RESEARCH ARTICLE OPEN ACCESS

Detection Of A Few Plants Leaf Disease Using Custom CNN

Debashis Dey [1], Samiran Das [2], Pallavi Thakuri [3], Dr. Bornali Gogoi [4], Prof. Nelson Varte [5]

[1],[2],[3] Student, Department of Computer Application, Assam Engineering College, Guwahati, Assam, India
[4],[5] Associate Professor, Department of Computer Application, Assam Engineering College, Guwahati, Assam, India

ABSTRACT

Todays agriculture faces numerous challenges, including increasing global food demand, climate change, limited resources, and the need for sustainable farming practices. The major reason for minimizing crop productivity is various diseases in plants. Plant leaf diseases detection is the important because profit and loss are depending on production. In this context, the model proposed is a deep learning-based approach for image recognition. A customized Convolutional Neural Network (CNN) is used in this project for image classification that detects three (3) different classes of Plant Leaf Diseases with accuracy ranging from 92% - 96.66% and suggests the Cure for the disease. This custom CNN model is using 7 layers, 48 kernels and 11 neurons. The proposed paper includes implementation steps like dataset collection, data preprocessing, model architecture building, User interface building, etc. using CNN to classify the leaves which are diseased based on the models prediction.

Keywords: Deep Learning, TensorFlow, Keras, Plant disease detection, LeNet.

I. INTRODUCTION

This work develops a smart system for detecting plant leaf disease by using a carefully designed CNN architecture. With the goal of providing farmers with an effective tool for crop

health management, it addresses the pressing need for early and accurate disease identification in agriculture, reducing losses and improving food security.

II. LITERATURE REVIEW

2.1 Customized CNN Architecture:

A customized CNN architecture designed to meet the difficulties of plant leaf disease detection is the foundation of the project. With this tailored architecture, optimal sensitivity and specificity are ensured while recognizing various leaf disease types.

2.2 Early Detection and Classification:

One of the primary objectives is to enable early detection of diseases, providing a crucial window for intervention and mitigation. The system goes beyond mere identification, employing advanced classification algorithms to precisely categorize detected diseases.

2.3 Resource Optimization:

Recognizing the resource constraints often faced in agricultural settings, the CNN architecture is strategically streamlined. This design choice aims to expedite the training process while minimizing the computational resources required for effective disease detection.

2.4 Empowering Farmers:

The overarching goal is to empower farmers with a tool that enhances their capacity to manage crop health effectively. By providing timely and accurate information, the system supports farmers in making informed decisions regarding disease control measures.

2.5 Local Plant Leaf Data Integration:

To further underscore the efficiency of the model in real-world scenarios, it incorporates a small local plant leaf dataset collected from the specific region. This inclusion demonstrates the system's adaptability and effectiveness in working with smaller datasets, reflecting the diverse nature of agricultural practices.

2.6 Impact on Agricultural Productivity:

Through its multifaceted approach, the project envisions a positive impact on overall agricultural productivity. The reduction of crop losses, coupled with sustainable farming practices, contributes to the long-term resilience of farming communities.

This initiative, rooted in technological innovation and agricultural science, holds the promise of transforming the way farmers engage with crop health management, ensuring a more secure and sustainable future for global agriculture.

III. RESEARCH METHODOLOGY

The proposed approach is related to predictions of Plant Diseases for early Prevention. The relevant research and the methodology used to position the suggested system in the literature are displayed.

Title: Agro-Samadhan - a plant leaf disease detection tool.

This Plant Leaf Disease Detection system utilizes a customized Lenet-5 CNN model with 2 Convolutional layers, 2 Maxpooling layers, 2 fully connected layers, and a Flatten layer to efficiently identify and classify plant diseases. The model strategically employs a minimal number of layers and neurons, accelerating training time and optimizing computer resources. This approach aims to facilitate disease monitoring, minimize crop losses, enable sustainable farming practices, and enhance farmer knowledge, thereby improving agricultural productivity and ensuring preventive disease control.

The entire procedure of plant leaf disease has been divided into the 6 steps:

A. Acquisition of Images

B. Pre-processing

C. Extraction of features

D. Final Classification

E. User Interface Formation

F. Model Deployment

Much effort is put into manually selecting the appropriate image to capture RGB pixels of the leaf. The photos undergo pre-processing to improve contrast and remove noise.

The data collection involves gathering plant leaf images in jpg format, labeling each with its respective name, and categorizing them into disease classes. The dataset is primarily sourced manually from cultivation fields, comprises of 300 leaf images categorized into 3 classes based on species and disease. Some of the image dataset is leveraged from online data source Plant Village to solve the problem of data scarcity. The goal is to efficiently train the model to detect diseases in plant leaves, including rare ones. The images are resized to 640x640 pixels for computational efficiency during model training and predictions. This small-scale dataset provides valuable insights into disease detection for a variety of plant leaves, optimizing resource utilization.

IV. PROPOSED SYSTEM

The suggested study uses a trained model that was learned using a deep learning technique to forecast plant leaf disease. It can also be changed to provide appropriate recommendations based on the type of disease that has been identified.

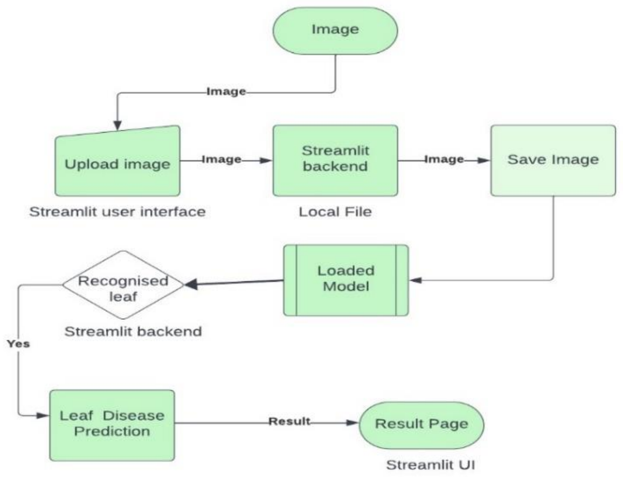


Fig 4.1: system workflow

4.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) has become a industry standard for resolving tasks related to images. The key to solving tasks such as recognition of objects, face, pose and more, all of them have one or another architecture variant of CNN.

4.2 Customized CNN model (Lenet-5):

Custom CNN Model architecture:

The Custom CNN Model is inspired by the LeNet- 5 architecture of CNN. It consists of 7 layers with 48 kernels and 11 neurons.

The key features and components of the Custom CNN model are:

4.2.1 Input Layer:

This layer receives images of dimension 640x640 pixels with 3 colour channels (RGB) i.e. (640,640,3).

4.2.2 Convolutional Layers:

Custom CNN starts with a convolutional layer that performs the initial feature extraction from the input image. The subsequent layers consist of residual blocks, each containing multiple convolutional layers. These convolutional layers learn and extract increasingly complex features from the input data.

We use the Convolution2D class to create the convolution layer in the neural net by passing a pair of arguments.

I. First Conv2D layer convolves the input image with 32 filter/kernel, each of size 3x3. ReLu (Rectified Linear Unit) activation is used here.
II. Second Conv2D layer convolves the input image with 16 filter/kernel, each of size 3x3. ReLu (Rectified Linear Unit) activation is used here.

4.2.3 Pooling Layers:

In addition to convolutional layers, Custom CNN incorporates pooling layers, such as max pooling, to downsample the feature maps. Pooling reduces the spatial dimensions while retaining the most salient information.

- First MaxPool layer uses 32 filters with pool size 3x3.
- Second MaxPool layer uses 16 filters with pool size 2x2.

4.2.4 Flatten Layer:

In this layer, the 2D feature maps is taken from the preceding convolutional and pooling layers and converts them into 1D vectors. This is done by arranging all the values of the feature maps into a linear array.

4.2.5 Fully Connected Layers:

Towards the end of the architecture, Custom CNN includes fully connected layers or Dense layer that convert the extracted features into class probabilities or predictions. These layers map the learned features to the desired output classes.

I. First Dense layer has 8 neurons and uses activation function **ReLu.**
II. Second Dense layer is also called output layer has 3 neurons and uses activation function **SoftMax**, this layer predicts the class with highest probability.

4.2.6 Activation function:

ReLu**:** ReLU introduces non-linearity to the model. The primary purpose of activation functions in neural networks is to introduce non-linearities, allowing the model to learn complex patterns and relationships in the data. which simply computes the function: $f(x) = \max(0, x)$.

SoftMax**:** The SoftMax activation function is utilized in the last dense layer of a neural network for multi-class classification tasks. Its primary purpose is to convert the raw output scores of the network into a probability distribution over multiple class.

4.2.7 Compiling and building the model: Keras library supplies the ‘compile’ method via the previously instantiated model.

Backpropagation:

Backpropagation is used to optimize model weights by iteratively adjusting the loss function. This minimizes prediction errors, enhancing the model’s accuracy.

Adam [Learning Rate = 0.0001]
metrics= accuracy
loss function = categorical cross entropy

Adam (short for Adaptive Moment Estimation) is an optimization algorithm commonly used for training neural networks to efficiently update model weights during the backpropagation process.

Metrics ([‘accuracy’]): ‘accuracy’ is chosen as the metric. Accuracy is a measure of the model’s correctness. It’s a commonly used metric for classification tasks because it provides a straightforward interpretation of the model’s performance.

Categorical cross entropy: “*Categorical Crossentropy*” is a loss function used in multi-class classification tasks, measuring the dissimilarity between predicted probability distributions (model output) and true class distributions.

Learning rate: The parameter 0.0001 represents the initial learning rate for the optimizer. It controls the size of the steps taken during optimization. A smaller learning rate can lead to more precise weight updates but might increase training time.

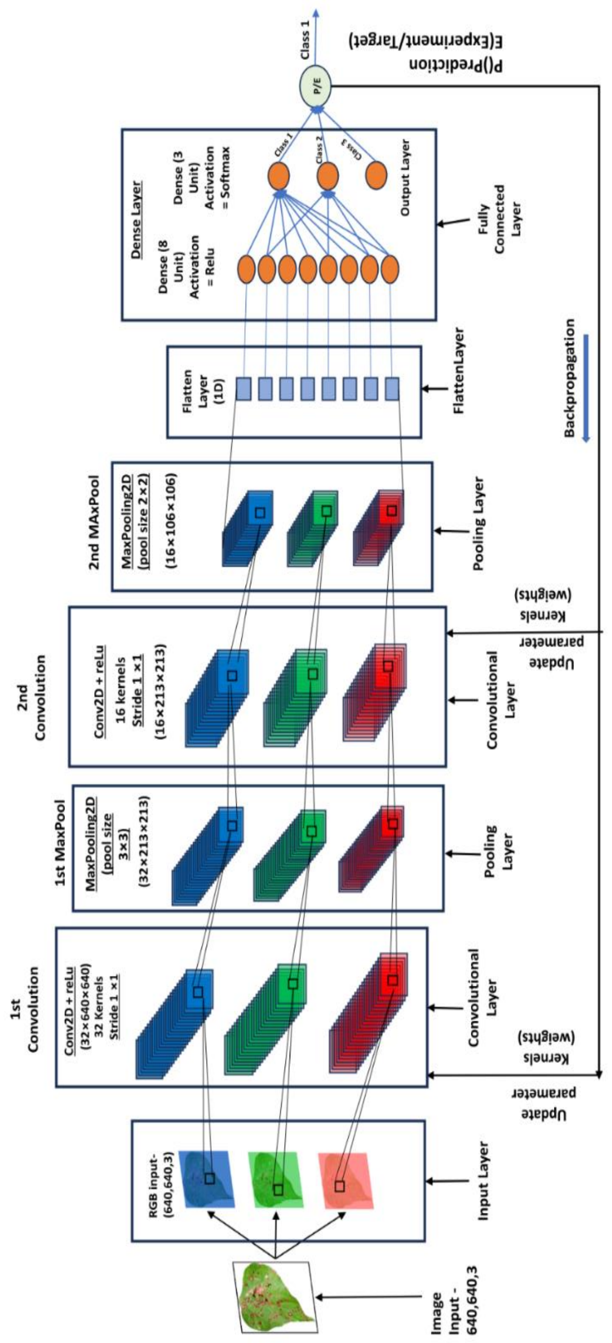


Fig 4.2: Network Architecture of the Model

4.3 LeNet TensorFlow Implementation:

We commence the implementation part with importing the below libraries:

- **TensorFlow**: Platform for implementing, training, and deploying machine-learning models.
- **Keras**: Library for implementing neural net architectures to be run on CPU as well as GPU.
- **NumPy**: Library for numerical calculation of N-dimensional matrices.

After importing all the libraries, we'll now bring in the data set. The Keras library has several dataset tools that can be used with ease for this purpose.

4.4 Data Splitting:

We also need to break the data set into following parts:

- **Training data set**:
  The image dataset includes 300 images in jpg format. Considering the test size of 0.2, the resulting training dataset consists of [300 × (1 - 0.2)] = 240 images.

- **Validation data set**:
  The validation dataset is of test Size 0.2 of the remaining data after training split. So, Validation Data = (240 ×0.2)= 48 images.

- **Test data set**:
  The test dataset is of test Size 0.2 of the total data which consists of 300 images. So, testing Data = (300 × 0.2) = 60 images.

0-1 is the required pixel intensity of the images among the data set, if the pixel intensity is not within that range, we need to normalize them among the data set.

4.5 Building User-Friendly Interface:

To enhance user experience, a graphical user interface (GUI) will be created. Users can effortlessly obtain accurate results and access recommended remedies for specific diseases. This component utilizes the Streamlit framework, an open-source Python library known for rapid model deployment in web interfaces. Developers can swiftly craft visually appealing user interfaces. The GUI receives user-input images, processes them through the model, predicts the specific plant disease, and provides relevant disease information along with potential solutions.

5. PERFORMANCE MATRIX:

To assess the predictability and reliability of the model, the following performance matrices have been used:

5.1 Accuracy:

The training process for the leaf disease detection model achieved an impressive accuracy rate ranging from 92% to 96.66%. During training, the training data went through 25 epochs in approx. 4 batches of 50 batch size each.

Last monitored performances:

- **Test accuracy**: The model gained a test accuracy of **96.66%** on the test dataset consisting of 60 images. The model has also proved its performance on unseen data.
- **Validation Accuracy:** The model gained a test accuracy of **97.91%.** It gauges the model's effectiveness on a validation set, helping to assess its performance during training and prevent overfitting**.**

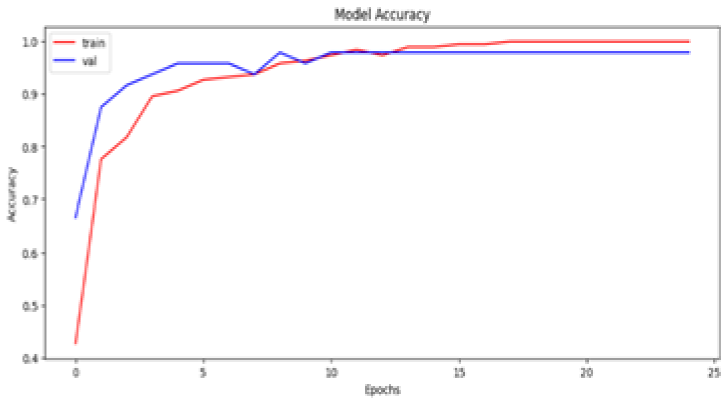


Fig 5.1: Model Training Accuracy graph

5.2 Loss:

Loss in training a model measures the disparity between predicted and actual values. It is calculated to guide the optimization process, enabling the model to adjust its parameters to minimize errors, improving performance on unseen data.

- **Training Loss:** The loss calculated on the training dataset during model training is 0. 01077591.The model demonstrates effective training as evidenced by a low training loss.
- **Validation Loss:** The loss computed on a validation dataset during model training is 0.9936374 suggesting that the model generalizes well to unseen data.

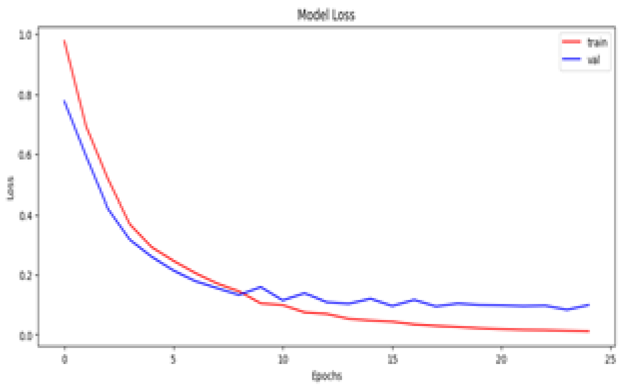


Fig 5.2: Model Training Loss graph

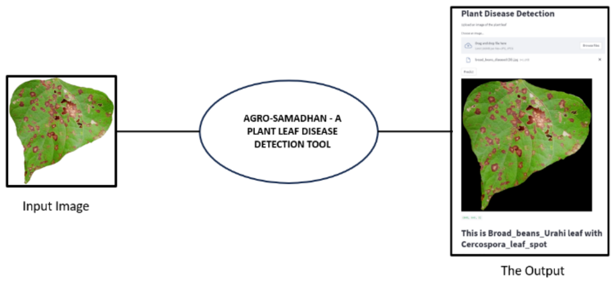
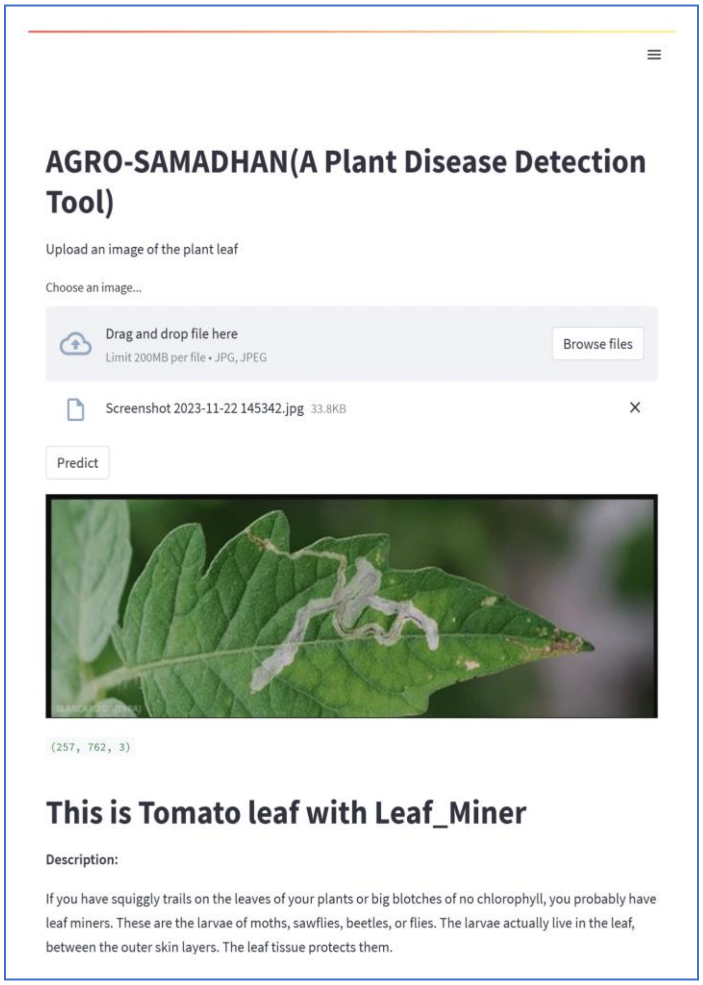

Input Image
AGRO-SAMADHAN - A PLANT LEAF DISEASE DETECTION TOOL
Plant Disease Detection
This is Broad_beans_Urahi leaf with Cercospora_leaf_spot
The Output

Fig 5.3: Context diagram of the system workflow

Fig 5.4: Front-End Demonstration (Output) of the plant disease prediction



AGRO-SAMADHAN(A Plant Disease Detection Tool)

Upload an image of the plant leaf

Choose an image...

Drag and drop file here
Limit 200MB per file • JPG, JPEG

Browse files

Screenshot 2023-11-22 145342.jpg 33.8KB

Predict

(257, 762, 3)

This is Tomato leaf with Leaf_Miner

Description:

If you have squiggly trails on the leaves of your plants or big blotches of no chlorophyll, you probably have leaf miners. These are the larvae of moths, sawflies, beetles, or flies. The larvae actually live in the leaf, between the outer skin layers. The leaf tissue protects them.

CONCLUSION

The study specifically concentrates on the detection of plant diseases, recognizing their detrimental impact on crop health and productivity. With the objective of addressing this issue, we have successfully developed robust machine learning models for disease prediction, achieving an accuracy rate of 92%-96.66%.

REFERENCES

[1] Juan K. Leonard, Ph.D. ‘*Image Classification and Object Detection Algorithm Based on Convolutional Neural Network’*

[2] Ashwani Kumar*, Sonam Srivastava. ‘*Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector’*

[3] Arpit Pawar, D.Bharath Kumar, Kuldeep Tamakuwala, Mrs. R. Logeshwari. *‘Realization of Plant Leaf Disease Recognition Using Lenet’*.

[4] Rahul Chauhan, Kamal Kumar Ghanshala, R.C Joshi. *‘Convolutional Neural Network (CNN) for Image Detection and Recognition’*.

[5] Sachin D. Khirade, *Plant Disease Detection using Image Processing*, ICCCCA, 2015.

[6] Prof. Patil Ashish and Patil Tanuja, “*Survey on Detection and Classification of Plant Leaf Disease in Agriculture Environment*”, International Advanced Research Journal in Science, Engineering and Technology, Vol.4, Sp. Issue 4, Jan 2017.

[7] Chit Su Hlaing, Sai Maung Maung Zaw, *‘Plant disease recognition for Smart Farming Using Model-based Statistical Features’*, IEEE 6th Global Conference on Consumer Electronics (GCCE), 2017.

[8] Vipinadas. M. J, ‘*A Survey on Plant Disease Identification’*, International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 6, Nov-Dec 2015

[9] Arpita Patel and Mrs. Barkha Joshi, “*A Survey on the Plant Leaf Disease Detection Techniques*”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 1, January 2017.

[10] Justine Boulent, Samuel Foucher, Jérôme Théau, Pierre-Luc St-Charles. ‘*Convolutional Neural Networks for the Automatic Identification of Plant Diseases’*. Article published on Front. Plant Sci., 23 July 2019 Sec. Technical Advances in Plant Science.