International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Plant Disease Detection Using Deep Learning

Boddapati Tarun Sai[1], Lattupalli Sai Charan Reddy[2], Karanakota Vignesh Goud[3], Dr. Shruthi SK[4]

[1,2,3]Dept. of Artificial, Intelligence and Data Science Methodist College Of Engineering and Technology (MCET) Hyderabad,India
[4]Assistant Professor Dept. of Computer Science, Methodist College Of Engineering and Technology (MCET) Hyderabad,India

ABSTRACT
The detection and diagnosis of plant diseases are critical for ensuring agricultural productivity and food security. This study explores the application of deep learning algorithms in identifying and classifying plant diseases, offering a scalable and efficient solution for modern agriculture. By leveraging techniques such as Convolutional Neural Networks (CNNs) and transfer learning, the proposed framework demonstrates high accuracy in disease
detection across various plant species. This paper also discusses the potential challenges and future research directions in deploying deep learning models for agricultural use.

KEYWORDS: Plant disease detection, deep learning, CNNs, transfer learning, agricultural technology.

1. INTRODUCTION

Agriculture is one of the economic sectors that play a huge role in the global economy, contributing to livelihood and food security. However, plant diseases pose a significant threat to the productivity of agriculture, bringing about severe economic losses and famine worldwide. Traditional methods, such as manual inspection of the plants, are time consuming, labor intensive, and open to human error. Thus, there is a great urgency for automated systems that are capable of accurately and promptly diagnosing plant diseases.

Plant diseases affect crop yield and quality, leading to financial losses for farmers and disruptions in the supply chain. These issues are even more significant in developing regions, where resources for manual inspection are scarce. Climate change and global trade have also introduced new pathogens, which have made the management of diseases in agriculture even more complex. These issues need innovative solutions that can balance scalability with precision.

2. LITERATURE REVIEW

2.1 Significance of Early Disease Detection

Detection of plant diseases at the right time is essential for minimizing crop losses and ensuring food security. Traditional approaches such as visual inspection by experts are limited by scalability and accuracy. Recent studies highlight the potential of automated systems to address these limitations. For example, Mohanty et al. (2016) showed that deep learning could be used for plant disease identification using image datasets with a significant accuracy over traditional methods.

2.2 Application of Deep Learning in Agriculture

Deep learning techniques have been applied successfully in many agricultural applications, such as yield prediction, weed detection, and disease identification. CNNs have been found to be very effective in analyzing visual data, such as images of infected leaves. For instance, Sladojevic et al. (2016) used deep neural networks to classify plant diseases based on leaf images, which demonstrates how CNNarchitectures outperform traditional machine learning methods in agricultural settings.

2.3 Existing Solutions and Limitations

revious works have used CNN for plant disease detection. Impressive results were observed after training on large datasets: Plant Village has gotten high accuracy in crops-disease identification, for tomato and potato crops especially, but data imbalance and demanding computational requirements are unresolved questions, as well as little generalization to field conditions in the results. Too et al. (2019) indicated that though the pre-trained models have much reduced the training time, their performance often drops on real-world scenarios because they are highly dependent on environment conditions and image quality variations.

2.4 Transfer Learning and Advanced Techniques

To overcome some of these limitations, transfer learning has been explored as a viable solution. Ferentinos (2018) investigated the use of pre-trained models like ResNet and Inception for plant disease detection, achieving high accuracy by fine- tuning these models on specific datasets. Despite these advancements, integrating multiple data modalities, such as spectral imaging and environmental factors, remains an area for future exploration.

2.5 Need for Generalizable and Scalable Systems

Although there is promising research that indicates deep learning's potential for plant disease detection, it is necessary to have generalizable and scalable systems. The models that exist now rely on controlled datasets, which restrict their application in various agricultural environments. Therefore, future studies must aim at developing robust models that can adapt to changing conditions such as different lighting, plant varieties, and disease severities..

3. MOTIVATION

With annual losses of $220 billion due to plant diseases (FAO), existing manual detection methods are no longer viable. Recent advances in AI offer real-time, highly accurate diagnostic systems that support sustainable farming. Automated solutions empower farmers, especially in remote areas, and enable early intervention while minimizing pesticide use. The increasing effects of climate change further justify the adoption of smart detection systems

4. OBJECTIVES

Develop High-Accuracy Models: Develop machine learning models that are capable of identifying plant diseases with over 95% accuracy, which can be relied upon in diverse agricultural settings.

Design Robust Image Processing Pipelines: Develop pipelines to preprocess and analyze plant images, addressing challenges such as varying lighting conditions, angles, and background noise.

Support Multiple Crop Types: Develop systems that are adaptable to a wide variety of crops, enabling universal applicability across different agricultural regions.

Actionable Advice on Integrated Platforms: Provide insights on actionable recommendations, from easy treatment plans for diseases up to intuitive mobile and web interfaces designed for farmers.

5.METHODOLOGY

The methodology for detecting plant diseases using machine learning algorithms will involve a number of key steps that integrate data collection, preprocessing, model development, and deployment. This section shall outline a structured approach for building a robust and scalable system.

5.1 Data Collection

PlantVillage dataset and other sources with healthy and diseased leaf images under varying conditions.

5.2 Data Preprocessing

Standardize image resolution, remove noise, augment data for robustness

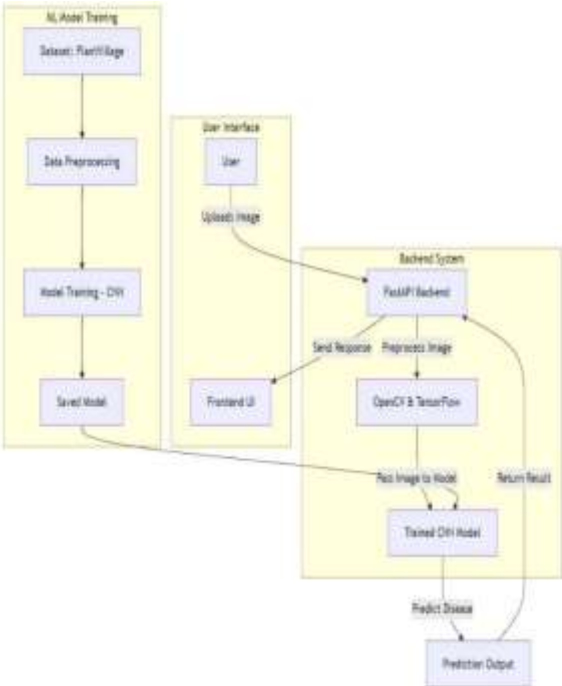
5.3 Feature Extraction

This involves extracting unique patterns of the data.Visual FeaturesExtract color patterns, texture, shape, and edge detail from images of leaves and fruits through image processing.
Utilize Convolutional Neural Networks (CNNs) for automatic feature extraction from images.
Contextual Features: Incorporate environmental factors (e.g., humidity, soil quality) as additional features to improve model predictions.

5.4 Model Selection

The choice of machine learning algorithms depends on the complexity and scale of the problem.Deep Learning Models: ResNet, VGG, and EfficientNet used with transfer learning for better accuracy.

6.PROPOSED ARCHITECTURE



The proposed architecture for plant disease detection is based on advanced machine learning techniques that will provide accurate and actionable diagnostics. The process starts with the user, who in this case is a farmer, launching the mobile application and uploading an image of a suspected diseased plant leaf. Once the image is uploaded, it undergoes preprocessing with the help of OpenCV. In the preprocessing, different techniques like resizing, normalization, and noise reduction are applied so that the image is optimized for analysis.

6.1 Collection of Input data

Collect high-quality datasets, containing images of diseased and healthy plants. These datasets are variations of lighting, angles, and environmental conditions to be more robust with the model.

6.2 Preprocessing

Image preprocessing includes techniques such as resizing, normalization, and augmentation that will improve model training. It further includes the application of noise reduction algorithms to remove unnecessary information.

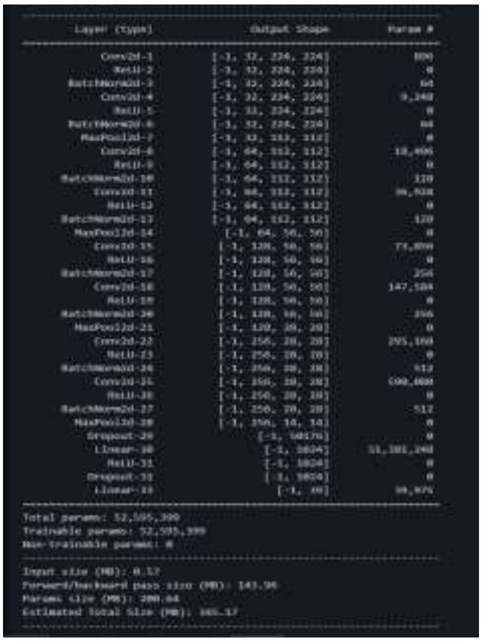
6.3 Machine Learning Models

Convolutional Neural Networks: Used for image classification and feature extraction.

Support Vector Machines (SVMs): Used for binary classification tasks. Ensemble Learning: Combines multiple models to improve overall accuracy and reduce biases.

7. IMPLEMENTATION: Model : Convolutional Network (CNN)

- Basically, First we Resize every image into 224 x 224. After that this image feed into the Convolutional Neural Network. We feed color image so it has 3 channels RGB.
- First conv layer we apply 32 filter size or output channels. That means 32 different filters apply to the images and try to find features and after that using 32 features, we create a features map that has channels 32. So from 3 x 224 x 224 it will become 32 x 222 x 222.
- After that we are applying ReLU activation function to remove non linearity and after that we are applying Batch Normalization to normalize the weights of the neuron.
- After that this image we feed to the max pool layer which takes only the most relevant features only so that why we get the output image in shape 128x 56 x 56.
- After that, we feed this image to the next convolutional layer and its process is the same as mentioned above.
- At last, we flatten the final max pool layer output and feed to the next linear layer which is also called a fully connected layer, and finally, as a final layer, we predict 39 categories.



Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 224, 224]	896
ReLU-2	[-1, 32, 224, 224]	0
BatchNorm2d-3	[-1, 32, 224, 224]	64
Conv2d-4	[-1, 32, 224, 224]	9,248
ReLU-5	[-1, 32, 224, 224]	0
BatchNorm2d-6	[-1, 32, 224, 224]	64
MaxPool2d-7	[-1, 32, 112, 112]	0
Conv2d-8	[-1, 64, 112, 112]	18,496
ReLU-9	[-1, 64, 112, 112]	0
BatchNorm2d-10	[-1, 64, 112, 112]	128
Conv2d-11	[-1, 64, 112, 112]	36,928
ReLU-12	[-1, 64, 112, 112]	0
BatchNorm2d-13	[-1, 64, 112, 112]	128
MaxPool2d-14	[-1, 64, 56, 56]	0
Conv2d-15	[-1, 128, 56, 56]	73,856
ReLU-16	[-1, 128, 56, 56]	0
BatchNorm2d-17	[-1, 128, 56, 56]	256
Conv2d-18	[-1, 128, 56, 56]	147,584
ReLU-19	[-1, 128, 56, 56]	0
BatchNorm2d-20	[-1, 128, 56, 56]	256
MaxPool2d-21	[-1, 128, 28, 28]	0
Conv2d-22	[-1, 256, 28, 28]	295,168
ReLU-23	[-1, 256, 28, 28]	0
BatchNorm2d-24	[-1, 256, 28, 28]	512
Conv2d-25	[-1, 256, 28, 28]	590,080
ReLU-26	[-1, 256, 28, 28]	0
BatchNorm2d-27	[-1, 256, 28, 28]	512
MaxPool2d-28	[-1, 256, 14, 14]	0
Dropout-29	[-1, 50176]	0
Linear-30	[-1, 1024]	51,381,248
ReLU-31	[-1, 1024]	0
Dropout-32	[-1, 1024]	0
Linear-33	[-1, 39]	39,975

Total params: 52,595,399
Trainable params: 52,595,399
Non-trainable params: 0

Input size (MB): 0.57
Forward/backward pass size (MB): 143.96
Params size (MB): 200.64
Estimated total size (MB): 345.17

- So as a model output we get tensor 1x16 size. And from that tensor, we take an index of the maximum value in the tensor

8.TRAIN TEST SPLIT:

Here in the code we first getting indices and then split the data into train , test and validation data. Total 36584 for train , 15679 for validaiton and remaining images for testing.

```
indices = list(range(len(dataset)))
split = int(np.floor(0.85 * len(dataset)))  # train_size
validation = int(np.floor(0.70 * split))  # validation
print(f"length of train size :{validation}")
print(f"length of validation size :{split - validation}")
print(f"length of test size :{len(dataset)-validation}")
```

```
length of train size :36584
length of validation size :15679
length of test size :24902
```

```
train_indices, validation_indices, test_indices = (
    indices[:validation],
    indices[validation:split],
    indices[split:],
)
```

```
batch_size = 64
train_loader = torch.utils.data.DataLoader(
    dataset, batch_size=batch_size, sampler=train_sampler
)
test_loader = torch.utils.data.DataLoader(
    dataset, batch_size=batch_size, sampler=test_sampler
)
validation_loader = torch.utils.data.DataLoader(
    dataset, batch_size=batch_size, sampler=validation_sampler
)
```

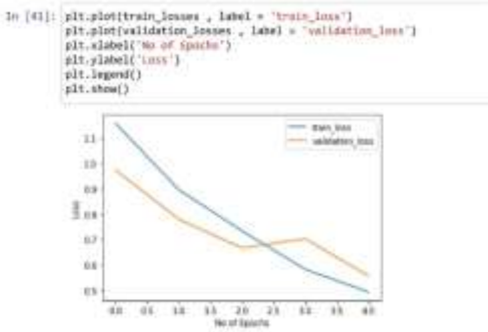
SubsetRandomSampler is used to sample our data. Here we are creating an object of SubsetRandomSampler Object and later we will use this sampler in train data loader and test

9.ACCURACY:

```
def accuracy(loader):
    n_correct = 0
    n_total = 0

    for inputs, targets in loader:
        inputs, targets = inputs.to(device), targets.to(device)

        outputs = model(inputs)

        _, predictions = torch.max(outputs, 1)

        n_correct += (predictions == targets).sum().item()
        n_total += targets.shape[0]

    acc = n_correct / n_total
    return acc
```

The above function is used for finding the accuracy of the mode Using this Current Model we are getting accuracy **98%** on Train data, **96 %** On Validation data and **83%** on Test data

10.COMPARATIVE GRAPHS :



11.CONCLUSION

In conclusion, Deep learning provides high accuracy in identifying plant diseases, surpassing traditional methods.Automates the detection process, saving time and reducing dependency on human expertise.Can be scaled and adapted for various crops, regions, and farming systems.Reduces the need for frequent expert consultations and labor-intensive monitoring, lowering overall costs for farmers.Encourages eco-friendly farming by optimizing pesticide use and protecting the environment.

REFERENCES

1. Chohan, Murk, et al. "Plant disease detection using deep learning." International Journal of Recent Technology and Engineering 9.1 (2020): 909-914.. .
2. Ferentinos, Konstantinos P. "Deep learning models for plant disease detection and diagnosis." Computers and electronics in agriculture 145 (2018): 311-318.
3. R. prkrum "control-value Theory of ArchivementEmotions,"Educational Psychologist,vol56
4. Barbedo, Jayme Garcia Arnal. "Plant disease identification from individual lesions and spots using deep learning." Biosystems engineering 180 (2019): 96-107..
5. Panchal, Adesh V., et al. "Image-based plant diseases detection using deep learning." Materials Today: Proceedings 80 (2023): 3500-3506.