

Plant Disease Detection System

¹Fazle Hasan, ²Pranay Bobde, ³Vedant Naikwade,
⁴Yash Thavkar, ⁵Prof. Vaishali Patil, ⁶Dr. Sharda Chhabria

^{1,2,3,4}Students, ⁵Assistant Professor, ⁶Project Incharge
^{1,2,3,4,5,6}Department Artificial Intelligence
G.H. Rasoni Institute of Engineering & Technology Nagpur, India

Abstract: Plant diseases are considered one of the main factors influencing food production and minimize losses in production, and it is essential that plant diseases should be recover with the recommendation of specialist. The recent expansion of deep learning methods which has been used in this project for plant disease detection, offering a robust tool with highly accurate results. This project identify the state of the art, with the use of convolutional neural networks (CNN) in the process of identification and classification of plant diseases. With the help of CNN, we can able to diagnose the diseases of plants and recommend the best medicine for the infected plant. This model successfully achieved the average accuracy of 95.69%. And then connecting the model with localhost website with help of API.

Keywords: - CNN (convolution neural network), GDP (gross development Product), DL (Deep learning), OS (Operating system) and IO (Input Output)

I. INTRODUCTION

Plant Disease are one of the greatest problems in the entire globe it affects the country's GDP is affected due to the Plant disease. Due to lack of knowledge the farmers misunderstand the disease and prefer erroneous medicine, so to overcome the problem we came up with a web application this will help the farmers to get proper medication for the plant with just a click of photo. The implemented models achieved a disease-classification accuracy rates of 98.42%, 99.11%, 97.02%, and 99.56% using InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0, respectively, which were greater than that of traditional handcrafted-feature-based approaches. We are using various technique to suggest proper medication, for example we used Convolution neural network for training the model and to learn from the dataset to display the best accuracy and application program interface to connect to our program, the program will be hosted on the local device and then will be connected with the CNN model to predict the Disease and then diagnose it with the best medicine recommendation for infected plant. Our model has created the multiple models for multiple variety of plants.

II. RELATED WORK

In this Project various datasets of plant like Potato, Tomato and Bell Pepper are collected and stores them separately for further training purpose.

To hit the problem statement, we designed several models to achieve best accuracy.

Trained multiple models for various category of plants to carry off a targeted output.

Based on training dataset we predict the disease of plant to reach a desirable result. Selected the best accomplishing algorithm to overcome the problem statement.

Our model describe these methods for the detection of plant diseases based on their leaf images, automatic detection of plant disease is done by the image processing and machine learning. The different leaf images of plant disease are collected, and feature extracted of the various machine learning methods.

The timely identification and early prevention of crop diseases are essential for improving production. In this paper, deep convolutional-neural-network (CNN) models are implemented to identify and diagnose diseases in plants from their leaves, since CNNs have achieved impressive results in the field of machine vision.

Standard CNN models require many parameters and higher computation cost. In this paper, we replaced standard convolution with depth separable convolution, which reduces computation cost.

The implemented models were trained with an open dataset consisting of 14 different plant species, and 38 different categorical disease classes and healthy plant leaves. To evaluate the performance of the models, different parameters such as batch size, dropout, and different numbers of epochs were incorporated.

Standard CNN	Customized CNN
1 Input Layer	1 Input Layer
2 Convolutional Layer	6 Convolutional Layer

2 Max Pooling Layer	6 Max Pooling Layer
2 Dropout Layer	2 Dropout Layer
0 Flatten Layer	1 Flatten Layer
2 Fully Connected Layer	2 Fully Connected Layer
1 Output Layer	1 Output Layer

III. METHODS

The procedure of proposed system starts with data collection, pre-processing, feature extraction, training and testing with different machine learning and deep learning algorithm and lastly evaluation of models by several measuring parameters.

[1] Tensor Flow

Tensor Flow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensor Flow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of Tensor Flow, named Tensor Flow 2.0, in September 2019. TensorFlow can be used in a wide variety of programming languages, including Python, JavaScript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors.

[2] Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

[3] NumPy

This article will help you get acquainted with the widely used array-processing library in Python, NumPy. It is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

[4] IPython.display

IPython (Interactive Python) is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language that offers introspection, rich media, shell syntax, tab c, completion, and history. IPython provides the following features

- Interactive shells (terminal and Qi-based).
- A browser-based notebook interface with support for code, text, mathematical expressions, inline plots and other media.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into one's own projects.
- Tools for parallel computing.

[5] OS

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see `open()`, if you want to manipulate paths, see the `os.path` module, and if you want to read all the lines in all the files on the command line see the `fileinput` module. For creating temporary files and directories see the `tempfile` module, and for high-level file and directory handling see the `shutil` module. Notes on the availability of these functions: The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function `os.stat(path)` returns `stat` information about `path` in the same format (which happens to have originated with the POSIX interface).

Extensions peculiar to a particular operating system are also available through the `OS` module but using them is of course a threat to portability. All functions accepting `path` or file names accept both bytes and string objects, and result in an object of the same type, if a `path` or file name is returned. On VxWorks, `os.popen`, `os.fork`, `os.execv` and `os.spawn*p*` are not supported. On Web

Assembly platforms wasm32-emscripten and wasm32-wasi, large parts of the OS module are not available or behave differently. API related to processes (e.g. fork (), execve ()), signals (e.g. kill (), wait ()), and resources (e.g. nice ()) are not available. Others like getuid () and getpid () are emulated or stubs.

[6] Fastapi

FastAPI is a modern, high-performance web framework for building APIs with Python based on standard type hints. It has the following key features:

- Fast to code: - It allows for significant increases in development speed.
- Reduced number of bugs: - It reduces the possibility for human-induced errors.
- Intuitive: - It offers great editor support, with completion everywhere and less time debugging.
- Straightforward: - It's designed to be uncomplicated to use and learn, so you can spend less time reading documentation
- Short: - It minimizes code duplication.
- Robust: - It provides production-ready code with automatic interactive documentation.
- Standards-based: - It's based on the open standards for APIs, OpenAPI and JSON Schema.

[7] Uvicorn

Uvicorn is an ASGI web server implementation for Python. Until recently Python has lacked a minimal lowlevel server/application interface for a sync framework. The ASGI specification fills this gap and means we're now able to start building a common set of tooling usable across all a sync framework

The python package uvicorn was scanned for known vulnerabilities and missing license, and no issues were found. Thus, the package was deemed as safe to use. See the full health analysis review

[8] IO

The io module provides Python's main facilities for dealing with various types of I/O. There are three main types of I/O: text I/O, binary I/O and raw I/O. These are generic categories, and various backing stores can be used for each of them. A concrete object belonging to any of these categories is called a file object. Other common terms are stream and file-like object. Independent of its category, each concrete stream object will also have various capabilities: it can be read-only, write-only, or read-write. It can also allow arbitrary random access (seeking forwards or backwards to any location), or only sequential access (for example in the case of a socket or pipe). All streams are careful about the type of data you give to them. For example, giving a str object to the write () method of a binary stream will raise a Type Error. So will giving a byte's object to the write () method of a text stream.

[9] Pillow

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating, and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it.

Pillow was announced as a replacement for PIL for future usage. Pillow supports many image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

[10]Nodejs

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine (i.e., V8 engine) and executes JavaScript code outside a web browser, which was designed to build scalable network applications.

Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

IV. IMPLEMENTATION

The applications of Neural Networks (NNs) have been formulated for classification of diseases that affect plant leaves. Recognizing the disease is mainly the purpose of the introduced approach. Thus, Convolutional Neural Networks Algorithm was tested on 3 plants.

1) Potato

2) Tomato

3) Pepper Bell

The experimental results indicate that the approach can significantly support an accurate detection of Leaf

Diseases with little computational effort. Convolutional approach is better as compared to Feed Forward

approach as it requires lesser number of iterations because a smaller value of gradient facilitates quick adaption and also reduces the Error.

V. RESULTS

With the help of Tensor Flow, NumPy, Pillow we get the accuracy of 99.6% we suggest the medicine based on the disease by which the plant is being infected by virus or Bacteria.

The accuracy score of LSTM is above 90%. Which proves that using deep learning approach will definitely boost the result.

- Hence, we have identified the state of the art, with the use of convolutional neural networks (CNN) in the process of identification and classification of plant diseases. With the help of CNN, we can be able to diagnose the disease as well as the best cure for plant. With the accuracy of 99.65%.

VI. SUMMARY

The motive of this paper is "Plant Disease Detection System" with the help of Kaggle fetched all the dataset in cluster format. This paper is results of the studies done on various deep learning approaches. One can easily find the working of this algorithms by the help of this paper. The results show the applicability of this model in various field such as farmers, agriculture, Argo pharmacist, colleges and academics etc. This work will help others in field of research and advancement in technology.

VII. ACKNOWLEDGMENT

Special Thank to Dr. Sharda Chhabria and Prof Vaishali Patil for guiding throughout the whole process of building the system. For discussion on various algorithm and techniques of deep learning which makes the model to reach the best average accuracy to detect the plant disease and diagnose it with best recommended medicine.

REFERENCES

1. Multidisciplinary Digital Publishing Institute Electronics 2021, Volume: 10, Issue: 1388
2. <https://www.mdpi.com/journal/electronics>
3. International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 07 | July 2020
4. <https://www.irjet.net/archives/V7/i7/>
5. Frontiers in Plant Science September 2016 Volume 7 |Article 1419
6. <https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full>
7. Dataset :
8. <https://www.kaggle.com/datasets/vipooool/new-plant-diseases-dataset/code>
9. Application of Machine Learning in Agriculture
10. <https://www.sciencedirect.com/science/article/pii/B9780323905503000072#:~:text=These%20techniques%20are%20based%20on,for%20reliable%20plant%20disease%20detection.>
11. Using Deep Learning for Image-Based Plant Disease Detection <https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full>
12. Artificial Intelligence-Based Drone System for Multiclass Plant Disease Detection Using an Improved Efficient
13. Convolutional Neural Network <https://www.frontiersin.org/articles/10.3389/fpls.2022.808380/full>
14. AI-powered banana diseases and pest detection <https://plantmethods.biomedcentral.com/articles/10.1186/s13007-019-0475-z>
15. Artificial intelligence for plant disease detection: past, present, and future <https://www.degruyter.com/document/doi/10.1515/9783110691276-011/html?lang=en>
16. Plant Disease Detection using AI based VGG-16 Model
17. https://thesai.org/Downloads/Volume13No4/Paper_84Plant_Disease_Detection_using_AI_based_VGG_16_Model.pdf
18. Plant Disease Detection using Advanced Deep Learning and ReactJS <https://towardsdatascience.com/plant-disease-detection-using-advanced-deep-learning-and-reactjs>