operations using visuals. Convolutional neural networks are typically has multiple capacities that may be generally located on certain functionality.

3.6 Architecture

A. Convolution Layer

The "dot products" between the weights and the inputs are "integrated" across “channels" in a 2D convolution process. Shared filter weights arise amongst receptive fields. The total number of filtering and the input volume streams are the same "depth" for the output volumes and filter, respectively.

B. Activation Layer

Used to augment network non-linearity without impacting the receptive fields of convolution layers. Leaky ReLUtackles the vanishing gradient dilemma. Prefer ReLU, which contributes toward rapid training. It yields a discrete probability distribution vector and is a specific form of activation layer that is often near the end of the FC layer outputs. It forms a discrete probability distribution vector and belongs to a narrow definition of the activation layer which frequently appears at the end of the FC layeroutputs.

$$P(y=j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^{\mathsf{T}}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^{\mathsf{T}}\mathbf{w}_k}}$$

Given sample vector input x and weight vectors{w},the predicted probability of y=j

C. Pooling Layer:

Convolutional layers provide activation maps. The pooling layer applies nonlinear down-sampling on activation patterns. The inclination is to deploy lesser filter sizes and surrender pooling; pooling is aggressive (discard info).

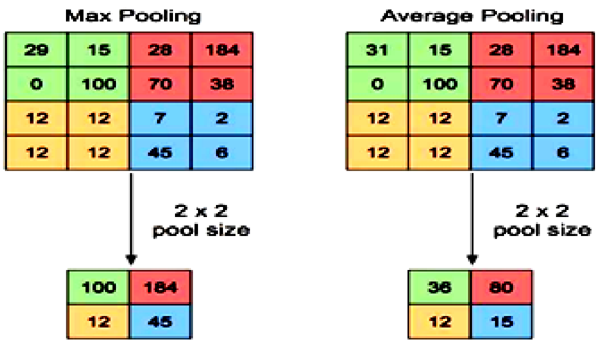

Fig. 7. Pooling Layer

D. FC Coating

- The last stage of learning, known as the regular neural network, links extracted visual information to intended outputs.
- Generally receptive to coding and task classification.
- A vector that is commonly produced is run through SoftMax to indicate the classification accuracy.
- Another usage for the outputs is as "obstacles."

3.7 Advantages Of Using The Algorithm

- CNN harvests significant information.
- High Correctness
- CNN Business Process Velocity.
- Most Suitable for Large data sets

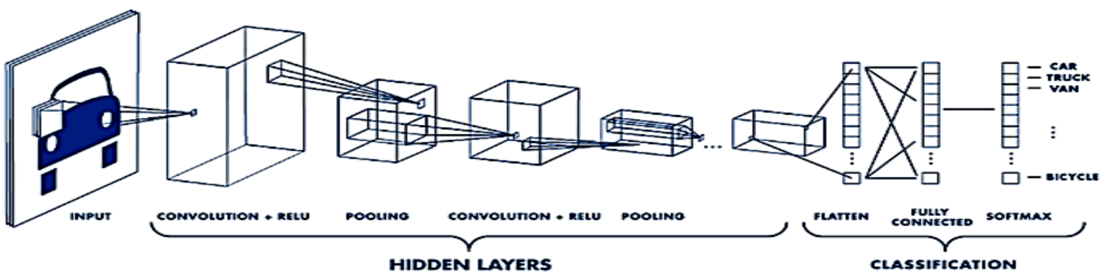

Fig. 8. System Design

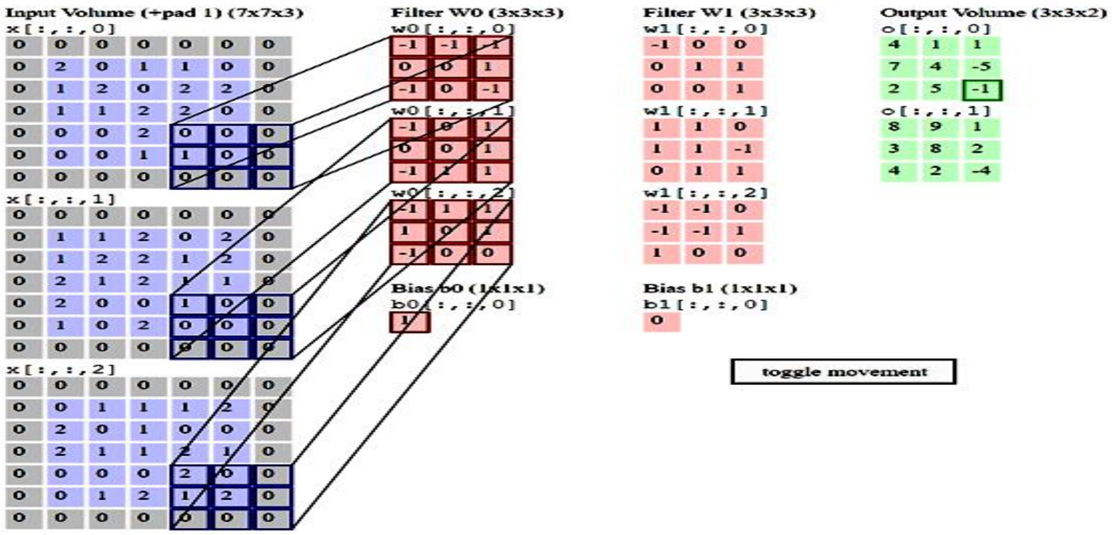

Fig. 9. Convolution Layer

3.8 Block Diagram

3.8.1 Training

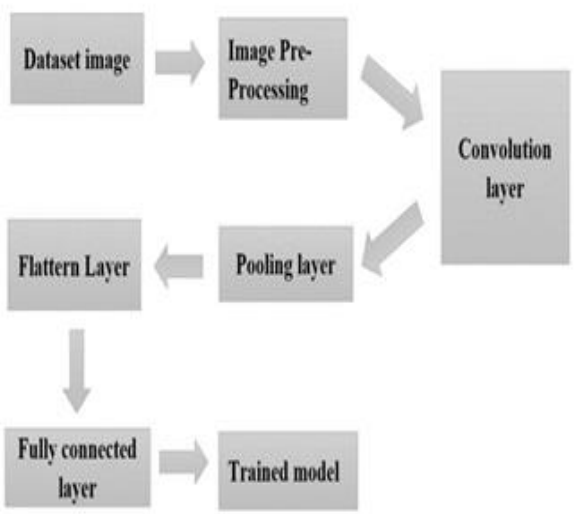

Fig. 10. Block diagram for Training

- **Dataset Image:** An image collection that will be utilized to train the CNN is the first step in the training procedure. This dataset may consist of images of normal and diseased leaves or any other type of image that the CNN is being trained to classify.
- **Image Preprocessing:** The photos are preprocessed to make sure they are in a format that is appropriate for the subject's face before they are utilized by the CNN. This could entail doing things like reducing, normalization, and data augmentation.
- **Convolutional Layer:** The preprocessed images are then passed through one or more convolutional layers. These layers perform operations known as convolutions, which detect and extract features from the input images. Each convolutional layer consists of a set of learnable filters that are trained to detect different types of features in the input images.
- **Pooling Layer:** A pooling layer or layers may be applied after the convolutional layer's output has been processed. By decreasing its convolutional layer's spatial accuracy, these layers reduce the output while keeping the salient characteristics. By doing so, over fitting is avoided, and the model's parameter count is decreased.
- **Flattening Layer:** After that, a 1D vector is created by flattening the combined layer's input. This vector represents the features that have been extracted from the input image.
- **Fully Connected Layer:** The flattened vector is then passed through one or more fully connected layers. By assigning the collected characteristics to the proper output class, these layers classify the data. The model can identify intricate patterns in the input data because every neuron in the fully connected layer is connected to every other neuron in the layer above.
- **Training Model:** Optimizing the model's parameters to reduce the error of the true and projected labels is the last stage of the training process. Usually, an optimization approach like random gradient descent is used for this. During the training process, the weights of the filters in the convolutional layers and the neurons in the fully connected layers are adjusted to improve the accuracy of the model's predictions on the training dataset.

3.8.2 Testing

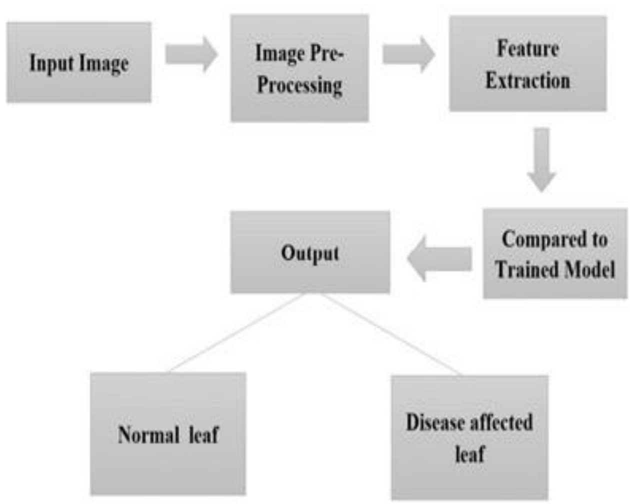

Fig. 11. Block diagram for Testing

4. System Architecture

- **Data Storage:** The first component of the system architecture is a database for storing the pictures that CNN will be trained and tested with. This database may be hosted on a cloud-based platform, or on a dedicated server.
- **Image Preprocessing:** Before the images can be used to train or test the CNN, to make sure that they are in an arrangement that is appropriate for the design, they must undergo preprocessing. Work like resizing might be involved in this, normalization, and data augmentation.
- **CNN Architecture:** The next component of the system architecture is the CNN itself. The CNN may consist of multiple layers for feature extraction, classification, and detection. The particular needs of the leaf categorization and detection task should guide the CNN's architectural selection.

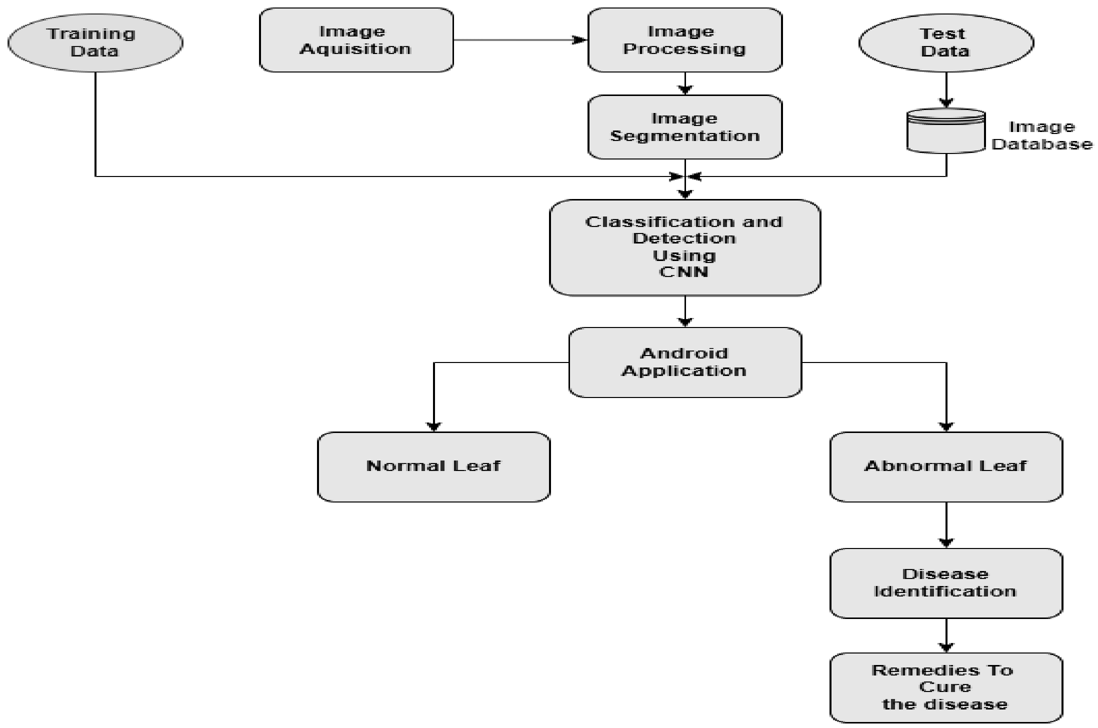

Fig. 12. System Architecture

- **Training:** Once the CNN architecture has been defined, The preprocessed photos in the training dataset can be used to train the model. During the training phase, the weights of the links between the network's cells are optimized according to the discrepancy between the true and anticipated labels.
- **Testing:** After the model has been trained, it can be tested on preprocessed images in the test dataset. The testing process involves evaluating the accuracy of the model's predictions.
- **Mobile Application:** The final component of the system architecture is a mobile application that allows users to input images of leaves and receive information about whether the leaf is normal or diseased, along with recommended treatments. The mobile application may communicate with the CNN via an API.
- **Integration:** The CNN and the mobile application must be integrated, such that the outputs of the CNN can be presented to the user via the mobile application.
- **Remedies and Recommendations:** If the CNN predicts that a leaf is diseased, the mobile application may provide the user with recommendations for how to cure the disease, based on a database of known treatments. The recommendations may be personalized based on factors such as the type of plant and the severity of the disease.

4.1 MODULES

- ✓ Model building and fitting
- ✓ Conversion of the model into TFLite
- ✓ Integrating the TFLite model into an application

4.1.1 Model building and fitting

This Python code defines a convolutional neural network (CNN) model using the Keras library. The model architecture consists of multiple layers, each performing a specific operation on the input data. Here is a brief explanation of each layer:

Conv2D: This convolutional layer applies a set of learnable filters to the input image to extract features. The first layer has 32 filters with a kernel size of 3x3.

Activation: This layer applies a non-linear activation function to the output of the previous layer. In this case, the Rectified Linear Unit (ReLU) activation function is used.

Batch Normalization: This layer normalizes the output of the previous layer to speed up training and improve generalization.

MaxPooling2D: This layer performs max pooling on the output of the previous layer, reducing the spatial size of the data while retaining the most important information.

Dropout: This layer randomly drops some neurons during training to prevent overfitting.

Flatten: This layer flattens the output of the previous layer into a 1D vector.

Dense: This layer is a fully connected layer that performs a linear transformation on the input. The first dense layer has 1024 neurons.

SoftMax: This layer applies the SoftMax function to the output of the previous layer to produce a probability distribution over the output classes.

4.1.2. Conversion of the Model into TFLite

tf.keras.models.load_model('/content/drive/MyDrive/Final year project /plant_disease.h5'): This loads the trained Keras model for plant disease classification from the specified file path *(/content/drive/MyDrive/Final year project /plant_disease.h5). tf.lite.TFLiteConverter.from_keras_model(model)*: This creates an instance of the TFLiteConverter class from the TensorFlow Lite library, which is used for converting the Keras model to a TensorFlow Lite model. The from_keras_model method is used to specify the Keras model that will be converted.converter.convert(): This method of the TFLiteConverter instance performs the actual conversion of the Keras model to a TensorFlow Lite model. The resulting tflite_model variable holds the converted model in a binary format suitable for deployment on mobile and embedded devices.

4.1.3 Integrating the TFLlite Model into an Application

model=tf.keras.models.load_model('/content/drive/MyDrive/Finalyear project /plant_disease.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

The TFLite model must first be loaded into your application. The TensorFlow Lite library's Interpreter class is what you may use to accomplish this. Running the inference on the flite model is the responsibility of the Interpreter class. The input data for the flite model must then be prepared. It is necessary for the input data to follow the same format requirements as the model's training set. The Interpreter class can be used to execute inference on the Flite model when the model has been loaded and the input data has been ready. Methods for configuring the input data, doing inference, and obtaining the output data are all provided by the Interpreter class. After running the inference, you will receive the output data from the TFLite model. You may need to post-process the output data to make it more readable or usable in your application. Finally, you need to integrate the TFLite model into your application. This can involve displaying the output data to the user, using the output data to make decisions in your application, or any other relevant use case.

5. Results and Discussions

The dataset for detecting leaf diseases is a fusion of a Kaggle dataset and a real-time dataset consisting of plant leaf images that are affected by various diseases. The dataset is a combination of images from the Kaggle dataset and newly collected images from diverse sources. The primary objective of this combined dataset is to facilitate the development of machine learning algorithms for identifying and classifying different types of leaf diseases. In the Kaggle dataset, there are 9000 plant leaf images that are affected by four types of diseases: powdery mildew, Septoria leaf spot, bacterial spot, and common rust.

The images were captured under controlled lighting conditions and from different angles to capture variations in the leaves. The real-time dataset consists of plant leaf images that were gathered from various sources, such as plant nurseries and botanical gardens. These images may differ in terms of lighting conditions, background, and other environmental factors. Both datasets are marked with the type of disease, and the combined dataset is designed to aid the research and development of computer vision and machine learning algorithms for leaf disease detection. The merged dataset can be utilized to train and assess machine learning models to precisely identify and categorize different types of leaf diseases real time.

The accuracy obtained by the dataset is 95.393 and it can be shown in the graph and it was done by CNN VGG-16. In a previous paper, it has been identified that multiple Object detection CNN is good when compared to SVM as shown below inTable6.1 describes the output of counting the total number of leaves in the training and validation dataset.

A Comparison Between Support Vector Machine (SVM) and Convolution Neural Network (CNN) Models for Image Classification Based on the accuracy.

Table 5.1. Accuracy for SVM vs CNN

Classification Method	Support Vector Machine Binary Classification	Convolution Neural Network Binary Classification
Accuracy	93%	95%

6. Conclusion

The key issues and shortfalls of earlier research employing CNNs to automatically detect domestic plant illnesses were covered in this study. This research makes it easier to identify leaf infections early, lowering the risk of crop loss and disease spread. CNN models are used to predict plant diseases accurately. It also has a respectable accuracy rating. An open CV is used to read the input image. CNNs perform far better than present systems. By adjusting different CNN settings to improve the system's recognition accuracy, the effectiveness of the suggested system and CNN design is assessed. Herein, the CNN algorithm is used to classify and predict leaf disease. CNN algorithm contains many numbers of hidden layers. When comparing CNN to SVM, the accuracy of CNN is 95.393. Colourized images are sent into CNN's input layer as the input. Layers of input deliver images to a hidden layer, which processes all types of input to produce an output. SVM performs better in the classification process than CNN. CNN performs better in prediction than SVM. CNN's classification process accuracy surpasses that of SVM and also SVM takes more amount of time than CNN.

7. Future Enhancement

Future enhancements could involve the development of mobile applications that utilize advanced image recognition technology to detect and diagnose plant diseases in real time. By snapping a photo of the diseased plant and getting advice on avoidance and therapy, these apps may help users detect plant illnesses more promptly. Another potential enhancement could involve the integration of weather and soil monitoring sensors with mobile applications. This could provide users with real-time data on weather patterns, soil conditions, and other environmental factors that could impact plant health. With this information, users could take proactive measures to ensure optimal growing conditions for their plants and prevent the occurrence of diseases.

References

[1] S. Kumar and S. R. Jindal, "Automated wheat disease detection using machine learning techniques," in Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019, pp. 449-453.

[2] P. Mahato et al., "Detection of Wheat Diseases using Artificial Neural Network," in Proceedings of the 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 11-15.

[3] K. Jha and P. K. Singh, "Wheat Disease Detection and Classification using Deep Learning," in Proceedings of the 2021 International Conference on Advances in Electronics, Computers and Communications (ICAECC), 2021, pp. 388-393.

[4] S. Singh and S. Gupta, "Identification of Wheat Leaf Diseases using Deep Convolutional Neural Networks," in Proceedings of the 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 44-48.

[5] T. R. Nayak et al., "Wheat leaf disease detection using deep convolutional neural network," in Proceedings of the 2021 International Conference on Computing, Communication and Signal Processing (ICCCSP), 2021, pp. 441-445.

[6] M. A. R. Khan et al., "A review on image processing based automated disease detection in plants," Computers and Electronics in Agriculture, vol. 151, pp. 313-332, 2018.

[7] PanelSunilS. Harakannanavar, JayashriM. Rudagi, Vee naI Puranikmath, Ayesha Siddiqua, R Pramodhini "Pla nt leaf disease detection using computer vision and machine learning algorithms in proceedings of 2022, Chinese global transitions.

[8] Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., Moussaoui, A.: Deep learning for plant diseases: detection and saliency map visualisation. In: Zhou, J., Chen, F. (eds.) Human and Machine Learning. HIS, pp. 93–117. Springer, Cham (2018).

[9] Chollet, F., et al.: Keras: The python deep learning library. Astrophysics Source Code Library (2018)

[10] Cortes, E.: Plant disease classification using convolutional networks and generative adverserial networks (2017)

[11] Ertam, F., Aydin, G.: Data classification with deep learning using tensorflow. In: 2017 International Conference on Computer Science and Engineering (UBMK), pp. 755–758. IEEE (2017)

[12] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[13] Hijazi, S., Kumar, R., Rowen, C.: Using convolutional neural networks for image recognition. Cadence Design Systems Inc., San Jose (2015)

[14] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)

[15] Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size (2016). arXiv preprint arXiv:1602.07360

[16] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[17] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM 60 (2017)

[18] McKenney, M.A.: Apple scab: what causes it and what can be done about it? (2018). https://hubpages.com/living/How-to-Recognize-Manage-and-Prevent-Apple-Scab-on-Your-Apple-Trees

[19] Miller, S.A., Beed, F.D., Harmon, C.L.: Plant disease diagnostic capabilities and networks. Ann. Rev. Phytopathol. 47, 15–38 (2009)

[20] Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1419 (2016)

[21] Naik, M.R., Sivappagari, C.M.R.: Plant leaf and disease detection by using hsv features and SVM classifier. Int. J. Eng. Sci. 6, 3794 (2016)

[22] Pratt, H., Coenen, F., Broadbent, D.M., Harding, S.P., Zheng, Y.: Convolutional neural networks for diabetic retinopathy. Procedia Comput. Sci. 90, 200–205 (2016)

[23] Rastogi, A., Arora, R., Sharma, S.: Leaf disease detection and grading using computer vision technology & fuzzy logic. In: 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), pp. 500–505. IEEE (2015)

[24] Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning deep convolutional networks for plant recognition. CLEF (Working Notes) 1391, 467–475 (2015)

[25] Rosebrock, A.: How to create a deep learning dataset using google images. https://www.pyimagesearch.com/2017/12/04/how-to-create-a-deep-learning-dataset-using-google-images

[26] Sambandam, P., Yuvaraj, D., Padmakumari, P., &

Swaminathan, S. (2023). Deep attention based optimized Bi-LSTM for improving geospatial data ontology. Data & Knowledge Engineering, 144, 102123.

[27] Alnuaimi, S. S. ., Rasheed, B. H. ., Yuvaraj, D. ., Sundaravadivel, P. ., & Isaac, R. A. . (2024). Hybrid Deep Learning Techniques for Large-Scale Video Classification. International Journal of Intelligent Systems and Applications in Engineering, 12(15s), 78–86.

[28] Rasheed, B. H. ., Yuvaraj, D. ., Alnuaimi, S. S. ., & Priya, S. S. . (2024). Automatic Speech Emotion Recognition Using Hybrid Deep Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 12(15s), 87–96.

[29] E, L. R. Flaih, D. Yuvaraj, S. K, A. Jayanthiladevi and T. S. Kumar, "Use Case of Artificial Intelligence in Machine Learning Manufacturing 4.0," 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 2019, pp. 656-659,

[30] S. S. Priya, M. Sivaram, D. Yuvaraj and A. Jayanthiladevi, "Machine Learning-based DDOS Detection," 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2020, pp. 234-237.

[31] Sivaram, M., Yuvaraj, D., Megala, G., Porkodi, V., & Kandasamy, M. (2019, December). Biometric Security and Performance Metrics: FAR, FER, CER, FRR. In 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) (pp. 770-772). IEEE.

[32] Sivaram, M., Shanmugapriya, S., Yuvaraj, D., Porkodi, V., Akbari, A., Hashim, W., & Huda, M. (2020). Decision Support System for Determining Academic Advisor Using Simple Additive Weighting. In Cognitive Informatics and Soft Computing (pp. 149-156). Springer, Singapore.

[33] ShanmugaPriya, S., Valarmathi, A., & Yuvaraj, D. (2019). The personal authentication service and security enhancement for an optimal strong password. Concurrency and Computation: Practice and Experience, 31(14), e5009

[34] Krishnan, S. N., Vadivel, P. S., Yuvaraj, D., Murthy, T. S., Malla, S. J., Nachiyappan, S., & Priya, S. S. (2022). Enhanced Route Optimization for Wireless Networks Using Meta-Heuristic Engineering. Computer Systems Science & Engineering, 43(1).

[35] Ahamed, B. B., Yuvaraj, D., Shitharth, S., Mirza, O. M., Alsobhi, A., & Yafoz, A. (2022). An Efficient Mechanism for Deep Web Data Extraction Based on Tree-Structured Web Pattern Matching. Wireless Communications and Mobile Computing, 2022.

[36] Alnuaimi, S. S. ., Rasheed, B. H. ., Yuvaraj, D. ., Sundaravadivel, P. ., & Isaac, R. A. . (2024). Hybrid Deep Learning Techniques for Large-Scale Video Classification. International Journal of Intelligent Systems and Applications in Engineering, 12(15s), 78–86.