

DEEP LEARNING BASED MULTI-PLANT AND MULTI-CROP LEAF DISEASE DETECTION AND CLASSIFICATION

Mrs T. Nivethitha, Syed Anas A , Sreekesh P, Sonu K Achenkunju And Senthamizhselvan S

¹Assistant Professor, ^{2,3,4,5} UG Scholar
Department of Electronics and Communication Engineering,
Hindusthan College of Engineering and Technology, Otthaklmandapam, Coimbatore, India.

-----***-----

ABSTRACT :

Now days the plant disease detection is very important because agriculture is the backbone of the county like India. We get various raw materials from agriculture and especially crops, which serve as a staple food for people. But the crops and plantations get destroyed mainly due to two major reasons, first reason is destruction by natural calamities such as flood, earthquake, drought, famine, etc. second reason is the destruction by the pathogens such as viral, bacterial and fungal. Farmer is not aware what type of disease plant having and how to prevent them from these diseases. Plant diseases are a serious factor in crop losses in the world's agriculture. Due to a lack of specialized expertise, detecting illnesses in plants is complex and difficult. Plant diseases are one of the primary causes of decreased agricultural production quality and quantity. The human eye approach is outdated, imprecise, time-consuming, and unsuitable for significantly larger farm fields of diagnosing infections. Additionally, it costs a lot because professionals must continuously check it. With ongoing changes in plant structure and cultivation techniques, new diseases are constantly arising on plant leaves. Thus, accurate classification and detection of plant leaf diseases in their early stages will limit the spread of the infection and support the healthy development of plant production. The identification of plant diseases from leaf photos is made possible by deep learning-based algorithms. This work proposes a novel lightweight deep convolutional neural network (CNN) model for obtaining high-level hidden feature representations. The deep features are then fused with traditional handcrafted local binary pattern (LBP) features to capture local texture information in plant leaf images. The proposed model is trained and tested on three publicly available datasets.

Keywords— Convolutional Neural Network ; Deep Learning; Matrix Laboratory ; Plant Diseases; Datasets

I. INTRODUCTION

Plants and fruits are primary energy sources for humans and animals. The plant leaves play an essential role in the growth of plants through photosynthesis. The leaves of the plants are also beneficial for humanity because of their medicinal properties (Dhingra et. at 2018). In the Asian and African countries, more than 50% population directly depends upon agriculture for food, shelter, medicine, and employment (Patel et al., 2017). The economy and living of the small farm holders entirely depends upon the production of agriculture products. However, any type of disease degrades the quality of crops and agriculture products and hence leads to huge economical loss of framers. Various crops are damaged because of diseases which declines the quantity and quality of agricultural production. Plant diseases are broadly categorized into parasitic diseases and non-parasitic diseases. The parasitic diseases may occur due to various pathogens such as viruses, bacteria, fungus, chromista, etc.; pests such as millet, mammals, slugs, rodents, etc; weeds such as monocots and dicots. In contrast, non-parasitic plant diseases may occur due to excess or shortage of water, temperature, irradiation, minerals, and nutrients (Kaur et al., 2019) (Poojary et.al 2019). Crop diseases are chief hazard to food security, but disease detection is challenging in many parts of the world as a result of the unavailability of infrastructure. Recent growth in smartphone technologies paved the way for computer vision-based disease detection. Plant diseases establish financial disasters to the smallholder farmers whose

livelihood depends on vigorous crops (Harvey et al., 2015). The manual leaf disease detection technique is time-consuming, tedious, and depends upon expert knowledge. The manual leaf disease detection is often subjected to less accuracy and inefficient because of fatigues, tiredness, and lack of interpretation of disease. Thus, image processing-based automatic machine learning and deep learning methods are used for plant leaf disease detection techniques (Chapaneri et al., 2020)(Joshi et.al 2020)(Kaur et al 2019). Machine learning models can be used for thing identification, review, prediction, categorization, and grouping of object. The learning task in ML categorized as “Supervised”, “unsupervised” and “semi-supervised”. Depending upon the problem statement different models is used. Deep neural networks are advances of the neural network that have been successfully applied recently for many computer vision-based applications. Deep neural networks are constructed by stacking the series of layers of nodes. The deep learning algorithms' performance can be improved by tuning the parameters of the deep learning layers. The accuracy of deep learning models depends on the size of the database.

II. EXISTING SYSTEM

- It is advantageous to diagnose plant diseases (Black Spot, other leaf spots, powdery mildew, downy mildew, blight, and canker) using an automated method since it lessens the amount of manpower required to maintain mega farms of crops and does so at an incredibly preliminary phase(when they appear on plant leaves).
- The detection of leaf diseases utilizing imagery methods and DML approaches is the focus of this review of the literature.
- By obtaining understanding and expertise through straightforward approximations, ML and DL techniques are able to handle complex issues.
- The capacity to create accurate renderings is the crucial component that has led to the extensive use of neural network and ML techniques.
- Machine learning classifier requires the utilization of features, which is a challenging procedure that has a significant impact on classifier.
- In fields including medicine, agriculture, and computer vision, DL techniques are often used to classify images.

Disadvantages

- Large amount of leaf disease dataset for training deep machine learning models.
- Presence of intra-class leaf-diseases variation.
- Diseased leaf image variation in scale, perspective, illumination, Clutter in the background

III. PROPOSED SYSTEM

- Our proposed architecture takes leaf images as input and outputs the class labels. At first, the input image is passed through a pre processing step where it is enhanced using Adaptive Histogram Equalization.
- Then, the enhanced image is fed to a transfer learning block, where we utilize a pretrained deep CNN model for efficient feature extraction.
- A more general crop disease identification algorithm, which can be applied to other vegetable and fruit image disease data sets, and at the same time provide a reference for the prevention and control of diseases.
- Compared with the pre-trained neural network model, this model achieves high accuracy of leaf or plant health/diseases image classification.
- For methods that use machine learning or pre-training models, they manually or directly extract features of disease images.

- Because these models are difficult to accurately locate high-informative disease regions, it often affects the classification effect. The development of new or improved pre-trained models can achieve better classification results.
- Using this method, we classify the diseases in different leaves like Grape, Tomato and Almond.
- Diseases that are classified through Grape leaves are Downey Mildew, Powdery Mildew and Black Rot.
- Diseases that are classified through Tomato leaves are Early Blight, Septoria spot and Late Blight.
- Diseases that are classified through Almond leaves are Leaf Scorch, Alternaria, Leaf Spot

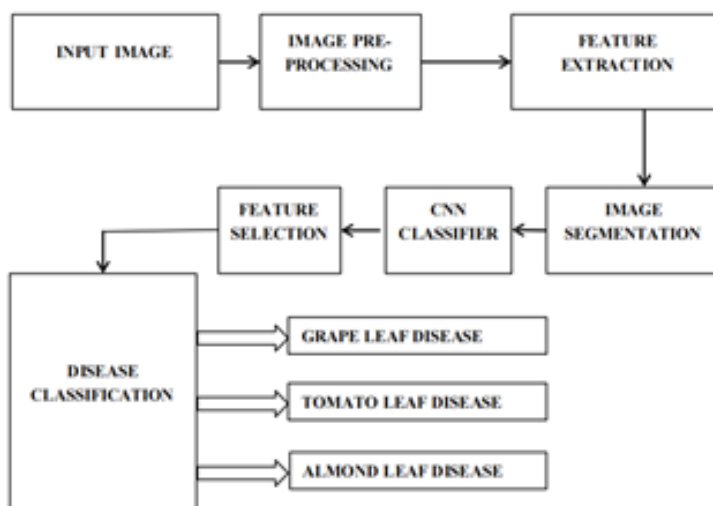


Figure 1:Proposed Block

IV. SOFTWARE IMPLEMENTATION

1. MATLAB

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java. You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing. The matrix-based MATLAB language is the world's most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together. MATLAB helps you take your ideas beyond the desktop. You can run your analyses on larger data sets, and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling you to deploy algorithms and applications within web, enterprise, and production systems. MATLAB tutorial provides excellent insight into the core fundamentals of MATLAB. By learning the core concepts of MATLAB, a core learner can go further to our advance MATLAB tutorial in the path of learning the MATLAB. In this MATLAB tutorial, we will start with the MATLAB as an environment and keep on studying with the MATLAB programming language. We will cover entire topics of MATLAB journey; from downloading & installing of the

MATLAB environment, to its features & capabilities, to the basics programming fundamentals, and to the advance object-orientation of the MATLAB programming language We will cover the areas where MATLAB is being used and what are its prospects. We will show you the strength of MATLAB in numerical computing, as to how MATLAB is so powerful in computing, plotting, algorithmic designs, and manipulating data. We will see MATLAB's rich add on toolboxes that are used to build future Iready application. **MATLAB** is a mathematical software package which can be used for analysing and solving mathematical and engineering problems such as numerical computations, graphs plotting, programming and more. MATLAB treat all the variables in form of scalar , vector or matrices form. For this book of **MATLAB Programming**, it is aimed to give a new MALAB users have an overview of how to use the basic and intermediate level of MATLAB and of course to keep this free for students who might be using this book for their course. To start off, MATLAB is a scripting language (Note that I didn't mentioned it is programming language) for data analysis, think of it as a super-powered programmable calculator with useful default functions that are built specifically for math applications. The name MATLAB itself is an abbreviation for "**MAT**rix **LAB**oratory". Originally, it was a simple interactive matrix calculator in early 1980's. Gradually, it evolves into programming language developed by MathWorks. It started out as a matrix programming language where linear algebra programming was simple.

- **MATLAB's Application programming interface (API):** MATLAB consists of an extensive API. Through this API, we can link our C/C++ programs directly to MATLAB. Some options available in MATLAB API are calling MATLAB programs, read and write M-files, and using MATLAB as an interface to run applications. MATLAB can be used both as a computation and analysis tool.
- **Machine Learning, Deep Learning, and Computer vision:** The most demanding technologies like Machine learning, Deep learning, and Computer vision can be done in MATLAB. We can create and interconnect layers of a deep neural network, We can build custom training loops and training layers with automatic differentiation. For machine learning, we can use the DBSCAN algorithm to discover clusters and noise in DATA. For computer vision, we can do object tracking, object recognition,gesture recognition, and processing 3D point clouds.
- **Computational Biology toolbox:** This toolbox provides a great way for biologists and researchers to create and analyze new algorithms and patterns for development in biological and biochemical domains. We can build biological models and analyze them using this toolbox. Moreover, for students, this toolbox can be very much educational if they want to explore the biological domain

Data Analysis and Visualization

MATLAB provides tools to acquire, analyze, and visualize data, enabling you to gain insight into your data in a fraction of the time it would take using spreadsheets or traditional programming languages. You can also document and share your results through plots and reports or as published MATLAB code. Data visualization is the process of translating data into graphical representations like plots, charts, maps, and 3D visualizations that help you easily identify patterns, trends, and outliers in the data. These data visualizations let you see relationships that would be difficult or impossible to observe merely by looking at the raw data, especially with big data sets from sources such as sensors, data loggers, medical records, web search patterns, and purchasing patterns. Data visualization plays a pivotal role in converting data into actionable information. Using historical or live market data, data visualization helps to quickly identify patterns and trends, detect anomalies, and get meaningful insights. Data visualization helps in performing analysis, developing predictive models, assessing risk, and formalizing trading strategies. Use the MATLAB Visualizations app to visualize data in a ThingSpeak™ channel. You can view and explore data using interactive visualizations such as an area plot, line plot, or scatter plot in static visualizations using other MATLAB plots. You can also make visualizations public and use the URL to embed them on websites. Thing Speak also allows you to use MATLAB to analyze your data. The MATLAB Analysis and MATLAB Visualizations apps provide code templates to assist you with basic operations on your historic or live data. Use code template tutorials to get started with Thing Speak. Use the apps to convert from one unit to another, compare different or similar data on the same plot, or visualize the statistical distribution of your data. You can Add Functions to Scripts

(MATLAB) in MATLAB Analysis and Visualization apps to allow for modular coding. In addition to the built-in MATLAB functions, you can also use functions from the toolboxes mentioned in Access MATLAB Add-On Toolboxes.

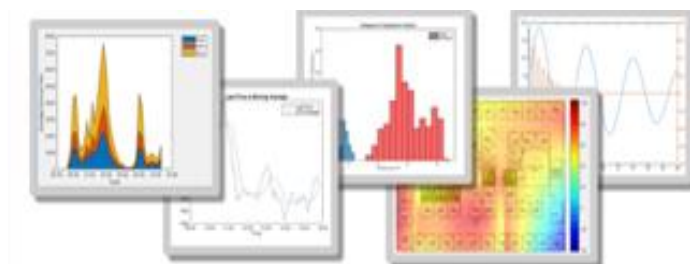


Figure 2: Data visualization

Acquiring Data

MATLAB lets you access data from files, other applications, databases, and external devices. You can read data from popular file formats such as Microsoft Excel; text or binary files; image, sound, and video files; and scientific files such as netCDF and HDF. File I/O functions let you work with data files in any format. Using MATLAB with add-on products, you can acquire data from hardware devices, such as your computer's serial port or sound card, as well as stream live, measured data directly into MATLAB for analysis and visualization. You can also communicate with instruments such as oscilloscopes, function generators, and signal analyzers. Data Acquisition Toolbox™ provides apps and functions for configuring data acquisition hardware, reading data into MATLAB® and Simulink®, and writing data to DAQ analog and digital output channels. The toolbox supports a variety of DAQ hardware, including USB, PCI, PCI Express®, PXI®, and PXI Express® devices, from National Instruments® and other vendors. The toolbox apps let you interactively configure and run a data acquisition session. You can then generate equivalent MATLAB code to automate your acquisition in future sessions. Toolbox functions give you the flexibility to control the analog input, analog output, counter/timer, and digital I/O subsystems of a DAQ device. You can access device-specific features and synchronize data acquired from multiple devices. You can analyze data as you acquire it or save it for postprocessing. You can also automate tests and make iterative updates to your test setup based on analysis results.

Analyzing Data

MATLAB lets you manage, filter, and pre-process your data. You can perform exploratory data analysis to uncover trends, test assumptions, and build descriptive models. MATLAB provides functions for filtering and smoothing interpolation, convolution, and fast Fourier transforms (FFTs). computes values necessary to evaluate the architecture using the properties of each element in the model instance. Add-on products provide capabilities for curve and surface fitting, multivariate statistics, spectral analysis, image analysis, system identification, and other analysis tasks. Use the MATLAB Analysis app to prepare, filter, and analyze data such as calculating average humidity, calculating dew point, and eliminating data outliers from a ThingSpeak™ channel. ThingSpeak allows you to use MATLAB to analyze and visualize your data. The analysis and visualization apps provide template code to assist you with basic operations on historic or live data. These tutorials take you step by step through each code template. Use code template tutorials to get started with ThingSpeak. Use the apps to convert from one unit to another, compare different or similar data on the same plot, or visualize the statistical distribution of your data. You can Add Functions to Scripts (MATLAB) in the MATLAB Analysis and MATLAB Visualizations apps to allow for modular coding. In addition to the built-in MATLAB functions, you can also use functions from the toolboxes mentioned in Access MATLAB Add-On Toolboxes. Use an analysis function to calculate the result of an analysis. For more information, see Analysis Function Constructs. You can use the datastore to manage incremental import of the data. To analyze the data

using common MATLAB functions, such as mean and histogram, create a tall array on top of the datastore. For more complex problems, you can write a MapReduce algorithm that defines the chunking and reduction of the data. Data Analysis. Data Analysis is the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data. MATLAB offers a wide range of built-in chart types like line plots, scatter charts, distribution plots, and geographic plots to visualize datasets from a diverse set of applications. You can create visualizations either interactively or programmatically using the MATLAB language.

Visualizing Data

MATLAB provides built-in 2-D and 3-D plotting functions, as well as volume visualization functions. You can use these functions to visualize and understand data and communicate results. Plots can be customized either interactively or programmatically. The MATLAB plot gallery provides examples of many ways to display data graphically in MATLAB. For each example, you can view and download source code to use in your MATLAB application

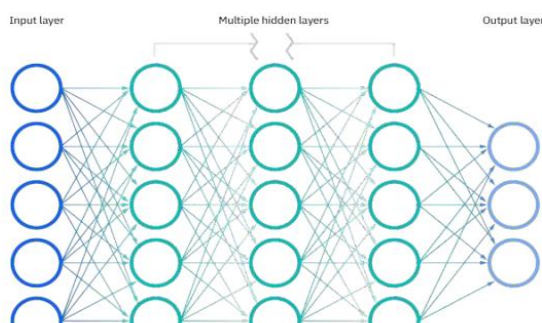


Figure 3: Neural networks

Understanding Neural Networks

Neural networks, in the world of finance, assist in the development of such processes as time-series forecasting, algorithmic trading, securities classification, credit risk modeling, and constructing proprietary indicators and price derivatives. A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.³⁶ A neural network contains layers of interconnected nodes. Each node is known as a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear. Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google's search algorithm

CNN architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

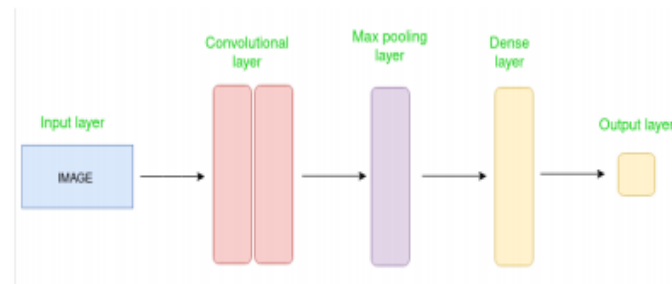


Figure 4: Simple CNN architecture

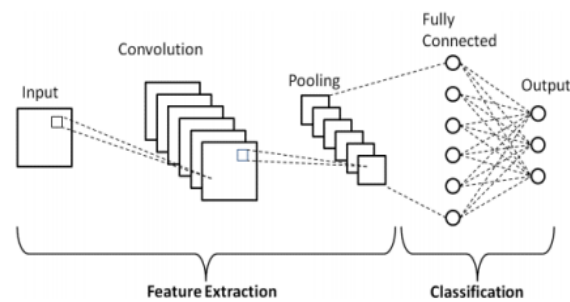
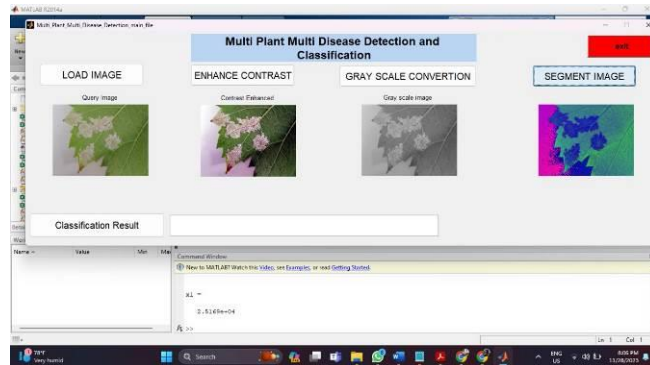
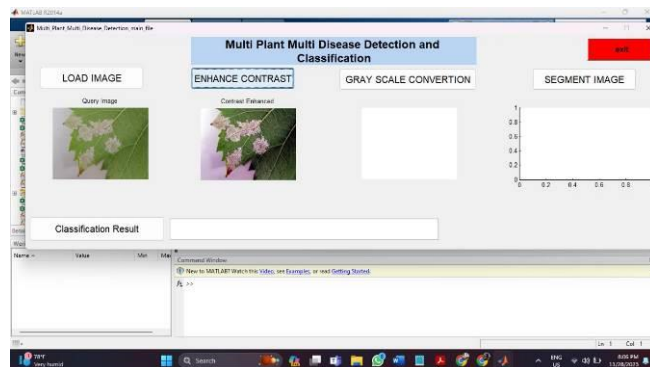
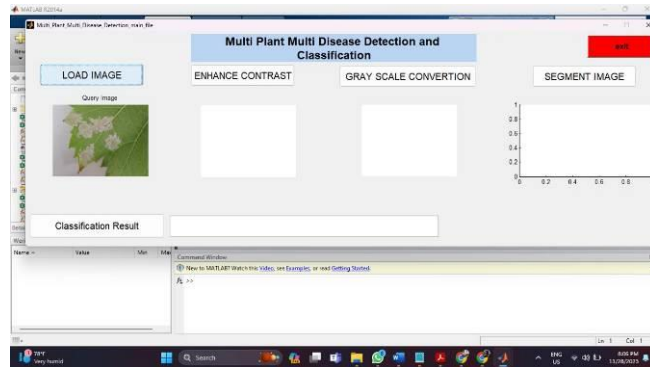


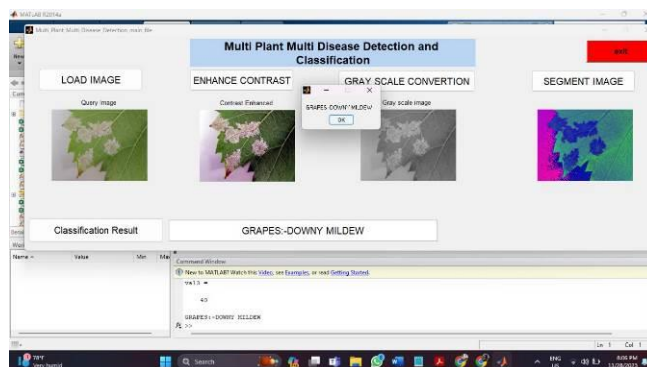
Figure 5

Motivation behind Convolution

Convolution leverages three important ideas that motivated computer vision researchers: sparse interaction, parameter sharing, and equivariant representation. Let's describe each one of them in detail. Trivial neural network layers use matrix multiplication by a matrix of parameters describing the interaction between the input and output unit. This means that every output unit interacts with every input unit. However, convolution neural networks have sparse interaction. This is achieved by making kernel smaller than the input e.g., an image can have millions or thousands of pixels, but while processing it using kernel we can detect meaningful information that is of tens or hundreds of pixels. This means that we need to store fewer parameters that not only reduces the memory requirement of the model but also improves the statistical efficiency of the model. If computing one feature at a spatial point (x_1, y_1) is useful then it should also be useful at some other spatial point say (x_2, y_2) . It means that for a single two-dimensional slice i.e., for creating one activation map, neurons are constrained to use the same set of weights. In a traditional neural network, each element of the weight matrix is used once and then never revisited, while convolution network has shared parameters i.e., for getting output, weights applied to one input are the same as the weight applied elsewhere. Due to parameter sharing, the layers of convolution neural network will have a property of equivariance to translation. It says that if we changed the input in a way, the output will also get changed in the same way.

V. OUTPUT





VI. CONCLUSION

The deep learning techniques are playing a significant role in the agriculture industry and other industries. This algorithm is easy to implement, we can solve complex problems using these techniques. Using image techniques and algorithms we successfully identified the affected area in the plant leaf. Various features of the image are extracted with their numeric values. The algorithm used here is very much efficient and best case time space complexities are achieved. For ninety percentage of images average clustering and processing time is less than minimum seconds.

REFERENCES

- U. Shruthi, V. Nagaveni, B.K. Raghavendra, A review on machine learning classification techniques for plant disease detection, in: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), IEEE, 2019, March, pp. 281–284.
- Singh AK, Ganapathysubramanian B, Sarkar S, et al. Deep learning for plant stress phenotyping: trends and future perspectives. *Trends Plant Sci.* 2018;23(10):883–898.
- Pandey AK, Burlakoti RR, Kenyon L, et al. Perspectives and challenges for sustainable management of fungal diseases of mung bean [*Vigna radiata* (L.) R. Wilczek var. *radiata*]: a review. *Front Environ Sci.* 2018;6:53
- Y. H. Bhosale and K. S. Patnaik, "IoT Deployable Lightweight Deep Learning Application For COVID-19 Detection With Lung Diseases Using RaspberryPi," 2022 International Conference on IoT and Blockchain Technology (ICIBT), 2022, pp. 1-6.
- Yogesh H. Bhosale, K. S. Patnaik, PuDi-COVID: Chronic obstructive pulmonary (lung) diseases with COVID-19 classification using ensemble deep convolutional neural network from chest X-ray images to minimize severity and mortality rates, *Biomedical Signal Processing and Control*, Volume 81, 2023, 104445, ISSN 1746-8094.
- Y.H.Bhosale, et al, "Human Monkeypox 2022 Virus: Machine Learning Prediction Model, Outbreak Forecasting, Visualization with Time-Series Exploratory Data Analysis," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2022, pp. 1-6.56
- Bhosale, Y.H., Patnaik, K.S. Application of Deep Learning Techniques in Diagnosis of Covid-19 (Coronavirus): A Systematic Review. *Neural Process Lett* (2022). <https://doi.org/10.1007/s11063-022-11023-0>.
- Ümit Atila, Murat Uçar, Kemal Akyol, Emine Uçar, Plant leaf disease classification using Efficient Net deep learning model, *Ecological Informatics*, Volume 61, 2021, 101182, ISSN 1574-9541.
- Singh, R.K., Tiwari, A. & Gupta, R.K. Deep transfer modeling for classification of Maize Plant Leaf Disease. *Multi med Tools Appl* 81, 6051– 6067 (2022).
- Vallabhajosyula, S., Sistla, V. & Kolli, V.K.K. Transfer learning-based deep ensemble neural network for plant leaf disease detection. *J Plant Dis Prot* 129, 545–558 (2022).
- Kaur, P., Harnal, S., Gautam, V. et al. A novel transfer deep learning method for detection and classification of plant leaf disease. *J Ambient Intell Human Comput* (2022).

- Gajjar, R., Gajjar, N., Thakor, V.J. et al. Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. *Vis Comput* 38, 2923–2938 (2022).
- G. Geetha, S. Samundeswari, G. Saranya, K. Meenakshi and M. Nithya, 2020, Plant Leaf Disease Classification and Detection System Using Machine Learning, *J. Phys.: Conf. Ser.* 1712 012012.
- Chouhan, S.S., Singh, U.P. & Jain, S. Automated Plant Leaf Disease Detection and Classification Using Fuzzy Based Function Network. *Wireless Pers Commun* 121, 1757–1779 (2021)
- R. Sujatha, Jyotir Moy Chatterjee, NZ Jhanjhi, Sarfraz Nawaz Brohi, Performance of deep learning vs machine learning in plant leaf disease⁵⁷ detection, *Microprocessors and Microsystems*, Volume 80, 2021, 103615, ISSN 0141-9331.