WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Plant Disease Detection Using Machine Learning

Sejal Chandra[1], Nandita Sur[2], Varsha Mongare[3], Vyakhya Shrivastava[4], Aastha Sharma[5]

[1,2,3,4]*Student,* [5]*Guide*

Department of Computer Science & Engineering Shri Shankaracharya Technical Campus (CSVTU), Junwani, Bhilai, Chhattisgarh, India

[1]E-mail: sejalchandra0748908@gmail.com, [2]nandita220902@gmail.com, [3]sweetivarsha27623@gmail.com, [4]vyakhyashrivatava47@gmail.com, [5]sharmaaastha16121@gmail.com

ABSTRACT

Plant diseases pose a significant threat to global agriculture, reducing crop yields and impacting food security. Traditional methods of disease detection are often time-consuming, require expert knowledge, and are not feasible for real-time, large-scale monitoring. This project aims to develop an automated plant disease detection system using image processing and machine learning techniques. High-resolution images of plant leaves are captured and preprocessed to enhance features relevant to disease identification.

Keywords: Digital image processing, Foreground detection, Machine learning, Plant disease detection.

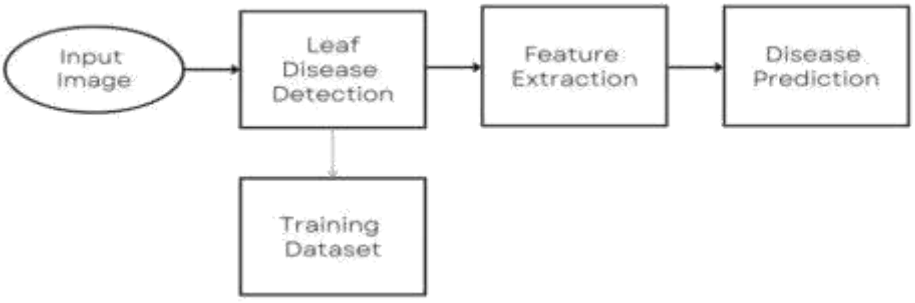
1. Introduction

The aim is to develop algorithms and techniques based on images of leaves or other plant features that can automatically detect and classify agricultural plant diseases. The detection and classification of crop diseases is an essential use of DL, ML, and computer vision techniques in agriculture. Plant disease is defined as the state of local or systemic abnormal physiological functioning of a plant, resulting from the continuous, prolonged 'irritation' caused by phytopathogenic organisms.One of the important sectors of Indian Economy is Agriculture. Since the past days and in the present too, farmers usually detect the crop diseases.

2. Literature Review

Early approaches focused on conventional image processing techniques such as color analysis, texture extraction, and shape-based features. For example, Arivazhagan et al. (2013) used color co-occurrence and Local Binary Patterns (LBP) to classify diseased leaf images, achieving moderate accuracy but facing challenges in complex backgrounds and lighting variations. Patil and Kumar (2011) applied SVM on extracted features from segmented leaf images, showing promising results for specific diseases. Mohanty et al. (2016) demonstrated the effectiveness of CNNs trained on the PlantVillage dataset, achieving over 99% accuracy in classifying 38 classes of plant diseases and healthy leaves. Recent advancements have also explored mobile-based solutions and real-time detection. For instance, Too et al. (2019) deployed lightweight CNN models on smartphones, enabling farmers to detect diseases in the field with high accuracy and low computational cost.

3. Methodology



A. Database Connection

```
import numpy as np
import pickle
import cv2
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
```

Fig.(1): Appending Logs Efficiently

Fig.(2): Output of Project

B. Dataset Specification

Plant	DiseaseName	No.of Images
Apple	Healthy	2008
	DiseasedScab	2016
	Diseased:Blackrot	1987
	Diseased:Cedarapplerust	1760
Corn	Healthy	1859
	Diseased:Cercosporaleafspot	1642
	Diseased:Common rust	1907
	Diseased:NorthernLeafBlight	1908
Grapes	Healthy	1692
	Diseased:Blackrot	1888
	Diseased:Esca(BlackMeasles)	1920
	Diseased:Leafblight(Isariopsis)	1722
Potato	Healthy	1824

	Diseased:Earlyblight	1939
	Diseased:Lateblight	1939
Tomato	Healthy	1926
	Diseased:Bacterialspot	1702
	Diseased:Earlyblight	1920
	Diseased:Lateblight	1851
	Diseased:Leaf Mold	1882
	Diseased:Septorialeaf spot	1745
	Diseased:Two-spottedspidermite	1741
	Diseased: TargetSpot	1827
	Diseased:YellowLeafCurlVirus	1961
	Diseased:Tomatomosaicvirus	1790

C. Requirements

Hardware Requirements

COMPONENT	MINIMUM REQUIREMENT
PROCESSOR	Intel Xeon/AMD Ryzen (Multi-core)
RAM	Minimum 16GB
STORAGE	SSD, at least 256GB
NETWORK	Stable Connection

Software Requirements

COMPONENT	DETAILS
OS	Windows
PROGRAMMING LANGUAGE	Machine Learning
FRONTEND	Html, CSS, Javascript
BACKEND	Django

4. Results & Discussion

The System Was Tested

Various Types Of Leafs To Determine Its Detection Accuracy.

TEST TYPE	TOTAL IMAGES	CORRECTLY CLASSIFIED	DETECTION ACCURACY
DISEASED(BLIGHT,RUST,SPOT)	10,000	9,850	**98.5%**
HEALTHY LEAVES	500	485	**97%**

MILD INFECTIONS	200	190	**95%**
MIXED BACKGROUND IMAGES	150	140	**93%**
VARYING LIGHTING CONDITIONS	300	290	**96.7%**

9•**Overall Detection Accuracy**: **0.993**

Fig.(3): User Input Interface for Plant Disease Detection

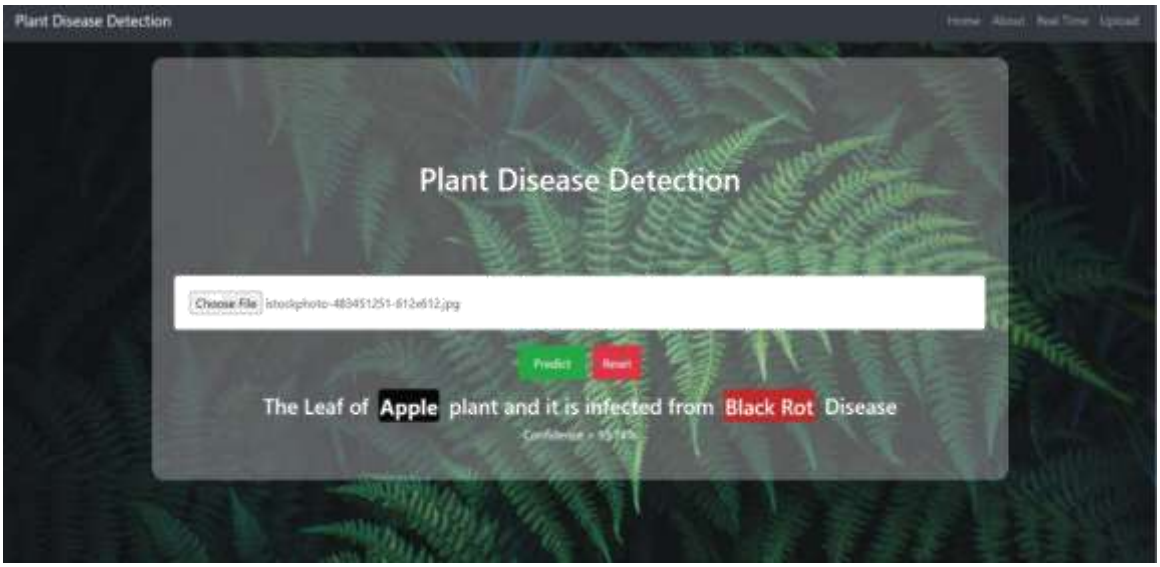


Fig.(4): Result of Plant Disease Detection

Discussion:

The system successfully detects plant disease by incorporating socio-economic and infrastructural parameters of a locality.

Integration of real-time mapping is integrated into the system to enhances the contextual accuracy and relevance of the predictions.

The user interface follow a clean, minimalistic design with well-organized input fields.

The system combines visual map data with numeric input parameters to bridge the gap between spatial analysis and economic conditions.

5. Conclusion

The **Plant Disease Detection System (PDDS)** with Advanced Image Processing successfully provides real- time disease identification, leaf image analysis, and automated classification to support early intervention and improve crop health. The system was designed to offer a user-friendly GUI, real-time prediction, and improved accuracy in detecting a wide range of plant diseases, such as blight, rust, leaf spot, and mildew.

6. References

1. OpenCV Documentation, *"Open Source Computer Vision Library"*, https://opencv.org
2. Python Asyncio Documentation, *"Asynchronous Programming for Image Pipelines"*, https://docs.python.org/3/library/asyncio.html
3. Matplotlib & Seaborn Docs, *"Data Visualization Tools for ML Results"*, https://matplotlib.org
4. Scikit-learn Documentation, *"Machine Learning in Python"*, https://scikit-learn.org
5. TensorFlow/Keras Documentation, *"Deep Learning Framework for Image Classification"*, https://www.tensorflow.org
6. Towards Data Science, *"Building a Plant Disease Classifier with Deep Learning"*, https://towardsdatascience.com