DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO - FUZZY NETWORKS USING DEEP LEARNING

A Project Report submitted in the partial fulfilment of the Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

CSE (ARTIFICIAL INTELLIGENCE)

Submitted by

D. HARSHITH (20471A4310)

N. JANI BASHA (20471A4340)

P. VISHNU VARDHAN (20471A4342)

Under the esteemed guidance of

Mr. B. VEERA BRAHMAM M. Tech

Asst. Professor

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Cycle -1 NIRF rank in the band of 251-320 and an ISO 9001:2015 Certified Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2023 - 2024

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

CERTIFICATE

This is to certify that the project entitled **"DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO - FUZZY NETWORKS USING DEEP LEARNING"** is a bonafide work done by **"D. Harshith (20471A4310), N. Jani basha (20471A4340), P. Vishnu Vardhan (20471A4342)"** in partial fulfilment of the requirements for the award of the degree of the **BACHELOR OF TECHNOLOGY** in the Department of **CSE (ARTIFICIAL INTELLIGENCE)** during 2023 - 2024.

PROJECT GUIDE

Mr. B. Veera Brahmam M. Tech

PROJECT COORDINATOR

Mr. K. Jaya Prakash M. Tech (Ph. D.)

HEAD OF THE DEPARTMENT

Dr. B. Jhansi Vazram M. Tech, Ph. D.

EXTERNAL EXAMINER

Viva Voice Date: ……

DECLARATION

We hereby declare that the work described in this project work, entitled **" DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING "** which is submitted by us in partial fulfilment for the award of **Bachelor of Technology** in the Department of **CSE (Artificial Intelligence)** to the **Narasaraopeta Engineering College**, is the result of work done by us under the guidance of Mr. **B. Veera Brahmam**, Asst. Professor, Dept of IT & CSE(AI)

The work is original and has not been submitted for any Degree/ Diploma of this or any other university.

D. HARSHITH **(20471A4342)**

N. JANI BASHA **(20471A4340)**

P. VISHNU VARDHAN **(20471A4342)**

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairperson **Sri M. V. Koteswara Rao** B.Sc., who took keen interest on us in every effort throughout this course. We owe our gratitude to our principal **Dr. M. Sreenivasa Kumar** M. Tech., Ph.D., MISTE, FIE, for his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude to **Dr. B. Jhansi Vazram** M. Tech, Ph.D., Professor & Head, Department of IT & CSE (AI) and also to Project Coordinator **Mr. K. Jaya Prakash** M. Tech, (Ph.D.), **Asst. Prof,** Department of IT & CSE (AI) whose unstinting encouragement enabled us to accomplish our project successfully and in time.

We extend our sincere thanks to our guide, **Mr. B. Veera Brahmam** M. Tech, **Asst. Prof,** Department of IT & CSE (AI), for extending his encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B. Tech degree. we have no words to acknowledge the warm affection, constant inspiration, and encouragement that we receive from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts, which had really helped us in successfully completing our project.

By

D. Harshit (20471A4310)

N. Janibasha (20471A4340)

P. Vishnu Vardhan (20471A4342)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills.

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

VISION OF THE DEPARTMENT

To be renowned department that imparts creative, learning and research skills to students in the domain of artificial intelligence.

MISSION OF THE DEPARTMENT

To Impart Strong foundation of statistics for understanding Artificial Intelligence.

M1: To establish high performance computational facilities and tools to develop innovative and intelligent solutions.

M2: To collaborate with renowned companies for multidisciplinary research and development.

M3: To guide the students in learning and creative for developing intelligent technology-based solutions to societal problems.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Ability to analyse and apply the knowledge of human cognition, Artificial Intelligence, Machine Learning and data engineering in terms of real world problems to meet the challenges of the future.

PSO2: Ability to develop computational knowledge and project development skills using innovative tools and techniques to solve problems in the areas related to Deep Learning, Machine learning, Artificial Intelligence.

PSO3: Ability to lead a product development company/team and use the acquired knowledge to identify real-world research problems.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: To Formulate, analyze and solve Engineering problems with strong foundation in Mathematical, Scientific, Engineering fundamentals and modern computing practices through advanced curriculum.

PEO2: Analyze the requirements, realize the technical specification and design the Engineering solutions by applying artificial intelligence theory and principles.

PEO3: Demonstrate technical skills, competency in AI and promote collaborative learning and teamwork spirit through multi -disciplinary projects and diverse professional activities.

PEO4: Equip the graduates with strong knowledge, competence and soft skills that allows them to contribute ethically to the needs of society and accomplish sustainable progress in the emerging computing technologies through life-long learning.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics,natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and needfor sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and writeeffective reports and design documentation, make effective presentations, and give and receiveclear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member andleader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO's):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project.

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		✓											✓		
C421.2	✓		✓		✓								✓		
C421.3				✓		✓	✓	✓					✓		
C421.4				✓		✓	✓	✓					✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6				✓		✓	✓	✓					✓		

Course Outcomes – Program Outcomes Correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2										2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: **The values in the above table represent the level of correlation between CO's and PO's:**

1. **Low level**
2. **Medium level**
3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the Model	Attained PO
CC2204.2	Gathering the requirements and defining the problem, plan to develop a smart model for predicting Plant Diseases.	PO1, PO3
CC421.1, CC2204.3	Each and every requirement is critically analyzed and the process model is identified.	PO2, PO3
CC421.2, CC2204.2	Logical design is done by using the unified modelling language which involves individual teamwork.	PO3, PO5, PO9
CC421.3, C2204.3	Each and every model is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4	Documentation is done by all our three members in the form of a group.	PO10
CC421.5, CC2204.2	Each and every phase of the work in group is presented periodically.	PO10, PO11
CC2202.2, CC2203.3, CC1206.3, CC3204.3, CC4104.2	Implementation is done and the project is deployed by using a front end.	PO4, PO7
CC32SC4.3	The design includes models like CNNs, Fuzzy Logic Systems, Neuro – Fuzzy Systems, Supervised Learning.	PO5, PO6

ABSTRACT

Significant risks to agricultural production come from plant diseases, which have an effect on food yields and the stability of the economy. Conventional illness detection techniques that depend on human observation are frequently expensive, time-consuming, and prone to errors. On the other hand, automatic detection that makes use of image processing techniques provides accurate and timely results, signaling a potentially fruitful path towards improving plant protection in agriculture.

This research uses a Deep Convolution Neuro-Fuzzy Network (DCNFN) to provide a novel method for disease recognition in tomato, pepper bell, and potato plants. Our methodology, which makes use of recent developments in computer vision, intends to transform precision agriculture by providing a powerful tool for the detection, evaluation, and classification of plant diseases.

The creation of a deep learning framework for the purpose of training and optimizing the DCNFN model, the establishment of a comprehensive collection of plant photos, and the professional evaluation of disease signs are crucial phases in our implementation. One notable aspect of our model is its simplicity: background photos and healthy leaves are smoothly incorporated into the training dataset, making it possible to discern between healthy and diseased foliage.

Our findings open the door for the DCNFN model's practical application in agricultural settings by demonstrating its effectiveness in precisely diagnosing and categorizing plant diseases across a wide range of species. Our technique has great potential to improve crop resilience, optimize resource allocation, and ultimately protect global food security, as it provides a comprehensive solution for disease detection and risk assessment.

INDEX

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction

Agriculture plants are the important plants that provides various proteins and vitamins for billions of people all over the globe [1]. It is the principal source of nutrition in the countries of India, and Indonesia. Anything that affects the plant crop, both in terms of superiority and measure, has a global impact. As a result, regular illness monitoring and identification are very necessary in order to effectively combat the problem. Diseases may have a severe impact on production if they are not treated in a timely manner [2]. A multitude of diseases and pests have contributed to the degradation of agriculture crops in recent years. Without proper surveillance and management, these attacks may cause severe disruption on farms, and the problem has only gotten worse in recent years. In addition, the early and accurate identification of problems in plants saves plants from dangerous diseases and significantly lessens the amount of money lost. The treatment of these disorders is supportive of our efforts to assure healthy agriculture [1].

The ability to automatically identify plant illnesses based on plant leaves is a major step forward in the agricultural sector. The efficiency and quality of harvests may be improved via early and correct diagnosis of plant leaf diseases. In order to maximize crop superiority, it is crucial to prevent the spread of disease throughout the farm. Diseases may be contained if their symptoms are recognized, their causes for proliferation are investigated, and control measures are put into place. Plant diseases may cause problems for plants at all stages of their life cycle.

Plants that are unwell will often have discoloured, wilted leaves. Generally speaking, it is possible to recognize abnormalities in the visual presentation of a disease by looking for its characteristic pattern. The leaves of a plant are usually the first to show signs of disease, making them an excellent resource for diagnosing plant ailments [3]. Semantic segmentation and picture identification are only two areas where deep learning techniques have been effectively used recently. As an added bonus, this technique has been used to diagnose a wide range of plant diseases [4-6]. When discussing "Deep Learning," the term "Deep" denotes to the number of transformational layers over which data is passed, each of which extracts increasingly subtle features from the input. Layers of nodes make up the Deep Learning algorithm, which may be broken down into the input layer, the hidden layer, and the output layer. By learning additional complex data properties of the leaf, it performs an excellent classification procedure. The raw picture data serves as input to the convolutional neural network, and the model's learned

classification serves as an output, making the neural network a full-stack solution. In the center, where learning occurs, neither human input nor well-crafted algorithms are necessary. However, this network has significant difficulties when dealing with ambiguous data. Some aspects of the training set, for instance, are often ambiguous or even noisy [7-9]. The incapacity of this neural network to produce fuzzy values for the feature map pixels has a detrimental effect on the detection accuracy of tomato leaf diseases. [10].

In order to interpret hazy, ambiguous, and erroneous data input, models might benefit from the utilization of fuzzy sets and fuzzy reasoning rules [11]. Human language is able to be used to represent ambiguous and hazy information thanks to the invention of fuzzy sets. As a measure of how fuzzy and uncertain a set is, the membership degree may be calculated using the fuzzy set concept's membership function. The goal of using fuzzy inference rules is to retrieve information that is ambiguous, fuzzy, or imprecise. To achieve this, image processing algorithms that mimic human decision-making have been included.

As digital image processing and identification technology advances, it is now feasible to quickly and easily identify ailing crops and classify the specific disease that has afflicted them [12,13]. However, AI and ML alone will not be adequate; thus, some academics have proposed using surveillance drones, the Internet of Things, and cloud services, among other things, to create a holistic system that can effectively aid farmers in getting good outcomes and decreasing expenses [14]. Nevertheless, the most important element would be a machine learning algorithm, method, or process that is both extremely effective and capable of detecting and precisely diagnosing plant illness. As a result, researchers are still looking for the most effective machine learning solution for the diagnostics of plant diseases. Although a recent study has been conducted in this area, the ideal and most precise answer is still an open research issue, and scholars are continually working near achieving this objective.

Jang et al. [15] introduced a neuro-fuzzy inference system adapting to new information, using a neural network-based mechanism for TSK fuzzy model parameter calculation. This method, a common neuro-fuzzy modelling approach, is somewhat simplistic, focusing on a single rule set. We propose employing a deep neuro-fuzzy network for detecting illnesses in tomato leaves. This network integrates fuzzy systems and deep learning, incorporating fuzzy implication rules in a deep framework. The novel fuzzy inference and fuzzy pooling layers enhance information extraction from tomato leaf images, offering both clear and fuzzy values for higher recognition

accuracy. The end-to-end network, with multiple hidden layers, ensures better performance and generalization in processing tomato leaf images and extracting relevant information.

1.2 Existing System

Nowadays, decisions are made by plant experts based on their experience and knowledge. This may lead to errors, consume a lot of time and with more cost expenditure.

Disadvantages:

- Doesn't generate accurate and efficient results.
- Lack of accuracy may cause to inappropriate results (or) outputs.
- Computation time is very high.
- Experts charges more cost.
- Consumes more man power
- Risky process for the farmer

1.3 Proposed System

We proposed to develop a system which will help framers itself to find or predict the plant disease with the help of leaf of a plant based on some attributes like mycorrhizae form a mutualistic relationship with host plant root systems. On the other hand, pathogenic fungi cause plant diseases such as anthracnose, leaf spot, rust, wilt, blight, coils, scab, gall, canker, damping-off, root rot, mildew, and dieback. So, there is a need for developing a system which will help farmers itself to understand the disease and then system suggests a medicines according to the plant disease. The Machine learning algorithms like Fuzzy logic, Productivity, Crops, Sensitivity and specificity, Feature extraction, Fuzzy neural networks have proven to be most accurate & reliable and hence, used in this project.

Advantages:

- Plant disease detection benefits.
- Automated disease detection.
- Optimized resource allocation.
- Money saving
- Time saving
- Convenience
- Immediacy

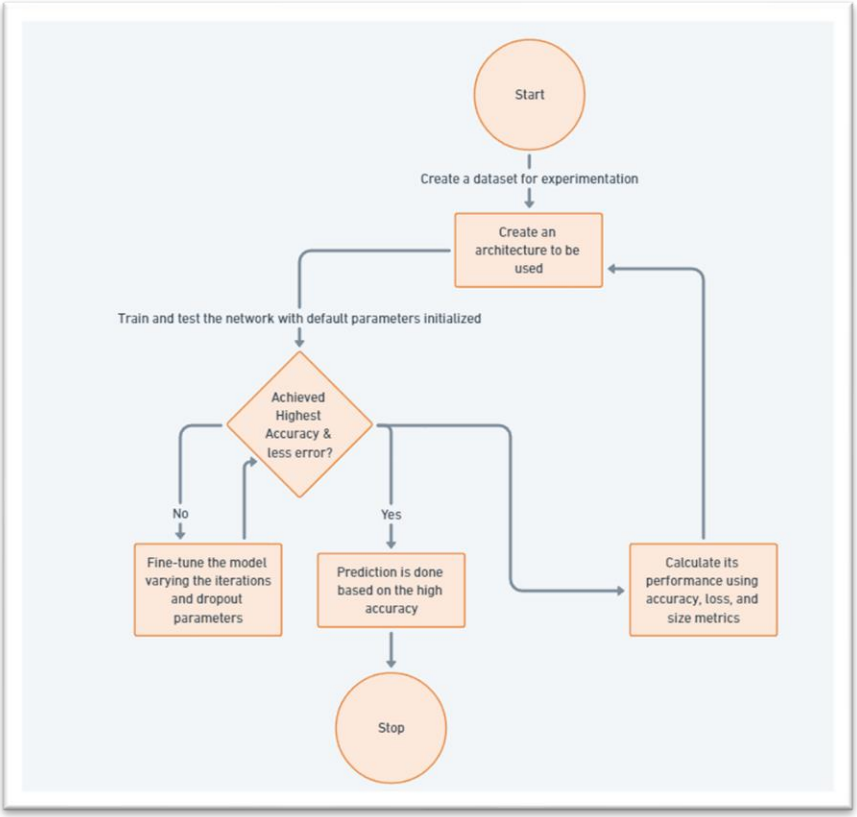

Figure: 1.1 Proposed System

1.4 System Requirements

1.4.1 Hardware Requirements:

- System Type : Intel Corei5 or Above
- Hard – Disk : 20GB or Above
- RAM : 2GB or Above

1.4.2 Software Requirements:

- Operating System : Windows 7 or Above
- Coding Language : Python
- Python Distribution : Anaconda, PyCharm

2. LITERATURE SURVEY

2.1 Deep Learning

Deep learning is a subfield of machine learning that focuses on the use of artificial neural networks to model and solve complex problems. What distinguishes deep learning from traditional machine learning is the use of deep neural networks, which consist of multiple layers (deep architectures). These layers enable the network to automatically learn hierarchical representations of data, capturing intricate patterns and features at various levels of abstraction.

At the core of deep learning are artificial neural networks, inspired by the structure and function of the human brain. These networks consist of interconnected nodes, or neurons, organized into layers. The input layer receives data, and the output layer produces the desired predictions or classifications. Between the input and output layers are hidden layers, where the network learns to extract features from the input data through iterative training processes.

Deep learning has demonstrated remarkable success in a wide range of applications. Convolutional Neural Networks (CNNs) are commonly used for image and video analysis, excelling in tasks like image recognition and object detection. Recurrent Neural Networks (RNNs) are effective for sequential data, making them suitable for natural language processing and time-series prediction. Additionally, architectures like Long Short-Term Memory (LSTM) networks address challenges related to capturing long-term dependencies in sequential data.

Training deep neural networks typically involves a process called backpropagation, where the network adjusts its internal parameters based on the error between predicted and actual outcomes. The availability of large labeled datasets and advances in computational power, especially the use of Graphics Processing Units (GPUs), has facilitated the training of deeper and more complex neural networks. As a result, deep learning has become a dominant approach in various fields, including computer vision, speech recognition, natural language processing, and autonomous systems. Moreover, efforts in model interpretability and explainability aim to make deep learning systems more transparent and understandable for practical applications.

Despite its success, deep learning also poses challenges, such as the need for substantial amounts of labeled data and computational resources, and ongoing research seeks to address.

2.2 Deep Learning Methods

Deep learning encompasses a variety of methods and architectures designed to model complex patterns and representations in data using deep neural networks. Here are some key deep learning methods:

Convolutional Neural Networks (CNNs):

- Description: CNNs are particularly effective for image and video analysis. They use convolutional layers to automatically learn spatial hierarchies of features in an image. This makes them well-suited for tasks such as image recognition, object detection, and image segmentation.

Recurrent Neural Networks (RNNs):

- Description: RNNs are designed for sequential data and have connections that form directed cycles. They are effective in capturing dependencies over time, making them suitable for tasks like natural language processing (NLP), speech recognition, and time-series analysis.

Long Short-Term Memory Networks (LSTMs):

- Description: LSTMs are a type of RNN designed to address the vanishing gradient problem, which can occur during the training of traditional RNNs. LSTMs use memory cells with gating mechanisms, allowing them to capture long-term dependencies in sequential data. They are commonly used in applications involving sequences, such as language modeling and speech recognition.

Autoencoders:

- Description: Autoencoders are neural networks designed for unsupervised learning and dimensionality reduction. They consist of an encoder and a decoder, with the objective of reconstructing the input data. Autoencoders are used for tasks such as data denoising, feature learning, and anomaly detection.

Generative Adversarial Networks (GANs):

- Description: GANs consist of two neural networks, a generator and a discriminator, which are trained simultaneously through adversarial training. The generator aims to create realistic data, while the discriminator tries to distinguish between real and generated data.

Transformers:

- Description: Transformers have gained prominence in natural language processing tasks. They use self-attention mechanisms to capture contextual relationships between words in a sequence. BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) are examples of successful transformer-based models.

Deep Reinforcement Learning:

- Description: Deep reinforcement learning combines deep neural networks with reinforcement learning principles. Agents learn to make decisions by interacting with an environment and receiving feedback in the form of rewards. Deep Q Networks (DQN) and Proximal Policy Optimization (PPO) are popular algorithms in this category, used in game playing, robotics, and autonomous systems.

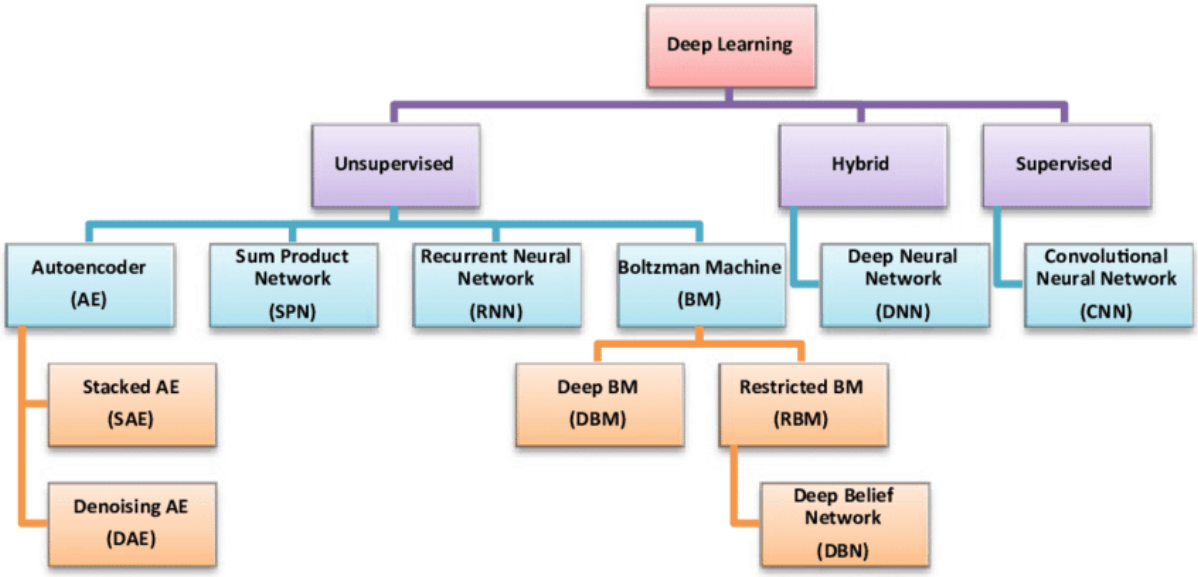

Figure: 2.1 Deep Learning Categories and Algorithms

2.3 Application of Deep Learning

A. HealthCare
B. Image Recognition
C. Stock Analysis
D. New Analysis
E. Fraud Detection
F. Self-Driving Cars

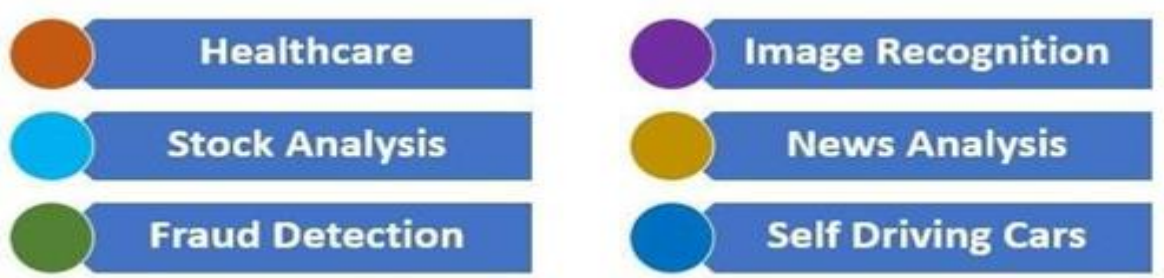

Figure: 2.2 Applications of Deep Learning

2.4 Characteristics of Deep Learning

Supervised, Semi-Supervised or Unsupervised When the category labels are present while you train the data then it is Supervised learning. Algorithms like Linear regression. Logistic regression, decision trees use Supervised Learning. When category labels are not known while you train data then it is unsupervised learning. Algorithms like Cluster Analysis, K means clustering, Anomaly detection uses Unsupervised Learning. The data set consists of both labelled and unlabelled data then we call it is Semi Supervised learning. Graph-based models, Generative models, cluster assumption, continuity assumption use Semi-Supervised learning.

A. **Huge Number of Resources** it needs advanced Graphical Processing Units for processing heavy workloads. A huge amount of data needs to be processed like big data in the form of structured or unstructured data. Sometimes more time also required to process the data, it depends on the amount of data fed in.
B. **Large Number of Layers in Model** a huge number of layers like input, activation, the output will be required, sometimes the output of one layer can be input to another layer by making few small findings and then these findings are summed up finally in the SoftMax layer to find out a broader classification for final output.

C. **Optimizing Hyper-parameter** hyperparameters like no of epochs, Batch size, No.of layers, Learning rate, needs to be tuned well for successful Model accuracy because it creates a link between layer predictions to final output prediction. Over-fitting and under-fitting can be well handled with hyper-parameters.

D. **Cost Function** it says how well the model performance in prediction and accuracy. For each iteration in Deep Learning Model, the goal is to minimize the cost when compared to previous iterations. Mean absolute error, Mean Squared Error, Hinge loss, Cross entropy are different types according to different algorithms used.

2.5 Advantages of Deep Learning

- Solve Complex problems like Audio processing in Amazon echo, Image recognition, etc, reduce the need for feature extraction, automated tasks wherein predictions can be done in less time using Keras and Tensorflow.
- Parallel computing can be done thus reducing overheads.
- Models can be trained on a huge amount of data and the model gets better with more data.
- High-Quality Predictions when compared with humans by training tirelessly.
- Works well-unstructured data like video clips, documents, sensor data, webcam data, etc.

2.6 Deep Learning Algorithms

To create a deep learning model, one must write several algorithms, blend them together and create a net of neurons. Deep learning has a high computational cost. To aid deep learning models, there are deep learning platforms like Tensor flow, Py-Torch, Chainer, Keras, etc. In deep learning, we have tried to replicate the human neural network with an artificial neural network; the human neuron is called perceptron in the deep learning model. We connect these perceptron units together to create a neural network; it has 3 sections:

A. Input layer
B. Hidden layer
C. Output layer

2.7 Architectural Methods for Deep Learning Algorithms

To build this architecture following algorithms are used:

Back Propagation

- In this algorithm, we calculate partial derivatives. In general, the gradient descent method for optimization, derivatives (gradients) are calculated at each iteration. In deep learning, functions are not simple; they are the composition of different functions.

Stochastic Gradient Descent

- In Gradient descent, the goal is to find global minima or optimum solution. But to get that, we have to consider local minima solutions (not desirable) also. If the objective function is a convex function, it is easy to find the global minima. The initial value for the function and learning rate are deciding parameters for finding global minima.

Learning Rate

- The learning rate is like the speed of the river; it can reduce training time and increase performance. In general, to learn any technique/sport, in the beginning, the learning rate is relatively high than at the end when one is to master it. After the intermediate stage, the learning will be slow; the focus will be on fine-tuning.

Batch Normalization

- In deep learning initial value of weight (randomly chosen) and learning rate is defined for a minibatch. In the beginning, there would be many outliers, and during backpropagation, these outliers must be compensated to compute the weights to get output. This compensation results in extra epochs. So, to avoid it, we use batch normalization.

Drop Out

- In deep learning, we generally encounter the problem of overfitting. Overfitting in large networks with several parameters makes it difficult to predict on test data. So, to avoid that, we use the dropout method, which drops random units during training by creating different thinned networks.

2.8 Occurrence of Plant Diseases

The occurrence of plant diseases is a common phenomenon that significantly impacts agricultural productivity and plant health. Plant diseases can manifest in various forms, including fungal infections, bacterial diseases, viral infections, and parasitic infestations. These diseases can affect different parts of the plant, such as leaves, stems, roots, and fruits, leading to symptoms such as wilting, discoloration, lesions, and deformities. The presence of pathogens in the environment, coupled with conducive conditions such as humidity, temperature, and poor soil health, creates an environment conducive to the spread and development of plant diseases.

Environmental factors, including climate variations and irregular precipitation, significantly influence plant disease occurrence. Global plant material movement, trade globalization, and increased human activities contribute to the introduction and spread of new pathogens. The interconnected nature of ecosystems and agricultural practices heightens the risk of disease transmission, emphasizing the need for proactive monitoring and management by farmers and researchers.

Mitigating plant diseases involves preventive measures, early detection, and interventions. Preventive actions like crop rotation, sanitation, and using resistant plant varieties aim to reduce disease likelihood. Early detection, aided by monitoring and advanced diagnostics, enables timely intervention, such as fungicide application. Integrated pest management (IPM) strategies, incorporating biological, cultural, and chemical controls, ensure sustainable disease management with minimal environmental impact. Understanding disease factors is crucial for resilient agricultural practices.

Plant disease incidence is the number of diseased plant units, usually relative to the total number assessed. The occurrence and prevalence of plant diseases vary from season to season, depending on the presence of the pathogen, environmental conditions, and the crops and varieties grown.

Infectious plant diseases are caused by a pathogenic organism such as a fungus, bacterium, mycoplasma, virus, viroid, nematode, or parasitic flowering plant.

There are four main types of Plant Diseases are found in every plant during analysis.

A. Bacterial
B. Fungal
C. Viral
D. Viroid

Figure: 2.3 Types of plant Diseases

2.9 Increasing of Plant Diseases

Analyzing Plant Disease occurrence data between 2000 and 2019 unveils a notable pattern in seismic activity. The bar graph illustrating the annual count of plant diseases indicates a steady rise in seismic events during this period.

Hyperspectral technology-based plant disease detection is drawing increasing attention. As shown in Figure: 2.5.1 hyperspectral-based plant disease analysis technology emerged in 2002 and has been developing rapidly in the following 10 years. It has been developing continuously with the maturity of related technologies in the past 10 years. These developments provide many methods and ideas for future research and analysis, as well as reliable support for plant protection.

Current technologies mainly concentrate on small scales, and there's a need for more development in satellite payloads. In the last thirty years, about 86% of hyperspectral imaging research has been centered on small areas like fields and labs, with a focus on leaves and canopies. However, for practical use, we need accurate analysis on a larger scale. While algorithms for small-scale hyperspectral data analysis can help regionally or at larger scales, achieving extensive monitoring is challenging without effective satellite payloads.

This gradual increase in Plant disease frequency suggests a complex interplay in agriculture factors and potentially underscores the importance of continuous monitoring and research to better understand the disease causes. The bar graph visually the increasing of plant disease occurrences from 2000 to 2019.

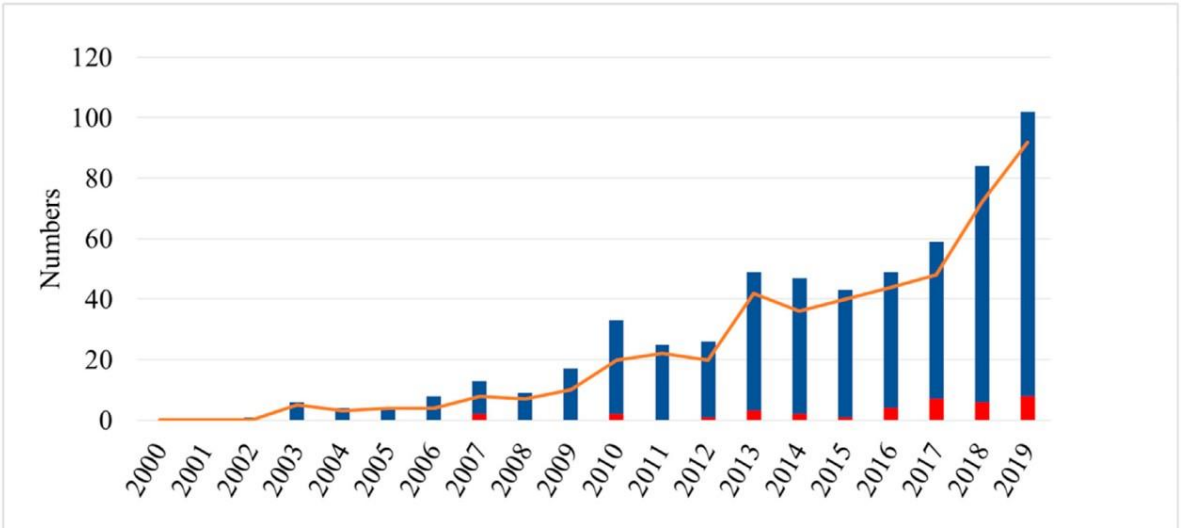

Figure: 2.4 Gradual Growth of Diseases in Plants

2.10 Importance of Deep Learning in Plant Disease

Deep learning plays a pivotal role in plant disease management, offering several key advantages in the identification, monitoring, and control of plant diseases.

Accurate Disease Detection:

- Deep learning models, particularly convolutional neural networks (CNNs), excel at image recognition tasks. In agriculture, this capability is harnessed for precise and accurate detection of plant diseases based on visual symptoms. The ability to analyze large datasets allows deep learning models to recognize subtle patterns and variations indicative of specific diseases.

Early Disease Diagnosis:

- Early detection of plant diseases is crucial for effective intervention and prevention of crop losses. Deep learning enables the development of automated systems that can quickly and accurately diagnose diseases in their initial stages. This early warning system helps farmers implement timely and targeted measures, reducing the impact of diseases on crop yield.

Large-Scale Monitoring:

- Deep learning facilitates large-scale monitoring of crops and plantations through the analysis of remote sensing data. Satellite and drone imagery, when processed by deep learning models, can provide comprehensive insights into the health of crops across vast agricultural areas. This capability is instrumental in identifying disease hotspots and implementing timely management strategies.

Handling Multimodal Data:

- Plant disease diagnosis often involves various types of data, such as visual images, spectral data, and environmental factors. Deep learning models, especially those designed for multimodal data fusion, can integrate and analyze diverse datasets. This holistic approach enhances the accuracy and robustness of disease diagnosis systems.

Adaptability and Learning:

- Deep learning models exhibit adaptability and the ability to learn from new data. As plant diseases evolve and new strains emerge, deep learning systems can continuously

update their knowledge base, ensuring that they remain effective in identifying and classifying novel diseases or variations.

Reduced Dependency on Human Expertise:

- Traditional methods of plant disease identification often rely on human expertise, which can be time-consuming and subjective. Deep learning reduces this dependency by automating the process, providing a more efficient and consistent means of disease diagnosis. This is particularly beneficial in regions where access to expert plant pathologists may be limited.

The importance of deep learning in plant disease management lies in its capacity for accurate, early, and large-scale detection, as well as its adaptability to handle diverse data types. These capabilities contribute to more effective and sustainable agricultural practices, ultimately supporting global food security.

2.11 Importance of Deep Learning using Python

Python, a widely used programming language, was developed by Guido van Rossum in 1991. Its versatile applications include.

Python 3 is the latest major version, but Python 2, while no longer receiving non-security updates, continues to maintain popularity. Python operates as an interpreted language, eliminating the need for a compilation step commonly found in languages like C or FORTRAN. To initiate the Python interpreter, simply enter "python" in the command line. This opens a prompt where Python commands can be entered directly. Try entering "2.5*3+5" to observe the result. To exit the Python interpreter, type ctrl-d.

As our Python programs or function definitions grow, it becomes practical to create and edit files using a text editor and then load them into Python when needed. This saves us from retyping everything each time we run the program. The standard Unix implementation of Python features an integrated development environment known as IDLE. To start IDLE, log in to the server from an xterm and type "IDLE". This opens a Python shell window, functioning as a regular Python interpreter but with limited editing capabilities.

Python can be written in Integrated Development Environments (IDEs) such as Thonny, Pycharm, Netbeans, Eclipse, and Anaconda, particularly beneficial when managing extensive collections of Python files.

Known for its readability, Python employs new lines to conclude a command, in contrast to other programming languages that often use semicolons or parentheses. The language relies on indentation, utilizing whitespace to define scope for loops, functions, and classes, as opposed to other languages that typically use curly brackets for the same purpose.

In the past, Deep Learning tasks required manual coding of algorithms, mathematical, and statistical formulas, making the process time-consuming and tedious. However, in modern times, Python has significantly eased and enhanced these tasks through various libraries, frameworks, and modules. Today, Python stands out as one of the most popular programming languages for Deep Learning, replacing many others in the industry, thanks in part to its extensive collection of libraries.

Python libraries that used in Deep Learning are:

- TensorFlow
- PyTorch
- Keras
- scikit-learn
- MXNet
- Caffe
- Theano
- CNTK (Microsoft Cognitive Toolkit)
- Fastai

TensorFlow:

- Is an open-source deep learning framework widely used for building and training neural networks. Its flexibility and scalability make it suitable for a range of applications, from research to production.

PyTorch:

- Developed by Facebook, is another popular deep learning framework. Known for its dynamic computational graph, PyTorch is favored by researchers for its intuitive design and ease of use in experimenting with complex neural network architectures.

Keras:

- Is a high-level neural networks API that runs on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK). It simplifies the process of building and experimenting with deep learning models, making it a popular choice for beginners.

Scikit-learn:

- Is primarily a machine learning library, it includes tools and utilities for neural network-based algorithms. It is particularly useful for implementing simple neural network models and is often used in conjunction with deep learning frameworks.

MXNet:

- Is an open-source deep learning framework known for its flexibility and efficiency. It supports both symbolic and imperative programming, making it suitable for a range of applications, including computer vision and natural language processing.

Caffe:

- Is a deep learning framework developed by the Berkeley Vision and Learning Center. It is widely used in computer vision applications and offers a flexible architecture for designing and training convolutional neural networks (CNNs).

Theano:

- Is a numerical computation library that allows developers to define, optimize, and evaluate mathematical expressions. While it is no longer actively developed, it played a significant role in the early development of deep learning frameworks.

CNTK (Microsoft Cognitive Toolkit):

- CNTK is a deep learning framework developed by Microsoft. It provides efficient training and evaluation of deep neural networks, supporting both convolutional and recurrent neural networks.

Fastai:

- Is a high-level deep learning library built on top of PyTorch. It simplifies the process of building and training complex neural network models by providing a concise and expressive API.

3. SYSTEM ANALYSIS

3.1 Scope of the project

The primary objective of this project is to predict the plant diseases through the analysis of pertinent information with the help of Hybrid Deep Learning with a neuro-fuzzy neural network was designed and developed for detecting and classifying plant leaves.

3.2 Analysis

The dataset that we use for the detection of plant disease was taken from the Kaggle repository which contain 20,639 images out of which 997 images belong to diseased Pepper bell's leaves, 2,000 images belong to diseased Potato's leaves, 14,421 images belong to diseased Tomato's leaves, 1,478 images belong to healthy Pepper bell's leaves. 152 images belong to healthy Potato's leaves, and 1,591 images belong to healthy Tomato's leaves.

- Pepper bell bacterial spot - 997 images
- Pepper bell healthy - 1478 images
- Potato early blight - 1000 images
- Potato late blight - 1000 images
- Potato healthy images - 152 images
- Tomato bacterial spot - 2127 images
- Tomato early blight - 1000 images
- Tomato late blight - 1909 images
- Tomato leaf Mold - 952 images
- Tomato Septoria leaf spot - 1771 images
- Tomato Spider mites Two spotted spider mite - 1676 images
- Tomato target spot - 1404 images
- Tomato Yellow Leaf Curl Virus - 3209 images
- Tomato mosaic virus - 373 images
- Tomato healthy - 1591 images

The provided information pertains to a collection of distinct leaf images, encompassing both diseased and healthy states. All the images are of jpg format.

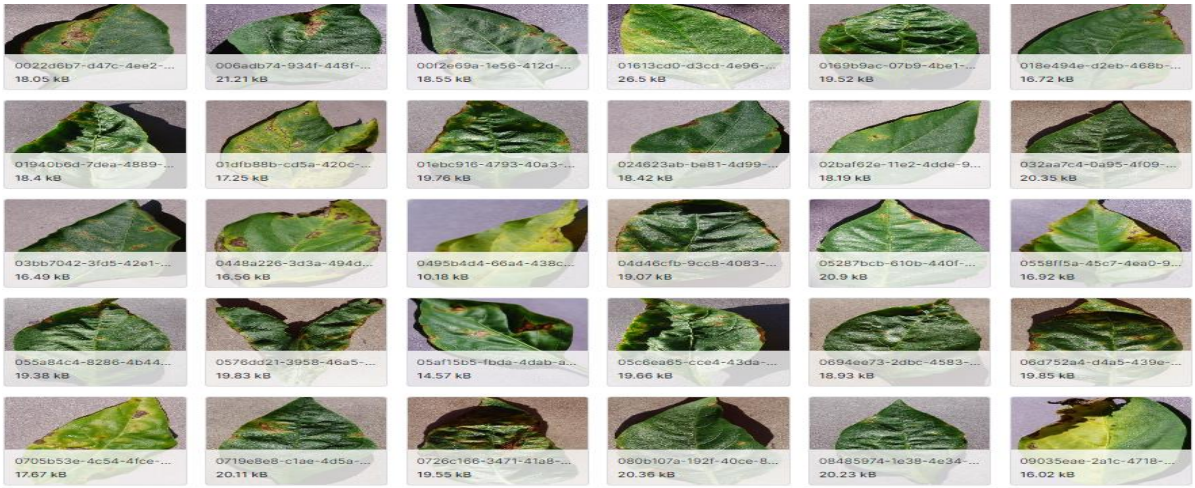

Figure: 3.1 Pepper bell diseased leaves

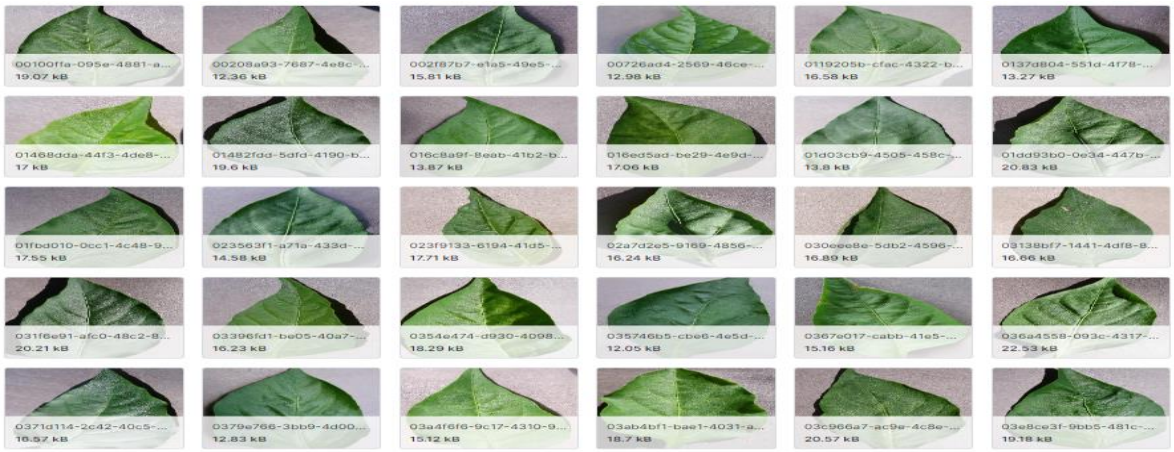

Figure: 3.2 Pepper bell healthy leaves

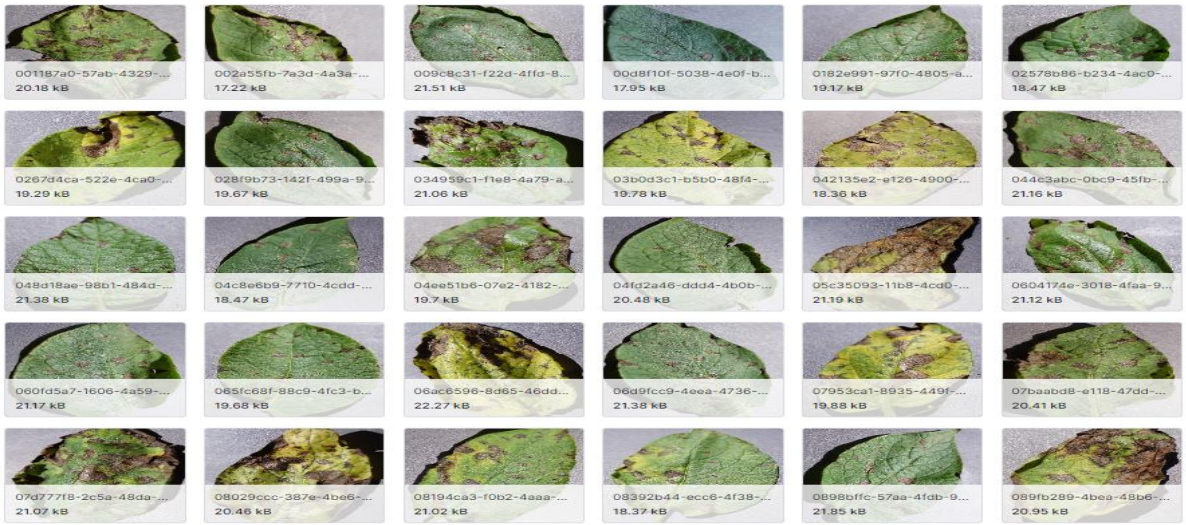

Figure: 3.3 Potato early blight diseased leaves

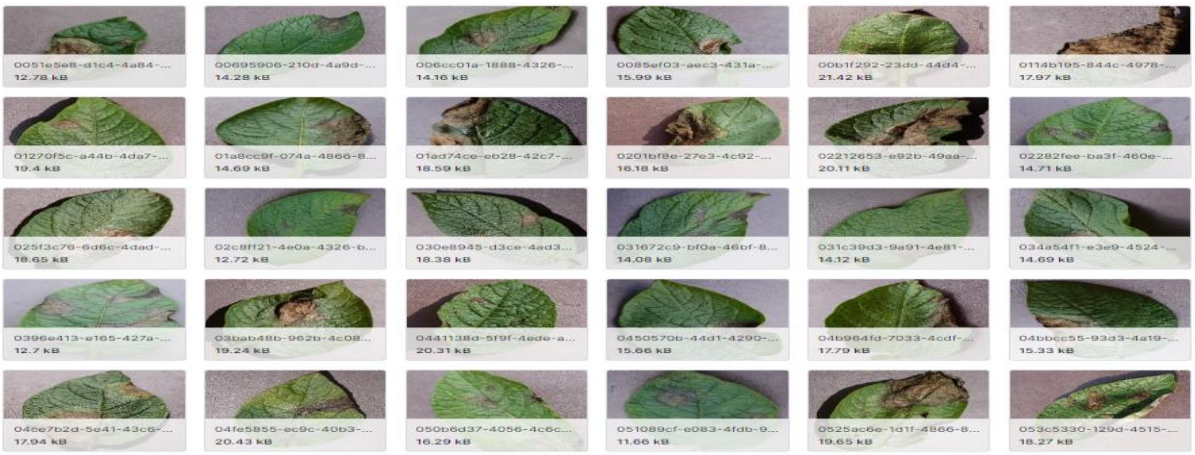

Figure: 3.4 Potato late blight diseased leaves

Figure: 3.5 Potato healthy leaves

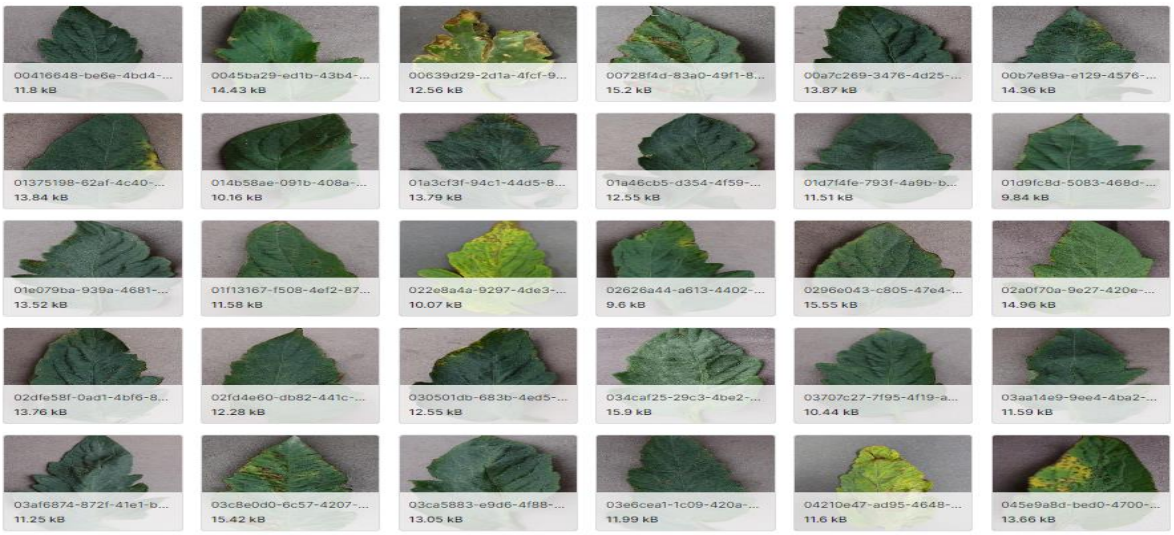

Figure: 3.6 Tomato bacterial spot diseased leaves

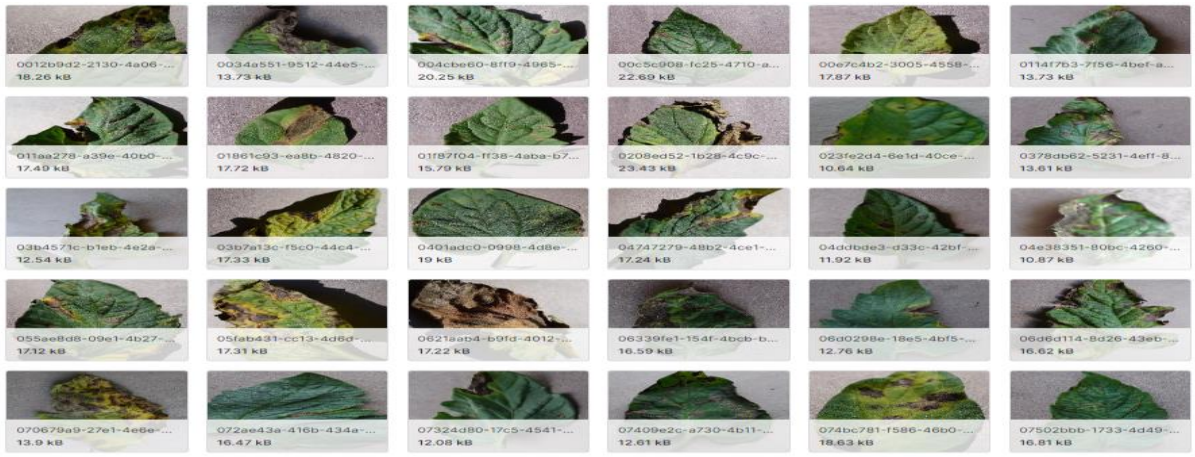

Figure: 3.7 Tomato early blight diseased leaves

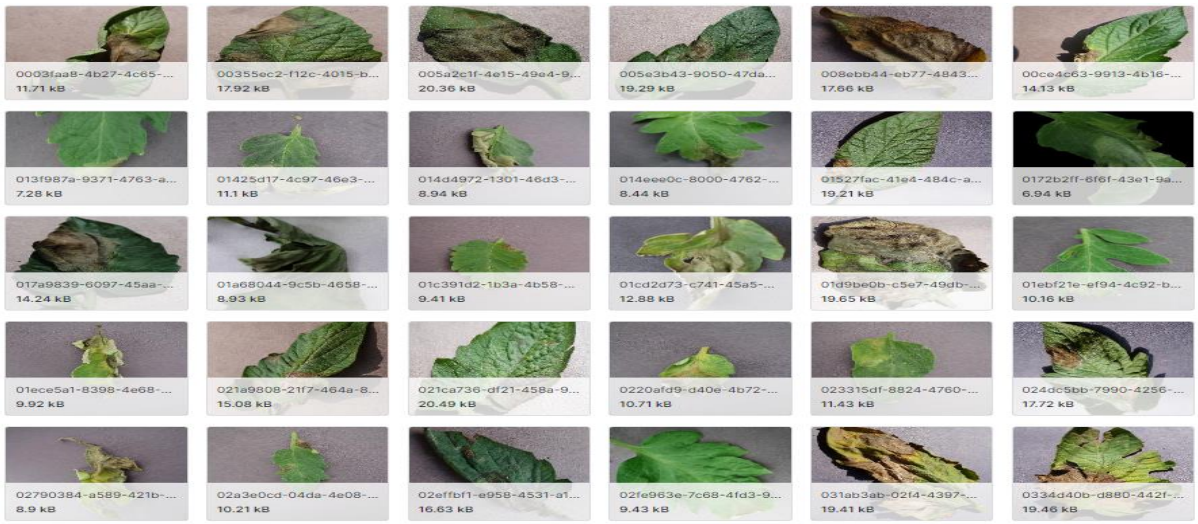

Figure: 3.8 Tomato late blight diseased leaves

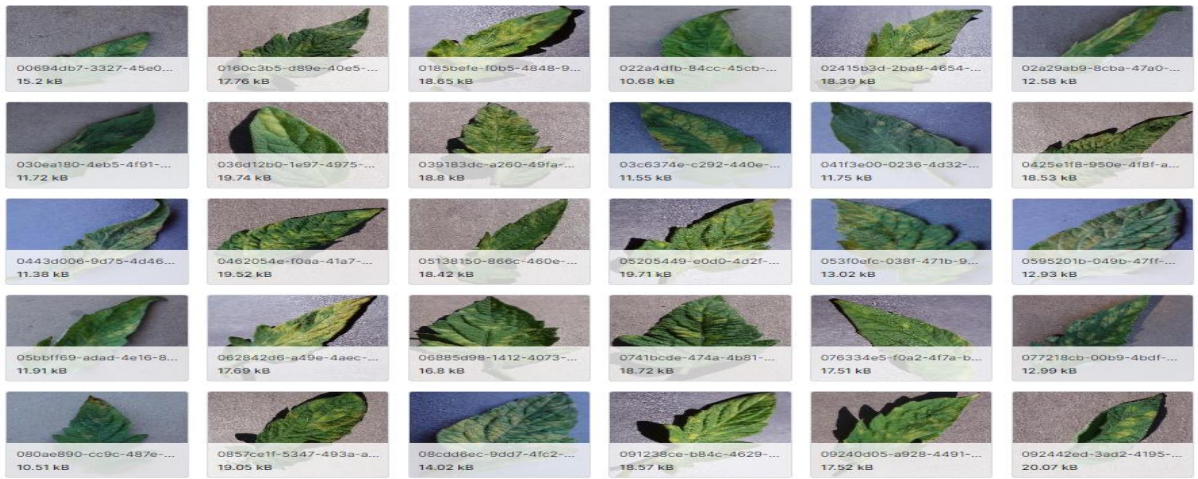

Figure: 3.9 Tomato leaf Mold diseased leaves

Figure: 3.10 Tomato Septoria leaf spot diseased leaves

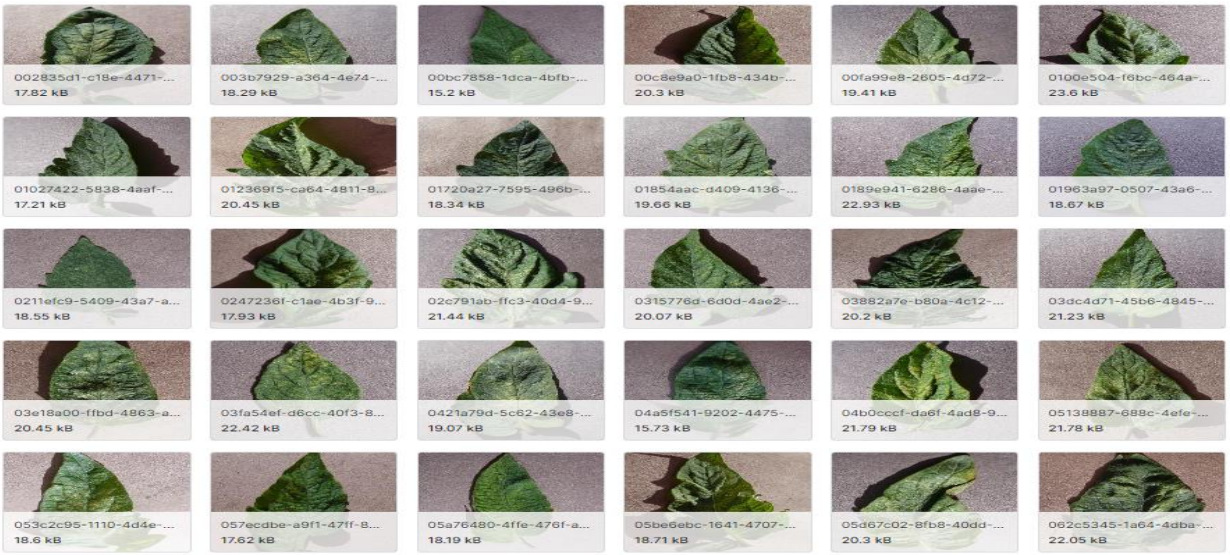

Figure: 3.11 Tomato Spider mites Two spotted spider mite diseased leaves

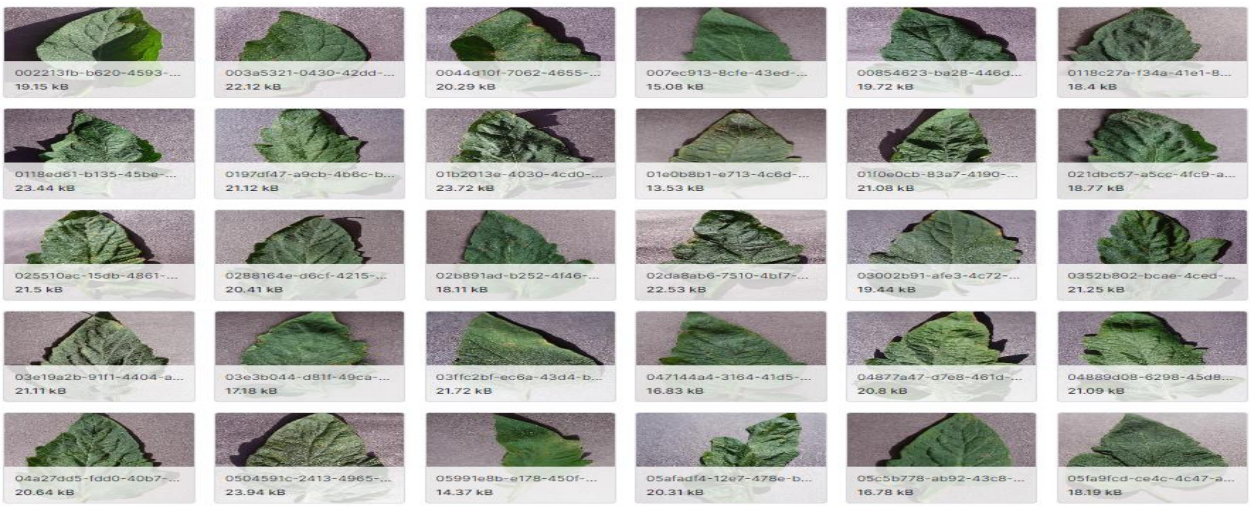

Figure: 3.12 Tomato target spot diseased leaves

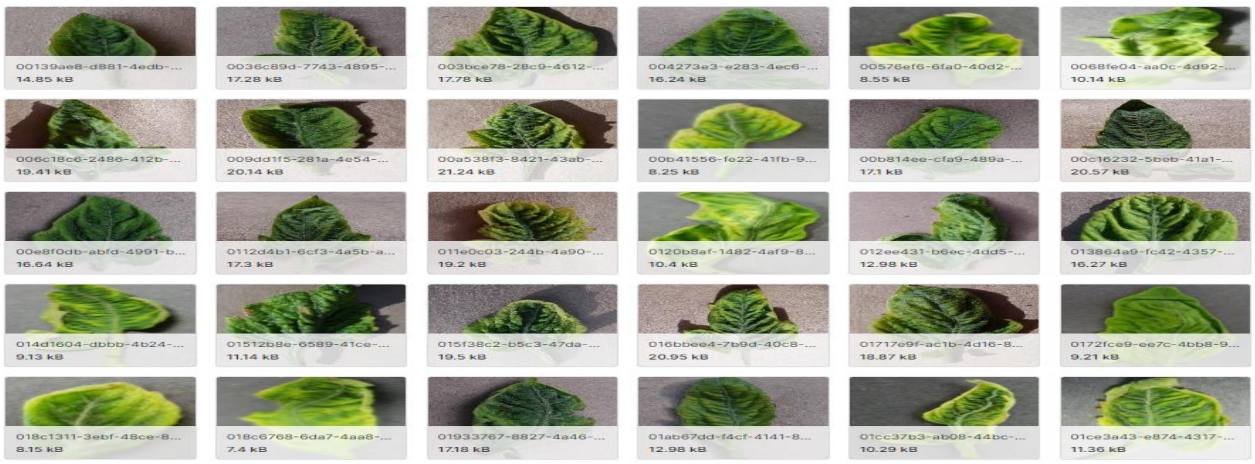

Figure: 3.13 Tomato Yellow Leaf Curl Virus diseased leaves

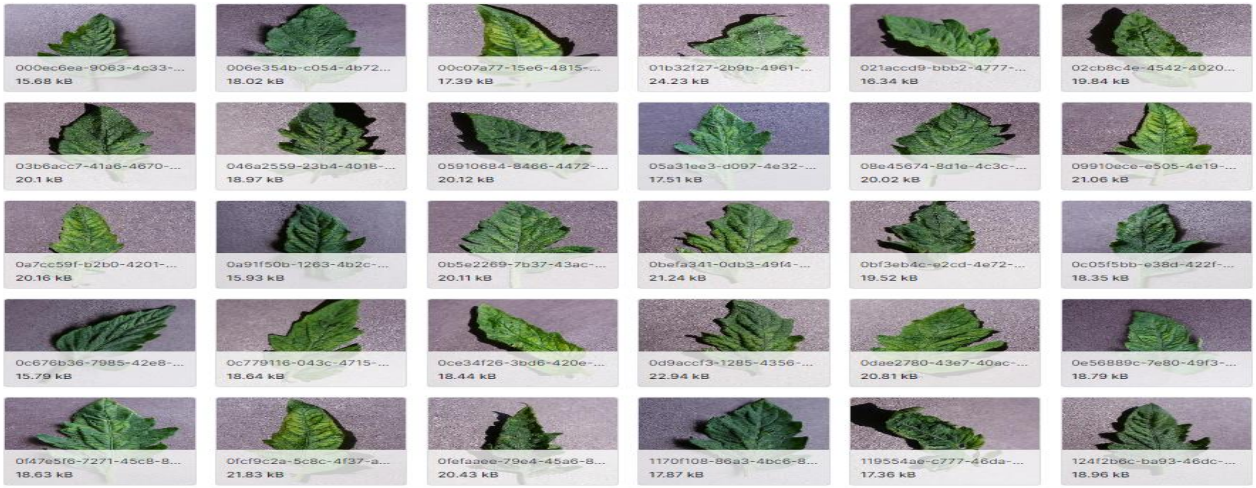

Figure: 3.14 Tomato mosaic virus diseased leaves

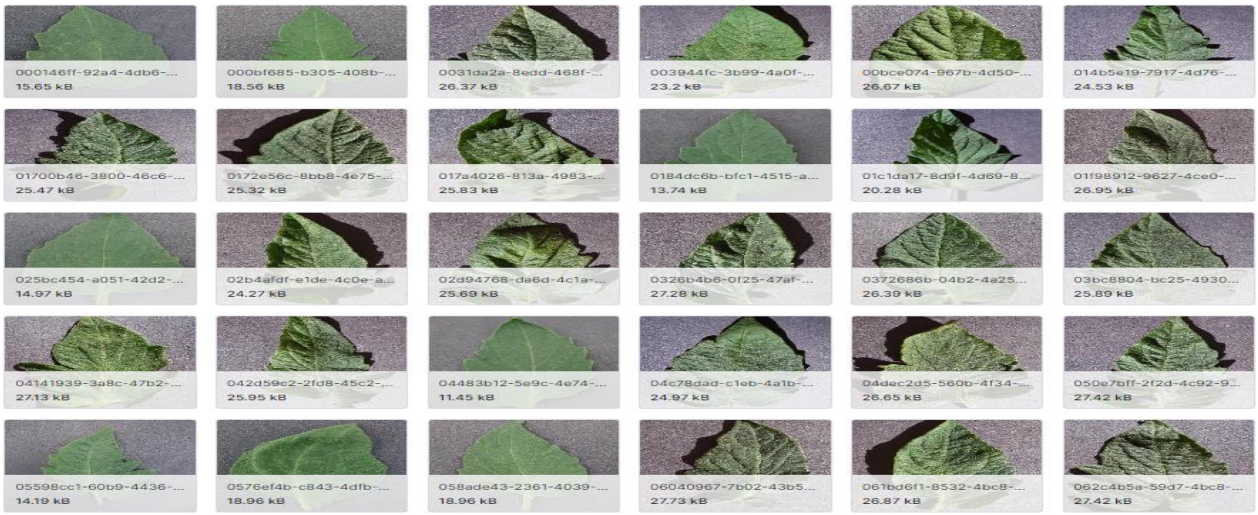

Figure: 3.15 Tomato healthy leaves

3.3 Data preprocessing and Feature Extraction

Data preprocessing is a crucial step in building effective neural network models. It involves cleaning and transforming raw data into a format that can be effectively used by the neural network.

Data Cleaning:

- Remove or handle missing values: Missing data can negatively impact model training. You can either remove instances with missing values or impute them using techniques like mean, median, or interpolation.
- Outlier detection and handling: Identify and handle outliers to prevent them from disproportionately influencing the model.

Data Normalization/Standardization:

- Scale numerical features to a similar range. Normalization involves scaling the values between 0 and 1, while standardization involves scaling to have a mean of 0 and a standard deviation of 1. This helps the neural network converge faster and can improve its performance.

Categorical Encoding:

- Convert categorical variables into numerical representations. One-hot encoding is a common technique where each category is represented by a binary value in a separate column.

Handling Imbalanced Data

- If your dataset has imbalanced classes, consider techniques such as oversampling the minority class, under sampling the majority class, or using synthetic data generation methods.

Feature Engineering:

- Create new features that might provide additional information to the model. Feature engineering can involve mathematical transformations, interaction terms, or domain-specific knowledge.

Figure: 3.16 illustrates the preprocessing steps for each image. To get precise results, some background noise should be removed before extraction of features. So first the RGB image is converted to greyscale and then Gaussian filter is used for smoothening of the image. Then to binaries the image, Otsu's thresholding algorithm is implemented. Then morphological transform is applied on inarized image to close the small holes in the foreground part. Now after foreground detection, the bitwise AND operation on inarized image and original color image is performed to get RGB image of segmented leaf. Now after image segmentation shape, texture and color features are extracted from the image. By using contours, area of the leaf and perimeter of the leaf is calculated. Contours are the line that joins all the points along the edges of objects having same color or intensity. Mean and standard deviation of each channel in RGB image is also estimated. To obtain amount of green color in the image, image is first converted to HSV color space and we have calculated the ratio of number of pixels having pixel intensity of hue (H) channel in between 30 and 70 and total number of pixels in one channel. Non green part of image is calculated by subtracting green color part from 1. After extracting color features from the image, we have extracted texture features from grey level co-occurrence matrix (GLCM) of the image00.

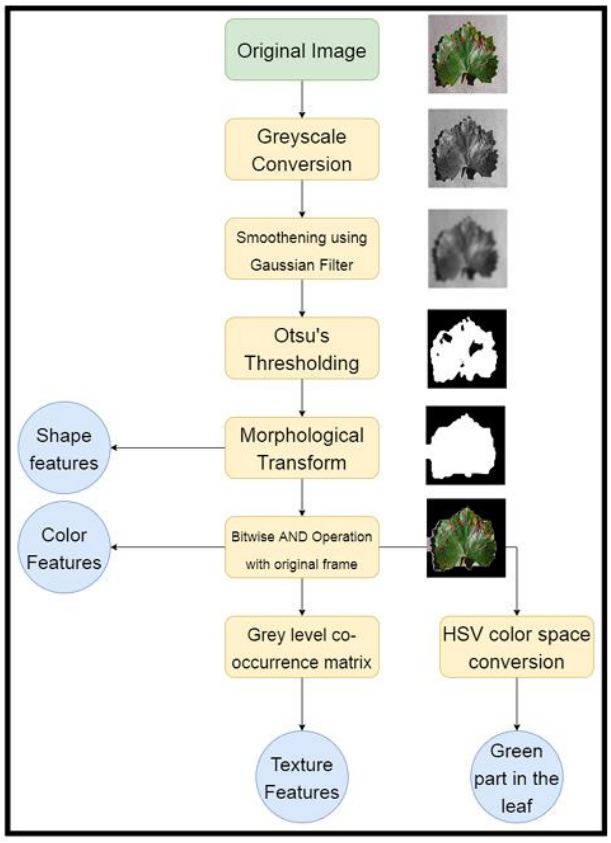

Figure: 3.16 Steps for data re-processing and feature extraction.

GLCM is the spacial relationship of pixels in the image. Extracting texture features from GCLM is one of the tradition methods in computer vision. We have extracted following features from GCLM:

- Contrast
- Dissimilarity
- Homogeneity
- Energy
- Correlation

After extracting all the features from all the images in the dataset, feature selection task is performed.

3.4 Feature Selection

Feature selection is an important step in all machine learning problems. In this project we are selecting the features on the basis of correlation of variables with target variable. Fig. 3.17 shows the correlation of each variable with each other for apple dataset. The correlation of feature green part of leaf (F1) and green part of leaf (F2) is very high (1) which means both variables are dependent on each other. So, we have dropped one of them (F2). Now for apple disease prediction, less correlated features such as green channel mean, red channel standard deviation, blue channel standard deviation, dissimilarity (f5) and correlation (f8) will not contribute too much in model development. So, we have dropped these variables also. After feature selection, the data is now parsed to machine learning classifiers to find the patterns in the data.

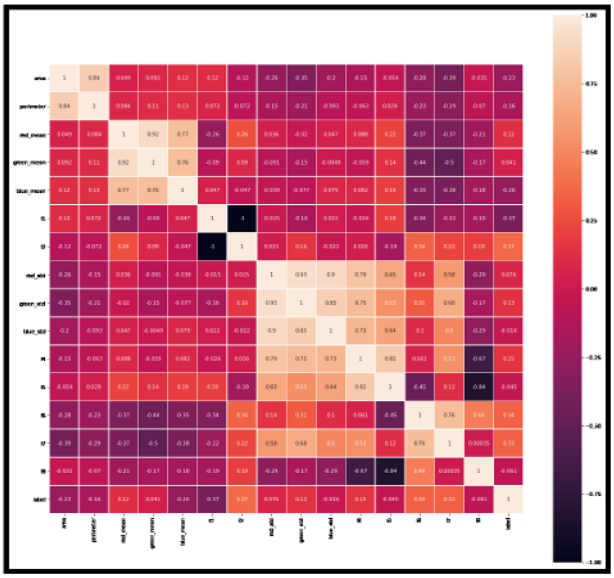

Figure: 3.17 Correlation plot for Apple dataset.

3.5 About Some Neural Network Models:

Data preprocessing to be done for the following Neural Network Models

A. **Google Net Model** the Google Inception V1 model, also known as GoogleNet, was introduced in 2014 through a collaborative effort between Google researchers and various universities. The research paper, titled "Going Deeper with Convolutions," presented this architecture, which emerged as the winner in the ILSVRC 2014 image classification challenge. Notably, it demonstrated a substantial reduction in error rates compared to the previous champions, namely AlexNet (ILSVRC 2012) and ZF-Net (ILSVRC 2013), as well as a significantly lower error rate than the 2014 runner-up, VGG. Key innovations in this architecture include the utilization of 1×1 convolutions within the network and the incorporation of global average pooling.

B. **Features of GoogleNet** the GoogLeNet architecture distinguishes itself from previous state-of-the-art designs, such as AlexNet and ZF-Net, by incorporating a variety of novel methods. Among these, the utilization of 1×1 convolutions and global average pooling plays a pivotal role, facilitating the creation of a more profound architecture. In the upcoming discussion, we will delve into the details of these techniques within the GoogLeNet framework, highlighting their specific contributions to the model's unique characteristics and performance.

- **1×1 convolution:** The inception architecture incorporates 1×1 convolutions as a strategic element. The purpose of these convolutions is to effectively decrease the number of parameters, encompassing both weights and biases within the architecture. This reduction in parameters serves a dual role, not only optimizing computational efficiency but also contributing to an increased depth in the architecture. To illustrate, consider the following example of a 1×1 convolution below:
- For Example, if we want to perform 5×5 convolution having 48 filters without using 1×1 convolution as intermediate:

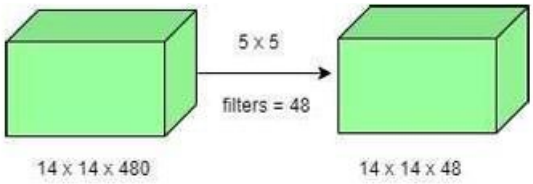

Figure: 3.18 5×5 convolutional layer with 48 filters without using 1×1 convolutions in between.

- Total Number of operations: (14 x 14 x 48) x (5 x 5 x 480) = 112.9M With 1×1 convolution:

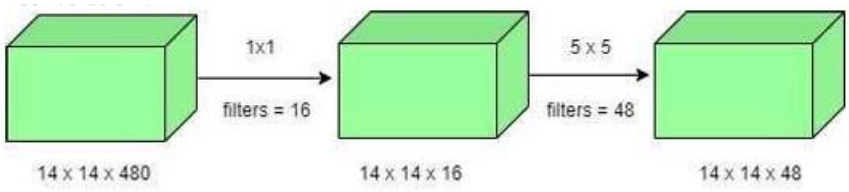

Figure: 3.19 1×1 convolution layer of GoogleNet

- (14 x 14 x 16) x (1 x 1 x 480) + (14 x 14 x 48) x (5 x 5 x 16) = 1.5M + 3.8M = 5.3M which is much smaller than 112.9M.

C. **Global Average Pooling** Unlike previous architectures like AlexNet, GoogLeNet avoids using fully connected layers at the end, which tend to harbour a substantial number of parameters, escalating computational costs. Instead, GoogLeNet employs a method known as global average pooling in its architecture's conclusion. This unique layer takes a 7×7 feature map and averages it down to 1×1, effectively reducing the trainable parameters to zero. This reduction not only optimizes computational efficiency but also enhances the top-1 accuracy by 0.6%.

D. **Inception Module** The inception module distinguishes itself from earlier architectures like AlexNet and ZF-Net by adopting a unique approach. Unlike its predecessors, this architecture employs a fixed convolution size for each layer. Within the Inception module, parallel operations involving 1×1, 3×3, 5×5 convolutions, and 3×3 max pooling are performed on the input. The outputs of these operations are then stacked together to generate the final output. This innovative design stems from the concept that convolution filters of various sizes can effectively address objects at multiple scales, enhancing the model's ability to handle diverse features.

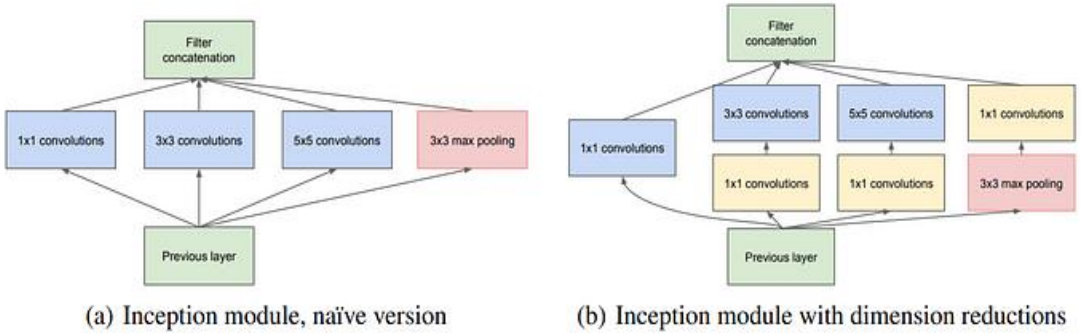

Figure: 3.20 Inception Module of GoogleNet

3.6 Model Architecture:

Below is Layer by Layer architectural details of GoogLeNet.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table: 3.21 Architecture of GoogleNet with Layered Details

The comprehensive architecture spans 22 layers, with a deliberate focus on computational efficiency. The design prioritizes the ability to run on individual devices, even those with limited computational resources. To further support the model's performance, two auxiliary classifier layers are integrated, connected to the outputs of the Inception (4a) and Inception (4d) layers. The auxiliary classifiers serve to enhance the network's capabilities and contribute to its overall efficiency. Now, let's delve into the specific architectural details of these auxiliary classifiers.

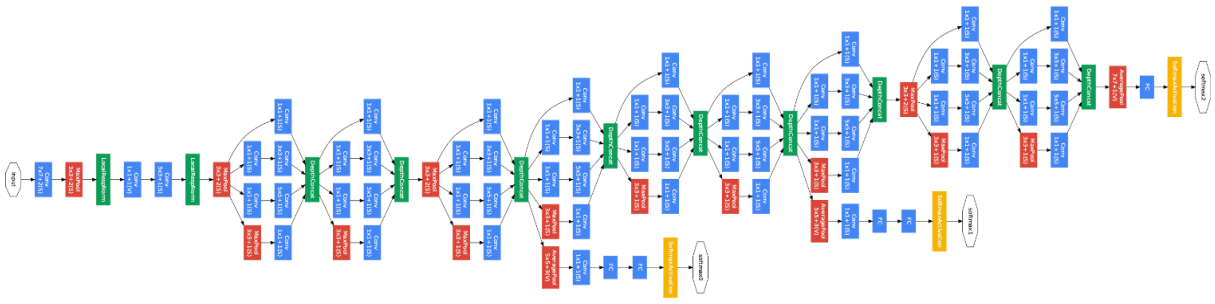

Figure: 3.22 Architecture of GoogleNet

This architecture takes image of size 224 x 224 with RGB color channels. All the convolutions inside this architecture uses Rectified Linear Units (ReLU) as their activation functions.

3.7 Results:

GoogLeNet was the winner at ILSRVRC 2014 taking 1st place in both classification a detection task. It has top-5 error rate of 6.67% in classification task. An ensemble of 6 GoogLeNets gives 43.9 % mAP on ImageNet test set.

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table: 3.23 Results of GoogleNet at ILSRVRC

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

Table: 3.24 Results of GoogleNet at ILSRVRC with details of data

3.8 Residual Networks:

Subsequent to Alex Net's triumph in the ImageNet 2012 competition, winning architectures aimed to boost performance by increasing the depth of neural networks. While this initially led to lower error rates, a common issue known as the Vanishing/Exploding gradient problem emerged with the rise in the number of layers. This problem occurs when gradients become either extremely small or excessively large during training, impacting the network's ability to learn effectively. The Vanishing/Exploding gradient problem poses a challenge in training deep neural networks, hindering the improvement in accuracy with more layers. To tackle this issue, researchers have explored techniques like careful weight initialization, normalization layers, and skip connections. Addressing the Vanishing/Exploding gradient problem is crucial to harness the potential benefits of deeper neural networks.

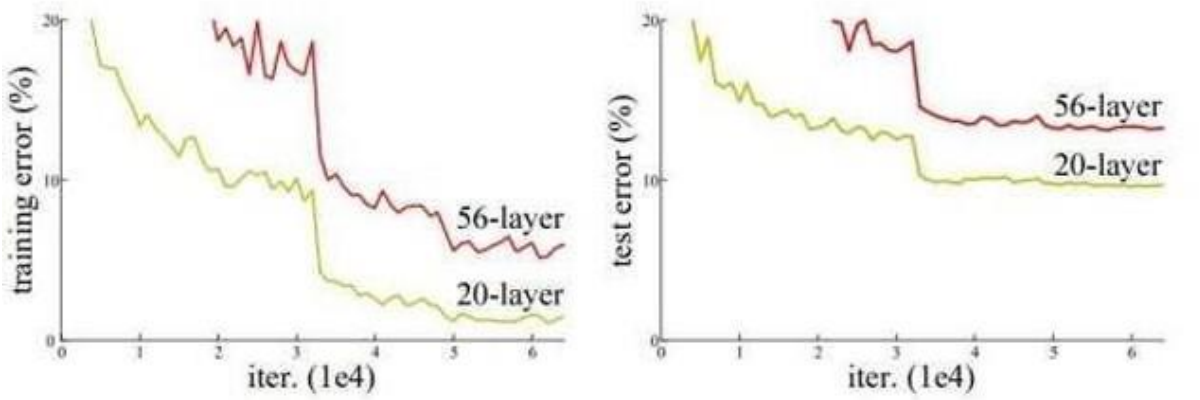

Figure: 3.25 Residual Networks with training and testing rates

In the depicted graph, it is evident that a 56-layer CNN exhibits higher error rates on both training and testing datasets compared to a 20-layer CNN architecture. If overfitting were the sole cause, one would expect lower training error in the 56-layer CNN, but paradoxically, it also exhibits higher training error. Further examination of the error rates led the authors to conclude that the issue stems from vanishing/exploding gradients. In 2015, Microsoft Research introduced ResNet, a novel architecture that addressed this problem through the introduction of Residual Networks.

3.9 Residual Block:

In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Network. In this network we use a technique called skip connections. The skip connection skips training from a few layers and connects directly to the output. The approach behind this network is instead of layers learn the underlying mapping, we allow network fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit, F(x): = H(x) – x which gives H(x) : = F(x) + x.

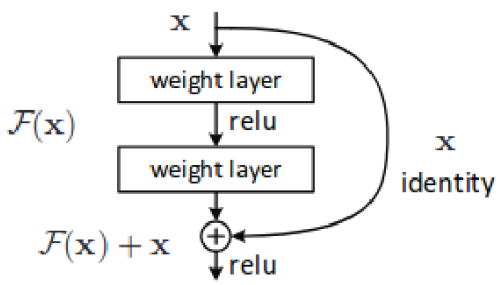

Figure: 3.25 Residual Block with ReLu

3.10 Network Architecture:

This network adopts a 34-layer plain network architecture, drawing inspiration from VGG-19. Subsequently, shortcut connections are introduced, transforming the architecture into a residual network.

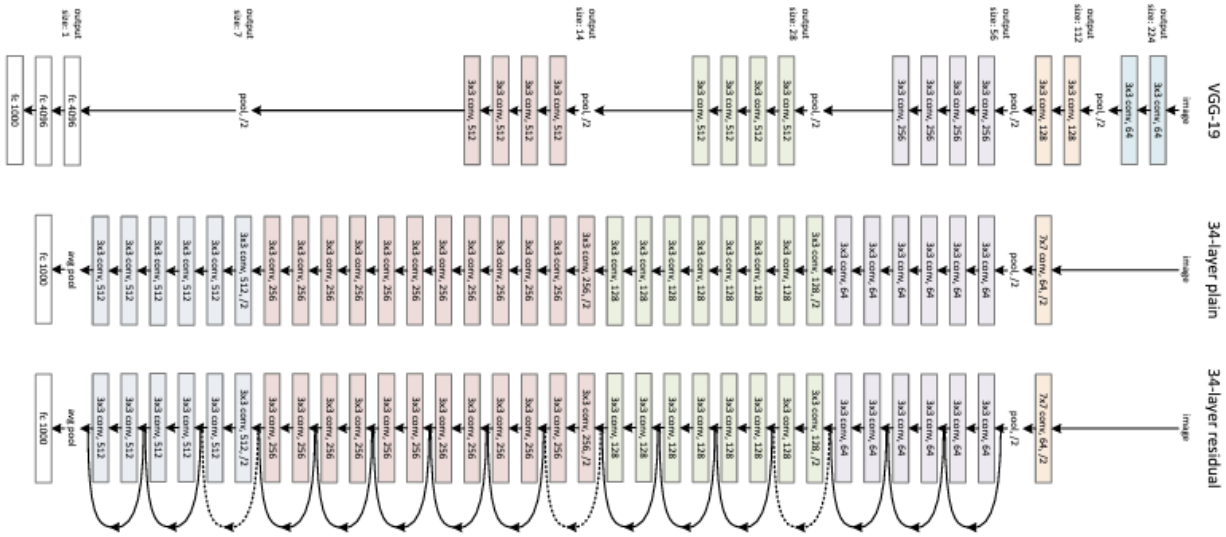

Figure: 3.26 VGG-34 Architecture

VGG-16 Model:

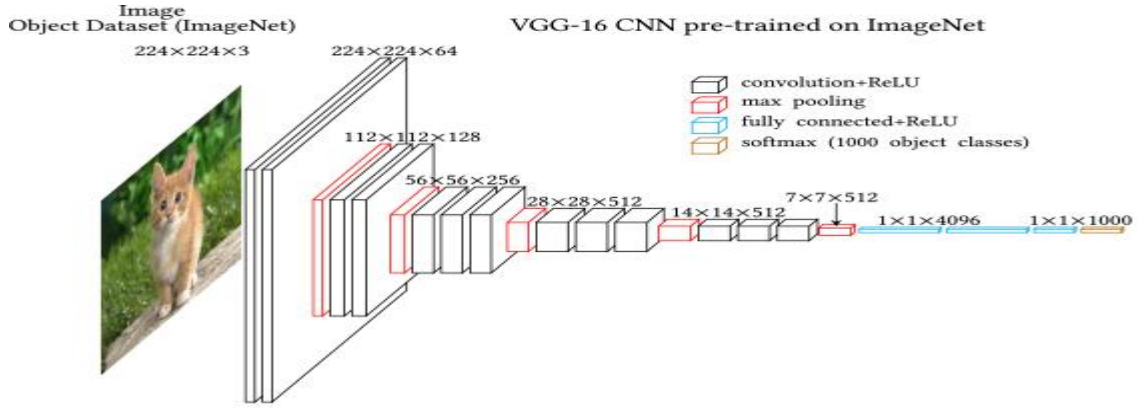

Figure: 3.27 VGG-16 Architecture

VGG16, a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition," has garnered recognition for achieving a top-5 test accuracy of 92.7% on the ImageNet dataset. This dataset comprises over 14 million images distributed

among 1000 classes and served as a benchmark for models submitted to ILSVRC-2014. VGG16's notable improvement over AlexNet involves the substitution of large kernel-sized filters (11 and 5 in the first and second convolutional layers, respectively) with multiple consecutive 3×3 kernel-sized filters. The training of VGG16 spanned several weeks, utilizing NVIDIA Titan Black GPUs for computational support.

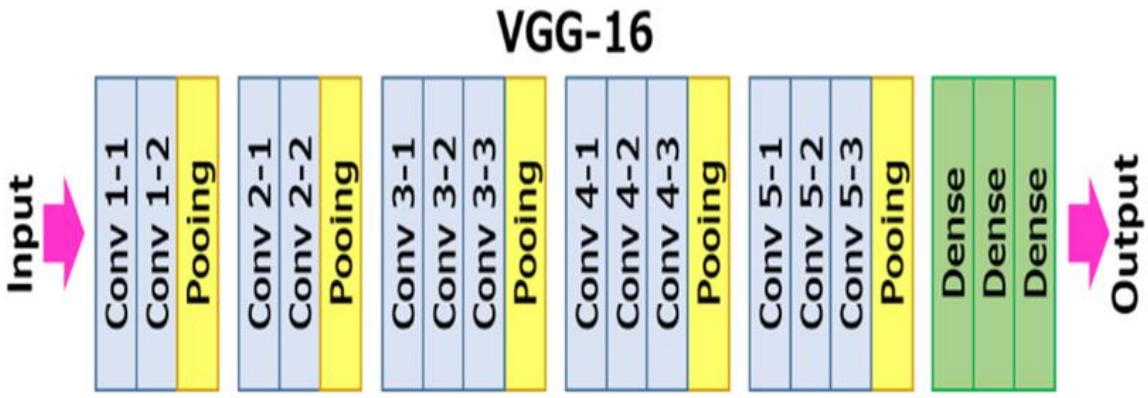

Figure: 3.28 VGG-16 Overview of layer

3.11 Configuration:

Table 3.29 illustrates the configurations of ConvNets 33abelled A to E, each varying in depth. Ranging from 11 to 19 weight layers, these configurations maintain a consistent generic design with 8 to 16 convolutional and 3 fully connected layers. The convolutional layer width starts at 64 channels in the initial layer, doubling after each max-pooling layer until reaching 512.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table:3.29 ConvNet Configuration

3.12 The Architecture

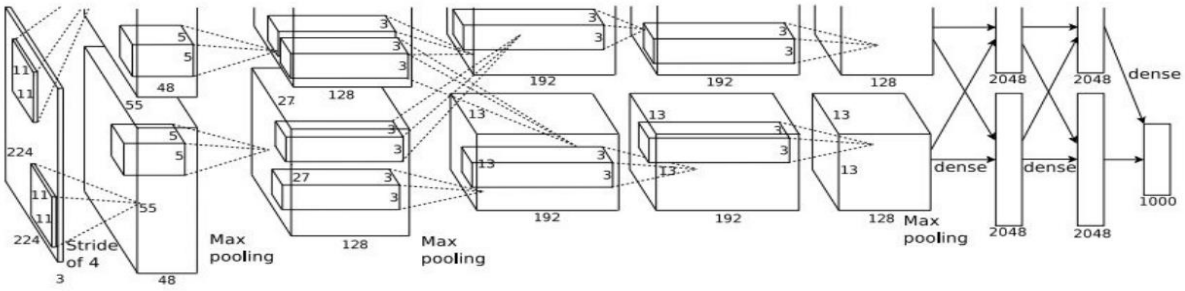

Figure: 3.30 AlexNet Model

A false positive (FP) occurs when the model incorrectly predicts the positive class, while a false negative (FN) refers to an outcome where the model incorrectly predicts the negative class.

Sensitivity or recall or hit rate or true positive rate (TPR)

- It is the proportion of individuals who actually have the disease were identified as having the disease.
  - TPR= tp / (tp + fn)

Precision or positive predictive value (PPV)

- If the test result is positive what is the probability that the patient actually has the disease.
  - PPV= tp / (tp + fp)

Negative predictive value (NPV)

- If the test result is negative what is the probability that the patient does not have disease.
  - NPV= tn/ (tn+fn)

Fall-out or false positive rate (FPR)

- It is the proportion of all the people who do not have the disease who will be identified as having the disease.
  - FPR=fp/ (fp+tn)

Accuracy

- The accuracy reflects the total proportion of individuals that are correctly classified.
  - ACC=(tp+tn) / (tp+tn+fp+fn)

4. DESIGN

4.1 The design of the system

There are 12 distinct steps in the proposed design, representing the layers and phases of a Convolutional Neural Network (CNN) for image processing.

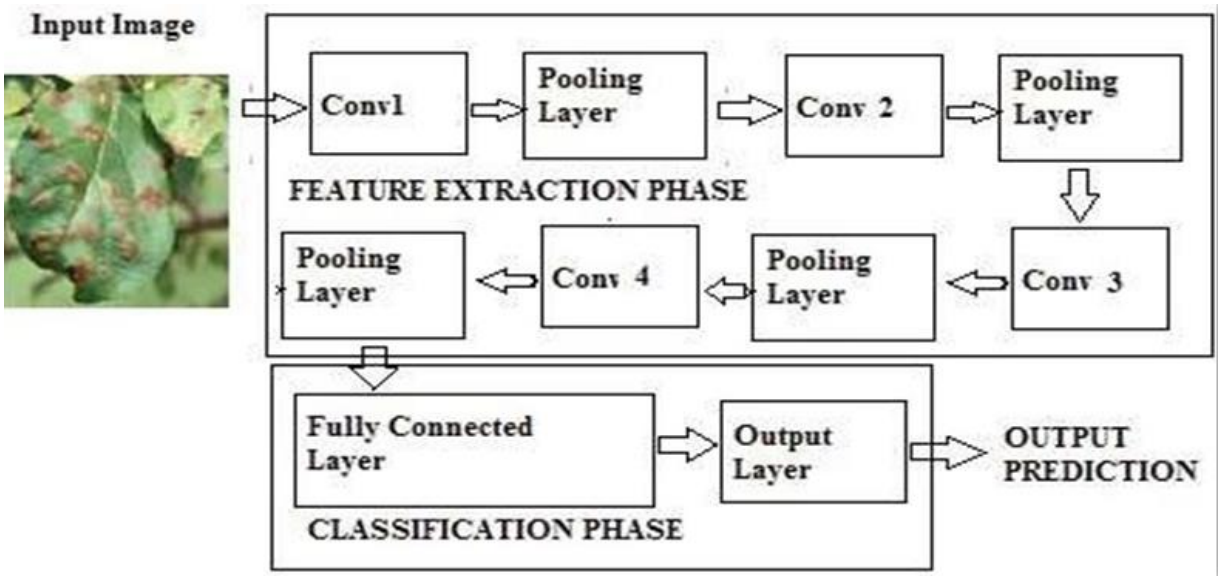

Figure: 4.1 Design

1. **Input Image:** The process starts with an input image of a plant leaf with visible signs of disease.
2. **Feature Extraction Phase:**
    - **Conv1:** The first convolutional layer (Conv1) applies filters to the input image to create a feature map, highlighting features like edges and textures.
    - **Polling Layer:** This layer reduces the spatial size of the feature map to decrease the computational power required to process the data while maintaining the important features.
    - **Conv2:** A second convolutional layer that further processes the data from the first pooling layer, extracting more complex features.
    - **Pooling Layer:** Another pooling layer to reduce the size of the feature map from Conv2.
    - **Conv3:** The third convolutional layer that processes the data from the second pooling layer.

- **Pooling Layer:** A pooling layer follows to reduce the size of the feature map from Conv3.
- **Conv4:** The fourth convolutional layer for additional feature extraction.
- **Pooling Layer:** The final pooling layer in the feature extraction phase.

3. **Classification Phase:**
    - **Fully Connected Layer:** This layer takes the flattened output from the final pooling layer and connects every neuron to every neuron in the next layer, which is useful for classifying the features extracted from the image.
    - **Output Layer:** The layer that outputs the prediction probabilities for each class.
    - **OUTPUT PREDICTION:** The final output of the network, which is the prediction of the plant disease class based on the input image.

This design represents a typical architecture for a CNN used in image classification tasks, where the network learns to identify and classify patterns within the image data.

4.2 Explanation of Algorithms Used in the Project

In this project we have used several algorithms and techniques would likely be used in combination to create a comprehensive solution for plant disease detection, risk assessment, and classification.

- Convolutional Neural Networks (CNNs)
- Fuzzy Logic System
- Neuro-Fuzzy Systems
- Supervised Learning
- Transfer Learning
- Evaluation Metrics
- Data Augmentation

Convolutional Neural Networks (CNNs)

CNNs are deep learning models specifically designed for processing structured grid-like data, such as images. They excel at feature extraction and hierarchical representation learning, making them well-suited for image classification tasks like identifying plant diseases based on images of leaves.

Convolutional Neural Networks (CNNs) are a type of deep learning model specifically designed for processing structured grid-like data, such as images. They are inspired by the human visual system and consist of multiple layers of neurons that perform operations such as convolution, pooling, and nonlinear activation. The key idea behind CNNs is to automatically learn hierarchical representations of features directly from raw input data.

At the core of CNNs are convolutional layers, which apply convolution operations to input data. These operations involve sliding a small filter (also known as a kernel) across the input image and computing dot products between the filter and local regions of the image. This process enables the network to capture spatial hierarchies of features, starting from low-level features like edges and textures to higher-level features like shapes and objects.

In a CNN, the convolutional layers apply filters (kernels) across the input image to detect patterns and features, such as edges, textures, or shapes. These filters are learned during the training process, allowing the network to adapt to the specific characteristics of the input data. Pooling layers then down sample the feature maps produced by the convolutional layers, reducing the spatial dimensions while retaining the most important information. Finally, fully connected layers integrate the extracted features to make predictions, such as image classification or object detection. Overall, CNNs have demonstrated remarkable performance in various computer vision tasks, owing to their ability to automatically learn and hierarchically represent complex visual features.

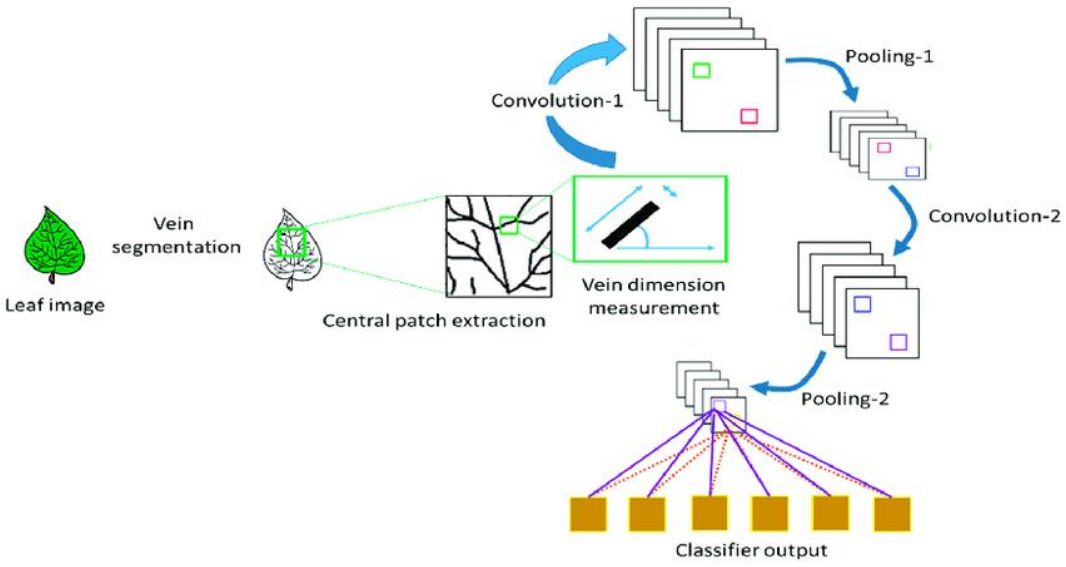

Figure: 4.2 Algorithm of Convolutional Neural Networks (CNNs)

Fuzzy Logic System

Fuzzy logic is a mathematical framework for handling uncertainty and imprecision in data. In your project, a fuzzy logic system could be used to model the uncertainty associated with plant disease diagnosis and risk assessment. This might involve defining linguistic variables, membership functions, fuzzy rules, and fuzzy inference mechanisms.

In a fuzzy logic system, linguistic variables are defined to represent input and output variables, along with membership functions that describe the degree of membership of a value in each linguistic term. Fuzzy rules are then formulated to capture the relationship between input and output variables, expressed in terms of fuzzy logic operators such as AND, OR, and NOT. These rules are combined using fuzzy inference mechanisms, such as Mamdani or Sugeno methods, to derive crisp output values from fuzzy input values.

Overall, fuzzy logic systems provide a powerful framework for capturing and formalizing human reasoning processes, making them particularly useful in domains where uncertainty and imprecision are inherent, such as decision-making, control systems, and pattern recognition.

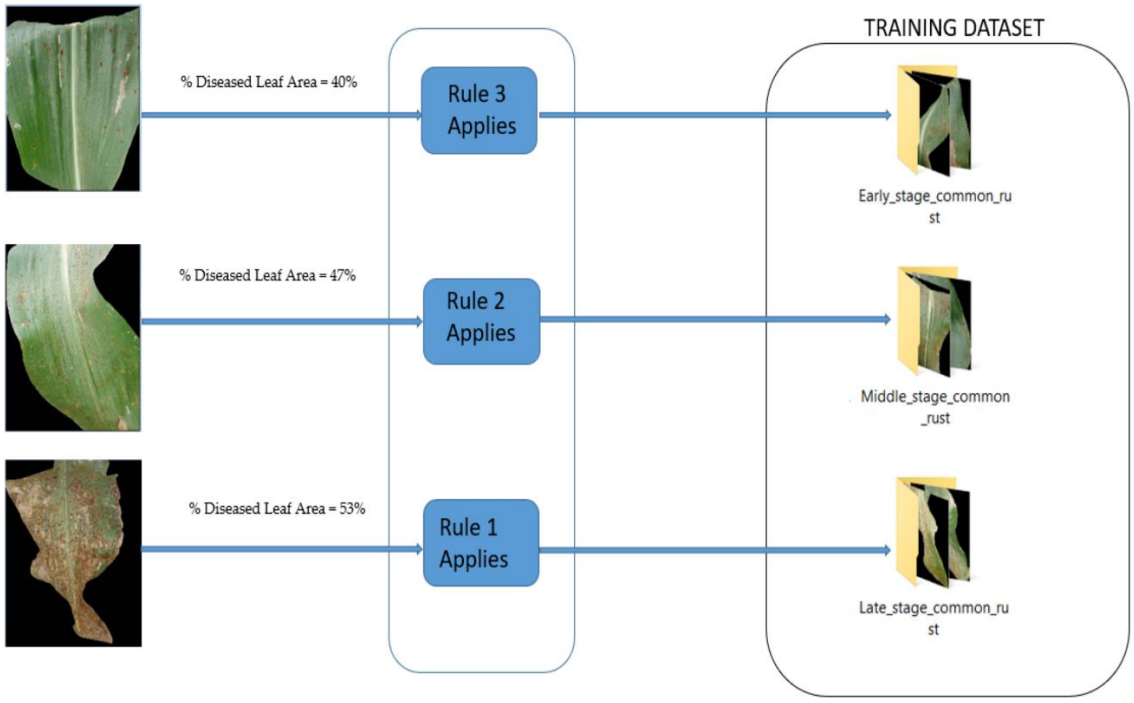

Figure: 4.3 Algorithm of Fuzzy logic System

Neuro-Fuzzy Systems

Neuro-fuzzy systems combine elements of fuzzy logic and neural networks to create hybrid models capable of handling both symbolic knowledge representation (fuzzy logic) and numerical data processing (neural networks). In your project, a neuro-fuzzy system could integrate fuzzy logic reasoning with the feature extraction capabilities of CNNs to enhance the interpretability and robustness of the model.

In a neuro-fuzzy system, fuzzy logic is used to model linguistic variables, membership functions, and fuzzy rules to capture human-like reasoning. Meanwhile, neural networks provide the computational power to learn from data and adaptively adjust the fuzzy logic components based on the input-output relationships observed in the training data. This integration allows neuro-fuzzy systems to effectively combine the interpretability of fuzzy logic with the learning capabilities of neural networks.

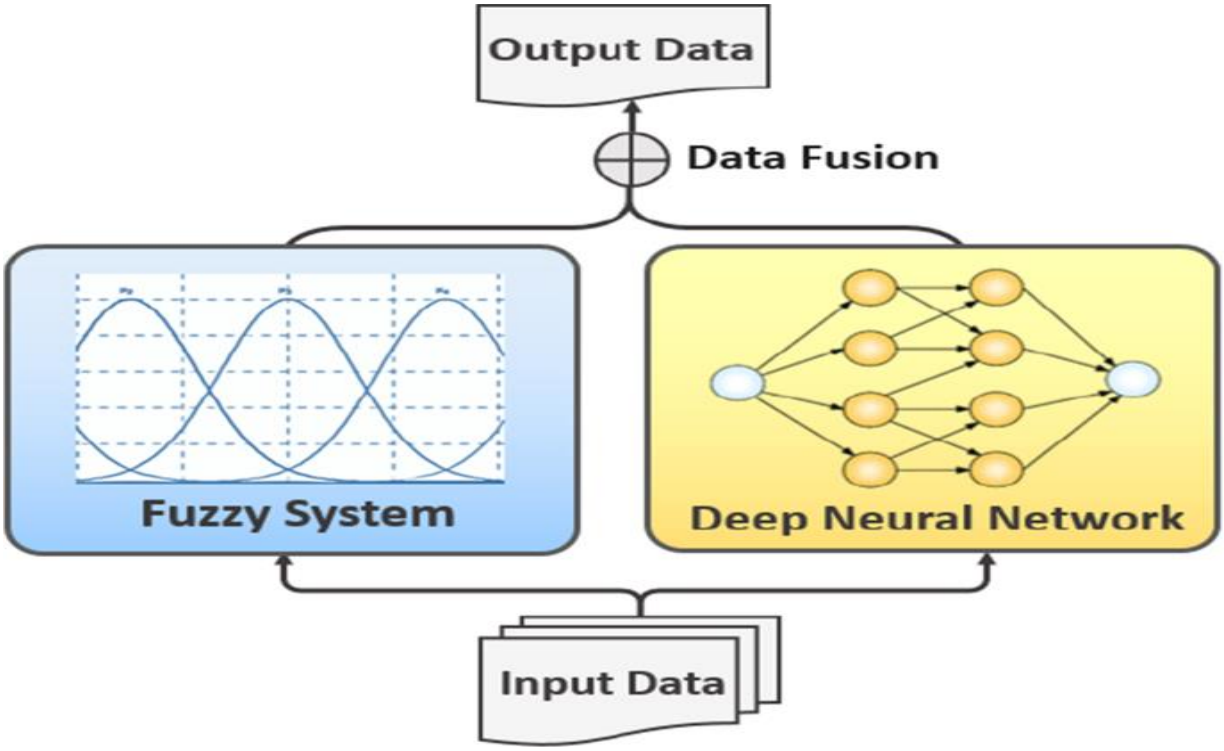

Figure: 4.4 Algorithm of Neuro-Fuzzy Systems

Supervised Learning

Since you'll likely have labeled data (images of plants labeled with their corresponding diseases), supervised learning algorithms will be used to train your model. This could include algorithms like backpropagation for training neural networks, which adjusts the model's parameters to minimize the difference between predicted and actual outputs.

During the training process, the supervised learning algorithm adjusts its internal parameters using optimization techniques to minimize the discrepancy between the predicted outputs and the actual labels in the training data. This process involves iteratively updating the model's parameters based on a chosen loss function, which quantifies the difference between predicted and true values. Common optimization algorithms used in supervised learning include gradient descent and its variants.

Supervised learning encompasses various types of tasks, including classification, regression, and structured prediction. In classification tasks, the model predicts discrete class labels for input instances, while in regression tasks, the model predicts continuous numerical values. Supervised learning algorithms are widely used in numerous applications, including image recognition, natural language processing, recommender systems, and predictive analytics.

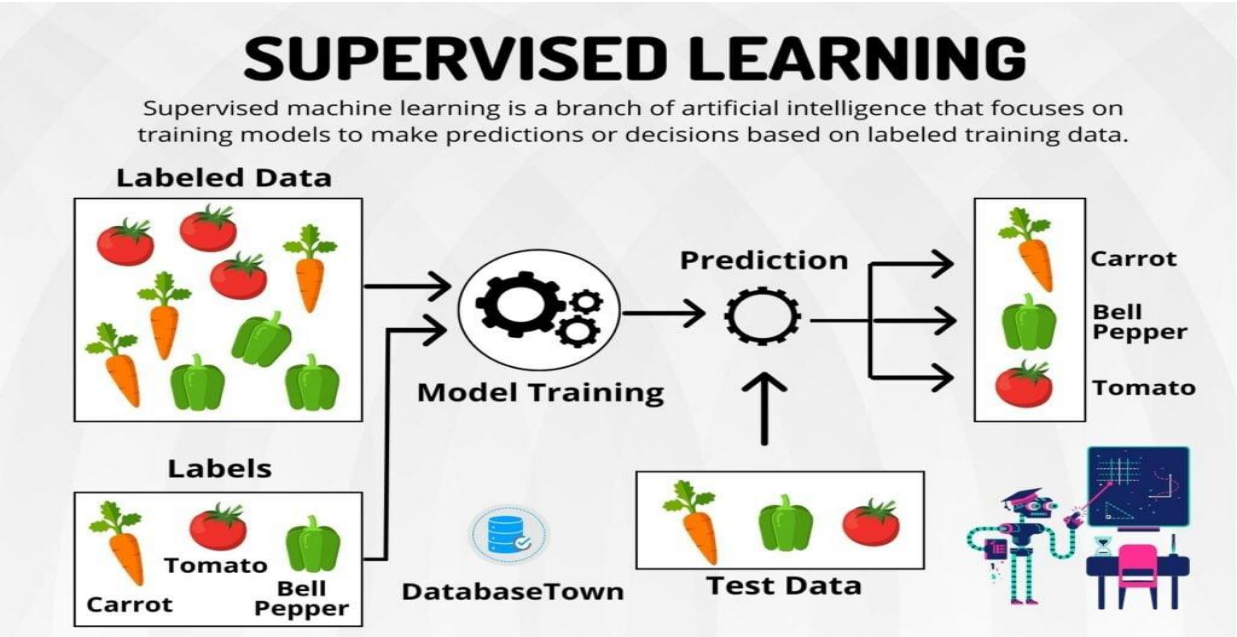

Figure: 4.5 Algorithm of Supervised Learning

Transfer Learning

Transfer learning is a technique where a model trained on one task is adapted for use on a different but related task. In your project, you might leverage pre-trained CNN models (e.g., VGG, ResNet) that have been trained on large-scale image datasets like ImageNet. By fine-tuning these pre-trained models on your plant disease dataset, you can take advantage of their learned features and potentially achieve better performance with less data.

The key idea behind transfer learning is to transfer the learned representations or knowledge from the source domain to the target domain, allowing the model to benefit from the features learned in the source task. This can lead to improved performance, faster convergence, and reduced training data requirements for the target task, especially when the source and target tasks share similar underlying patterns or features.

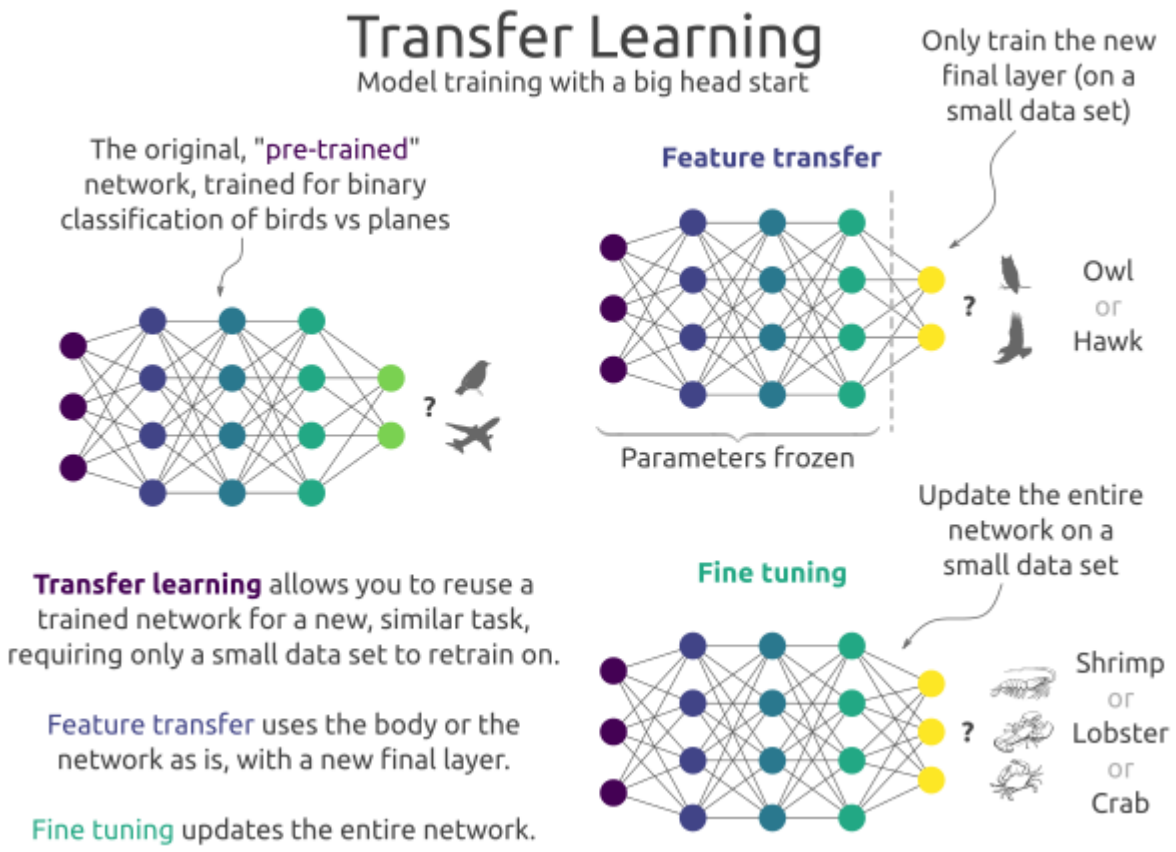

Figure: 4.6 Algorithm of Transfer Learning

Evaluation Metrics

Evaluation metrics are measures used to assess the performance of machine learning models on specific tasks. These metrics provide quantitative insights into how well a model is performing and help researchers and practitioners understand its strengths and weaknesses. Evaluation metrics vary depending on the nature of the task, such as classification, regression, clustering, or ranking.

For classification tasks, common evaluation metrics include accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). Accuracy measures the proportion of correctly classified instances out of all instances. Precision measures the proportion of true positive predictions among all positive predictions, focusing on the correctness of positive predictions. Recall, also known as sensitivity, measures the proportion of true positive predictions among all actual positive instances, focusing on the coverage of positive instances. F1-score is the harmonic mean of precision and recall, providing a balanced measure of both precision and recall. AUC measures the area under the receiver operating characteristic curve, which plots the true positive rate against the false positive rate at various threshold settings, providing a measure of the model's ability to discriminate between positive and negative instances.

For regression tasks, common evaluation metrics include mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and R-squared (R2) coefficient of determination. MSE measures the average squared difference between predicted and actual values, penalizing larger errors more heavily. RMSE is the square root of MSE, providing a measure of the average error in the same units as the target variable. MAE measures the average absolute difference between predicted and actual values, providing a more robust measure of error. R2 coefficient of determination measures the proportion of the variance in the target variable that is explained by the model, with higher values indicating better model fit.

Data Augmentation

Data augmentation is a technique used to artificially increase the size and diversity of a dataset by applying transformations to the existing data samples. This technique is commonly used in machine learning, particularly in tasks involving image or text data, to improve model generalization and robustness.

For image data, data augmentation techniques include rotation, flipping, scaling, cropping, translation, shearing, and changing brightness or contrast. These transformations create new variations of the original images, effectively enlarging the dataset and exposing the model to a wider range of scenarios and conditions. For example, flipping an image horizontally or vertically can create mirror images, while rotating an image can simulate different viewpoints or orientations. Similarly, scaling and cropping can simulate variations in object size and location within the image.

For text data, data augmentation techniques may include synonym replacement, random insertion or deletion of words, shuffling word order, or adding typographical errors. These techniques introduce variations in the textual content while preserving its semantic meaning, allowing the model to learn more robust representations of the language.

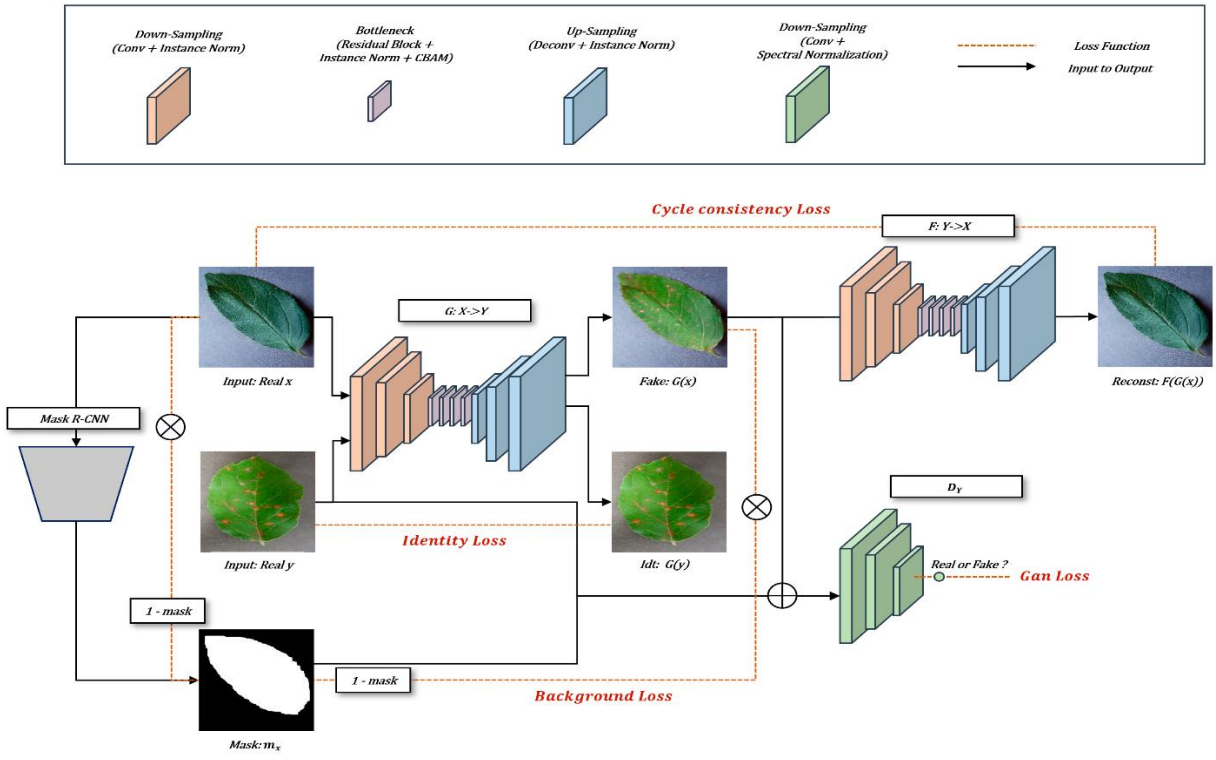

Figure: 4.7 Algorithm of Data augmentation

5. IMPLEMENTATION

Model:

Import necessary packages

```
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
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

Using TensorFlow backend.

```
EPOCHS = 25
INIT_LR = 1e-3
BS = 32
default_image_size = tuple((256, 256))
image_size = 0
directory_root = '../input/plantvillage/'
width=256
height=256
depth=3
```

Function to convert image to array

```
def convert_image_to_array(image_dir):
try:
image = cv2.imread(image_dir)
if image is not None:
image = cv2.resize(image, default_image_size)
return img_to_array(image)
else:
return np.array([])
except Exception as e:
print (f"Error: {e}")
return None
```

Fetch images from directory

```
image_list, label_list = [], []
try:
print("[INFO] Loading images ...")
root_dir = listdir(directory_root)
for directory in root_dir :
# remove .DS_Store from list
if directory == ".DS_Store" :
root_dir.remove(directory)
for plant_folder in root_dir :
plant_disease_folder_list = listdir(f"{directory_root}/{plant_folder}")
for disease_folder in plant_disease_folder_list :
# remove .DS_Store from list
if disease_folder == ".DS_Store" :
plant_disease_folder_list.remove(disease_folder)
```

```
for plant_disease_folder in plant_disease_folder_list:
print(f"[INFO] Processing {plant_disease_folder} ...")
plant_disease_image_list =listdir(f"{directory_root}/{plant_folder}/{plant_disease_folder}/")
for single_plant_disease_image in plant_disease_image_list :
if single_plant_disease_image == ".DS_Store" :
plant_disease_image_list.remove(single_plant_disease_image)
for image in plant_disease_image_list[:200]:
image_directory = f"{directory_root}/{plant_folder}/{plant_disease_folder}/{image}"
if image_directory.endswith(".jpg") == True or image_directory.endswith(".JPG") == True:
image_list.append(convert_image_to_array(image_directory))
label_list.append(plant_disease_folder)
print("[INFO] Image loading completed")
except Exception as e:
print(f"Error : {e}")
```

Get Size of Processed Image

```
image_size = len(image_list)
image_size
```

Transform Image Labels using Scikit Learns' s LabelBinarizer

```
label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
pickle.dump(label_binarizer,open('label_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
```

Print the classes

print(label_binarizer.classes_)

np_image_list = np.array(image_list, dtype=np.float16) / 225.0

print("[INFO] Spliting data to train, test")

x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels, test_size=0.2,random_state = 42)

[INFO] Spliting data to train, test

aug = ImageDataGenerator(

rotation_range=25, width_shift_range=0.1,

height_shift_range=0.1, shear_range=0.2,

zoom_range=0.2,horizontal_flip=True,

fill_mode="nearest")

model = Sequential()

inputShape = (height, width, depth)

chanDim = -1

if K.image_data_format() == "channels_first":

inputShape = (depth, height, width)

chanDim = 1

model.add(Conv2D(32, (3, 3), padding="same",input_shape=inputShape))

model.add(Activation("relu"))

model.add(BatchNormalization(axis=chanDim))

model.add(MaxPooling2D(pool_size=(3, 3)))

model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding="same"))

model.add(Activation("relu"))

```
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(1024))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(n_classes))
model.add(Activation("softmax"))
```

Model Accuracy

```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

Save model using Pickle

```
# save the model to disk

print("[INFO] Saving model...")

pickle.dump(model,open('cnn_model.pkl', 'wb'))
```

6. TESTING ANALYSIS

6.1 Testing analysis for our custom CNN Model

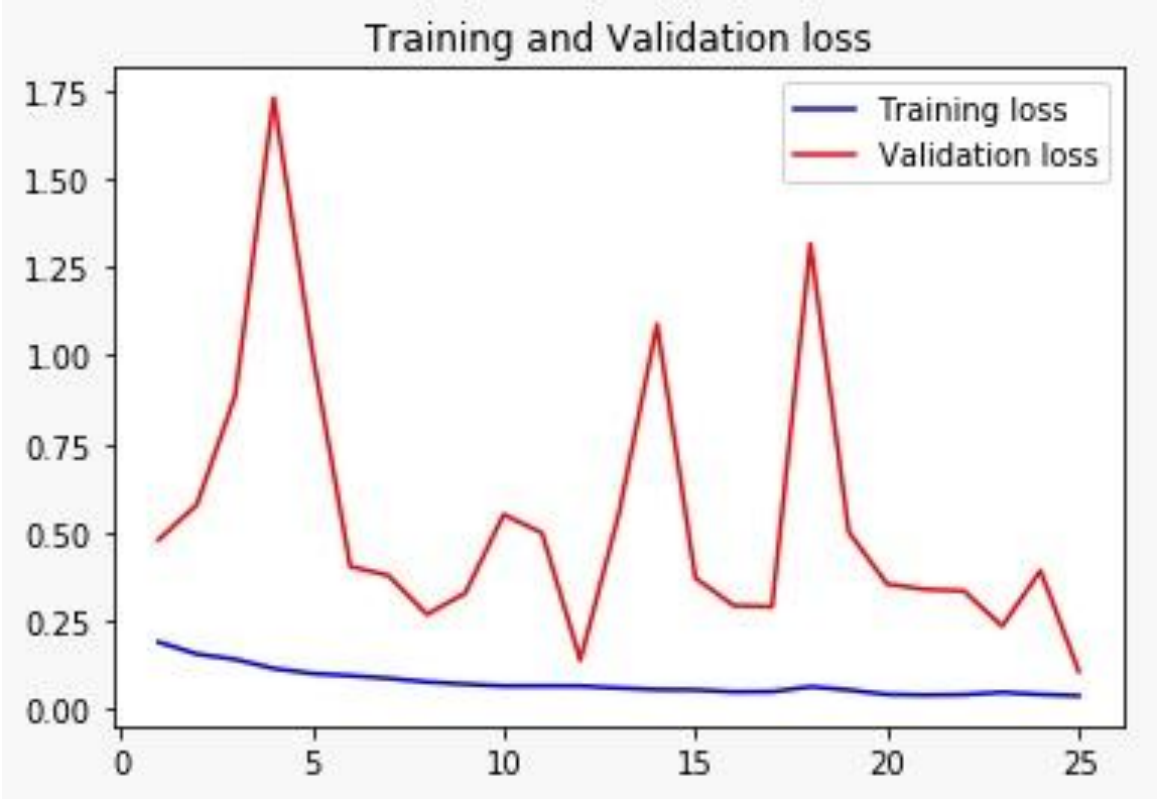

Figure: 6.1 Training and Validation loss

The figure 6.1 displays a line graph "Training and Validation loss." There are two lines on the graph: one in blue representing the training loss and one in red representing the validation loss. The x-axis is labeled but the specific label is not visible; it likely represents the number of epochs or iterations during the training of a machine learning model. The y-axis represents the loss value, which measures the model's error during training.

The blue line (training loss) starts at a higher value and generally trends downward, indicating that the model is learning and improving its performance on the training data over time. The red line (validation loss) also shows a downward trend but with significant spikes at certain points, which could suggest overfitting at those epochs where the model performs well on the training data but poorly on unseen validation data.

The x-axis is numbered from 0 to 25, suggesting that the graph covers 25 epochs or iterations. The y-axis ranges from 0 to 1.75, with the loss values plotted within this range.

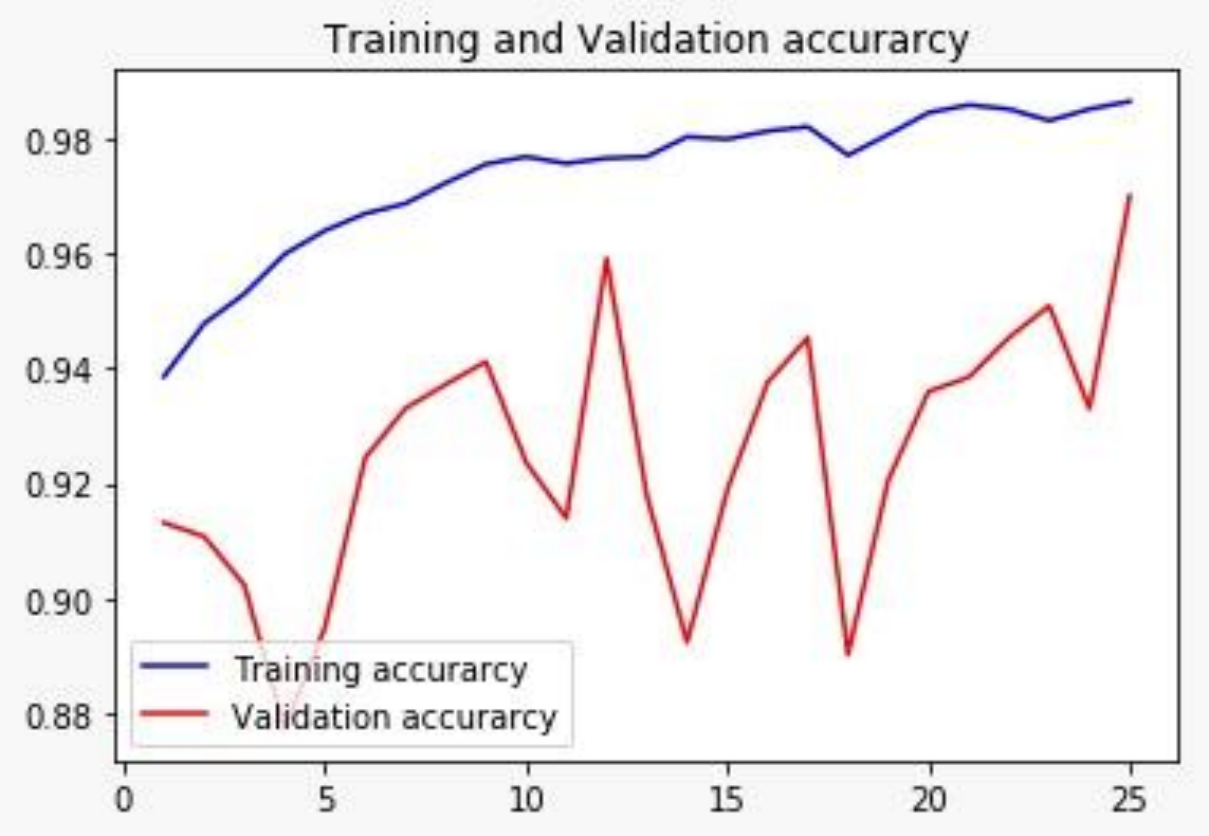

Figure: 6.2 Training and Validation accuracy

The figure 6.2 displays a line graph "Training and Validation accuracy," which plots two lines representing the accuracy of a machine learning model during its training phase. The x-axis represents the epoch number, which ranges from 0 to 25, and the y-axis represents accuracy, ranging from 0.88 to 0.98.

There are two lines on the graph:

A. The blue line represents the training accuracy. It starts off just below 0.98 and remains relatively stable throughout the training process, with only minor fluctuations.
B. The red line represents the validation accuracy. This line starts at approximately 0.91, shows more variability than the training accuracy, dips slightly below 0.90, and then generally trends upward, reaching around 0.96 by the 25th epoch.

The graph is used to evaluate the performance of the model, where ideally, both training and validation accuracy should be high and close to each other. The fluctuations and the gap between the two lines might indicate issues such as overfitting or underfitting, and could be a point of analysis for further model tuning.

7. RESULT ANALYSIS

7.1 Result analysis for our custom CNN Model

The accuracy of a deep neuro fuzzy network's predictions is influenced by both the training rate and the batch size. To identify the optimal combination of these parameters, we experimented with various initial learning rates and batch sizes, observing shifts in the learning rate. Our exploration ranged from an initial learning rate of 0.00001 to 0.0001, then to 0.001, and finally to 0.002. Notably, at an initial learning rate of 0.00001, the recognition accuracy fluctuates based on the chosen batch size among the four available options. We maintained consistent settings with those applied to the deep neuro-fuzzy network for training. Subsequently, utilizing the trained model for predictions on the test set resulted in an improved identification accuracy of 96.9%, regardless of any other adjustments.

Figures 7.1 and 7.2 provide a comparison of the precision between the two distinct networks.

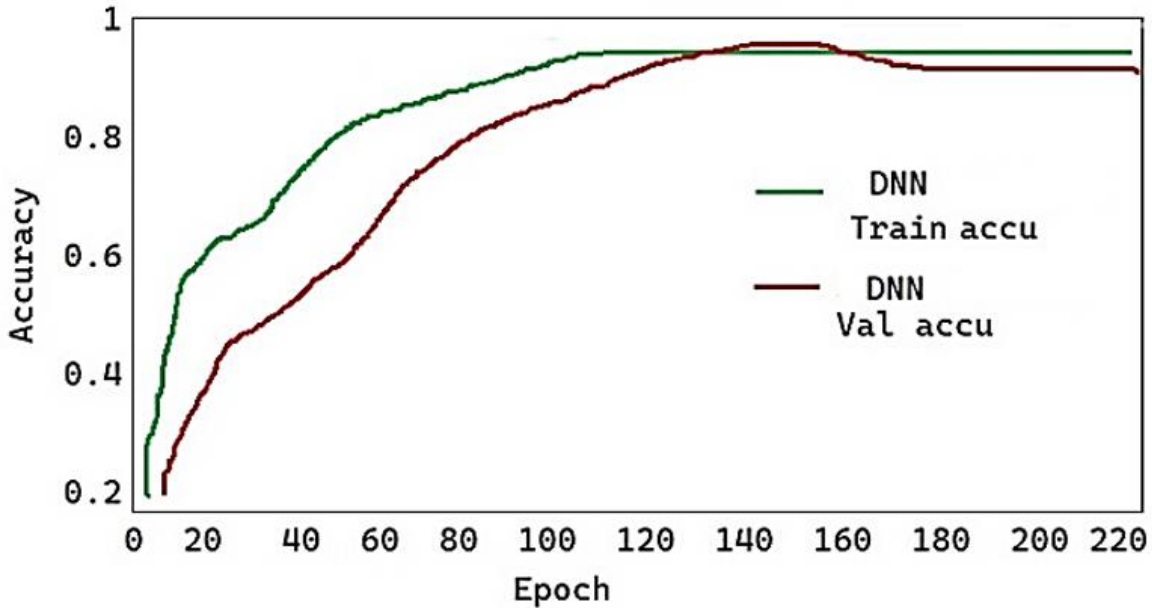

Figure: 7.1 Accuracy of the DNN model

The Figure: 7.1 displays a graph plotting the accuracy of a Deep Neural Network (DNN) over epochs for both training and validation datasets. The green line represents the training accuracy, which increases and plateaus near 1, indicating high performance on the training data. The red line represents the validation accuracy, which also increases but plateaus at a lower value than the training accuracy, suggesting some overfitting. The x-axis is labelled "Epoch" and the y-axis is labelled "Accuracy," with the epoch range from 0 to 220 and accuracy from 0.2 to 1.

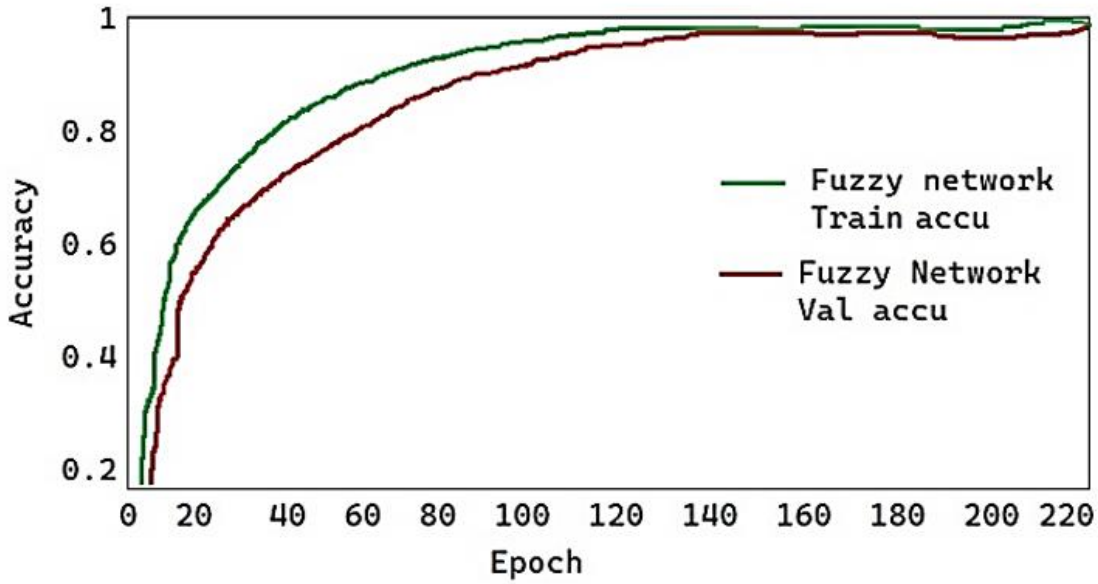

Figure: 7.2 Accuracy of the model with fuzzy logic

The Figure: 7.2 displays a line graph depicting the accuracy of a fuzzy network over epochs for both training and validation datasets. The green line represents the training accuracy, which increases sharply and then plateaus close to 1, indicating a high level of accuracy on the training data. The red line represents the validation accuracy, which follows a similar trend but plateaus at a slightly lower level than the training accuracy, suggesting a good fit but with a slight overfitting to the training data. The x-axis is labelled "Epoch" ranging from 0 to 220, and the y-axis is labelled "Accuracy" ranging from 0.2 to 1.

The proposed model along with the fuzzy neural network is compared with the other baseline model. When compared to alternative baseline models, the suggested model performs better. Table 1 presents the comparison of the planned model with other models. Table 6.1 provides a comparison of the newly proposed method to previously used techniques by taking into account several performance indicators such as the testing accuracy, precision, and F1Score.

Model	Precision	Accuracy	F1 Score
LSTM	0.75	0.72	0.76
SVM	0.56	0.54	0.52
CNN	0.85	0.86	0.85
DenseNet	0.89	0.87	0.88
AlexNet	0.90	0.92	0.90
GoogleNet	0.93	0.94	0.94
Proposed model	0.96	0.97	0.96

Table: 7.1 We compared our new network to other models using performance metrics.

The fig.7.3 shows the comparison graph with other baseline models. The proposed model produces the highest accuracy rate compared with other models.

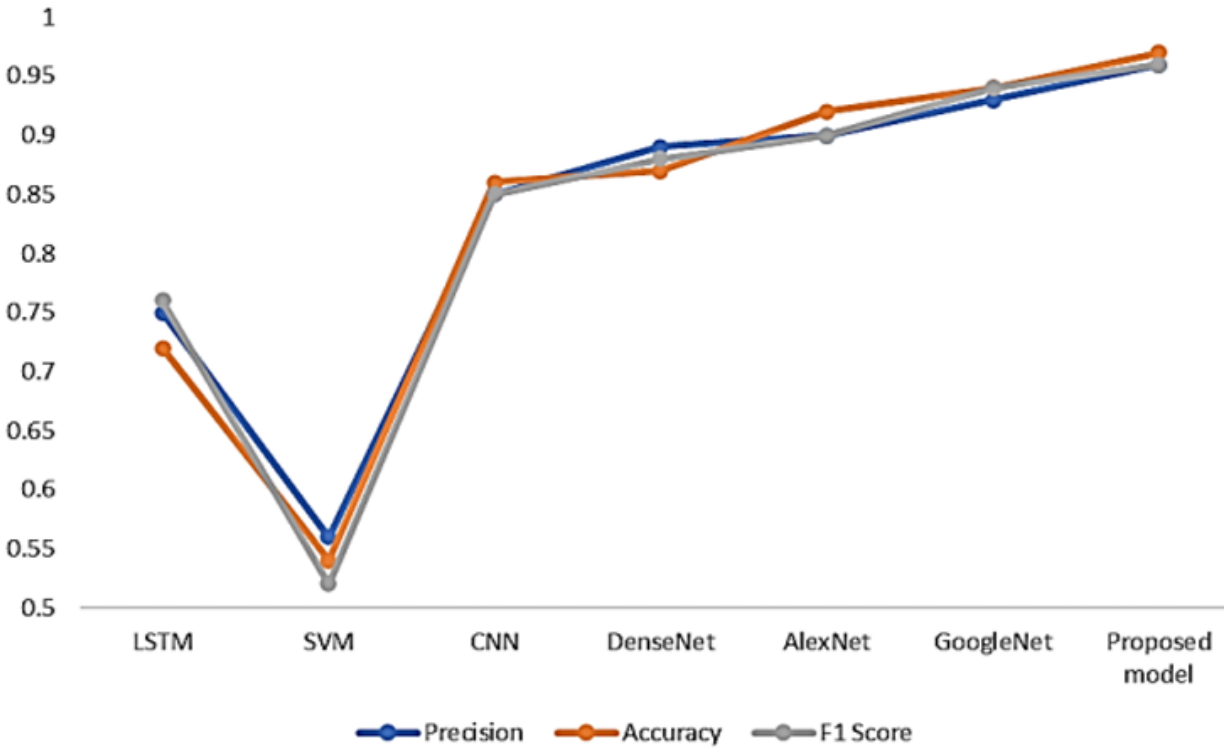

Figure: 7.3 Comparison chart of the proposed model

8. SCREENSHOTS

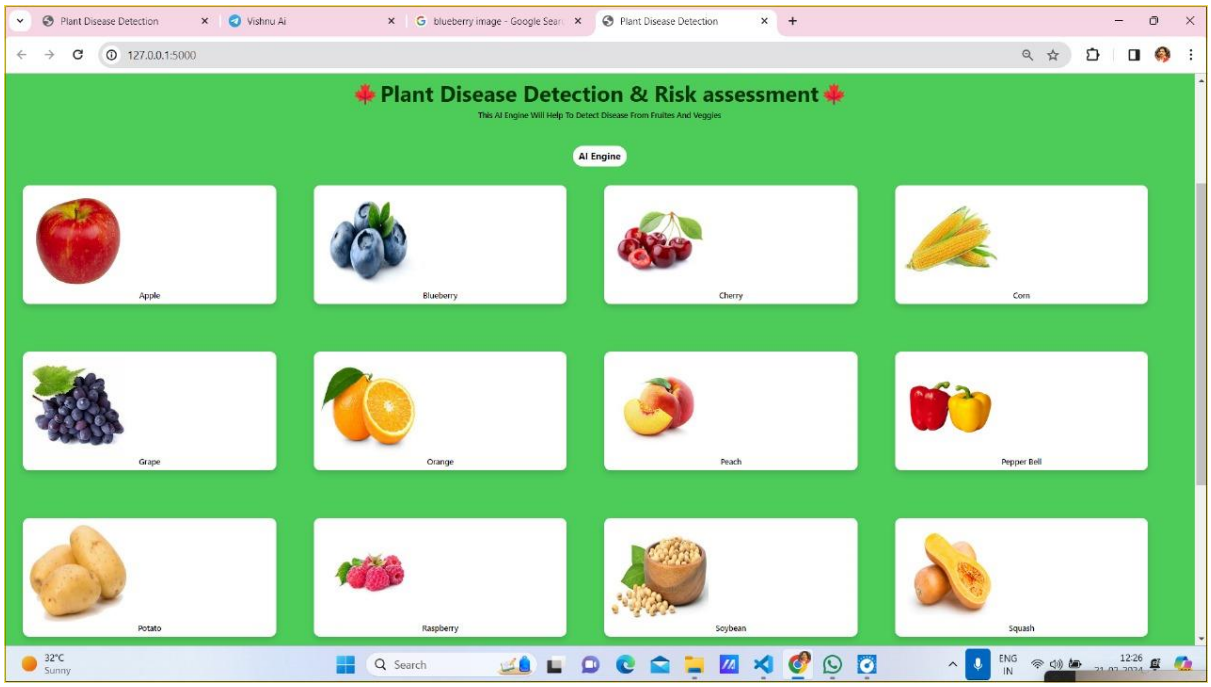

Figure: 8.1 Illustrates Home Screen

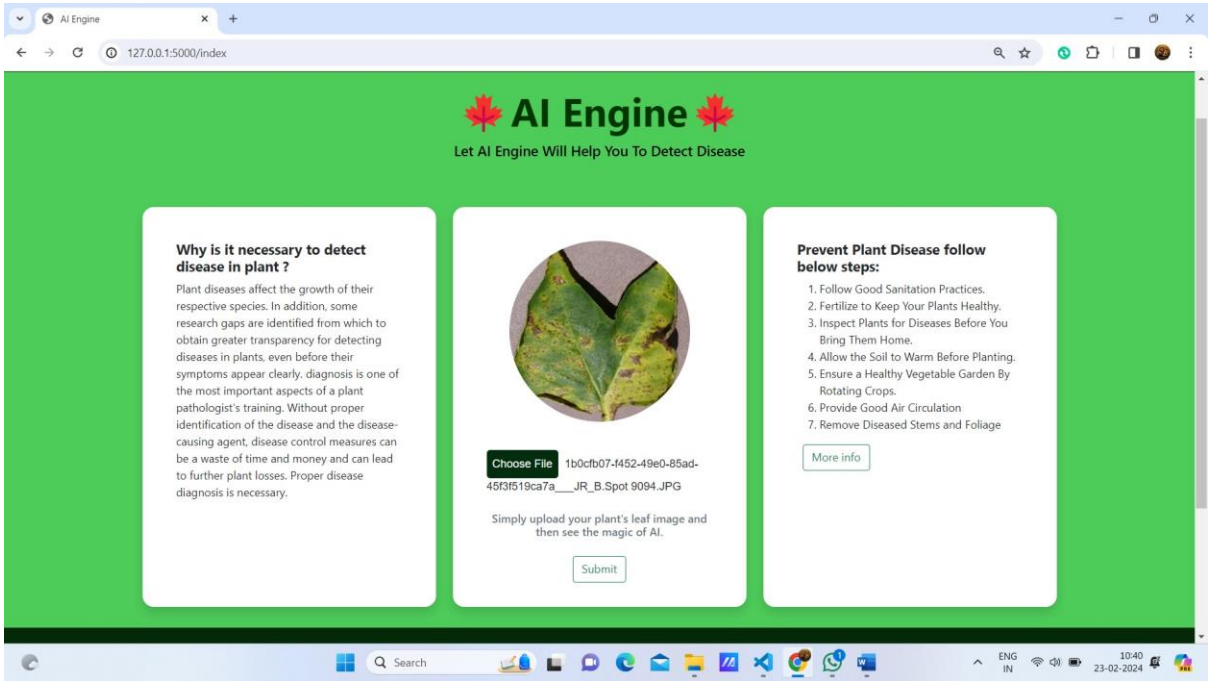

Figure: 8.2 Illustrates AI Engine (Input) Screen

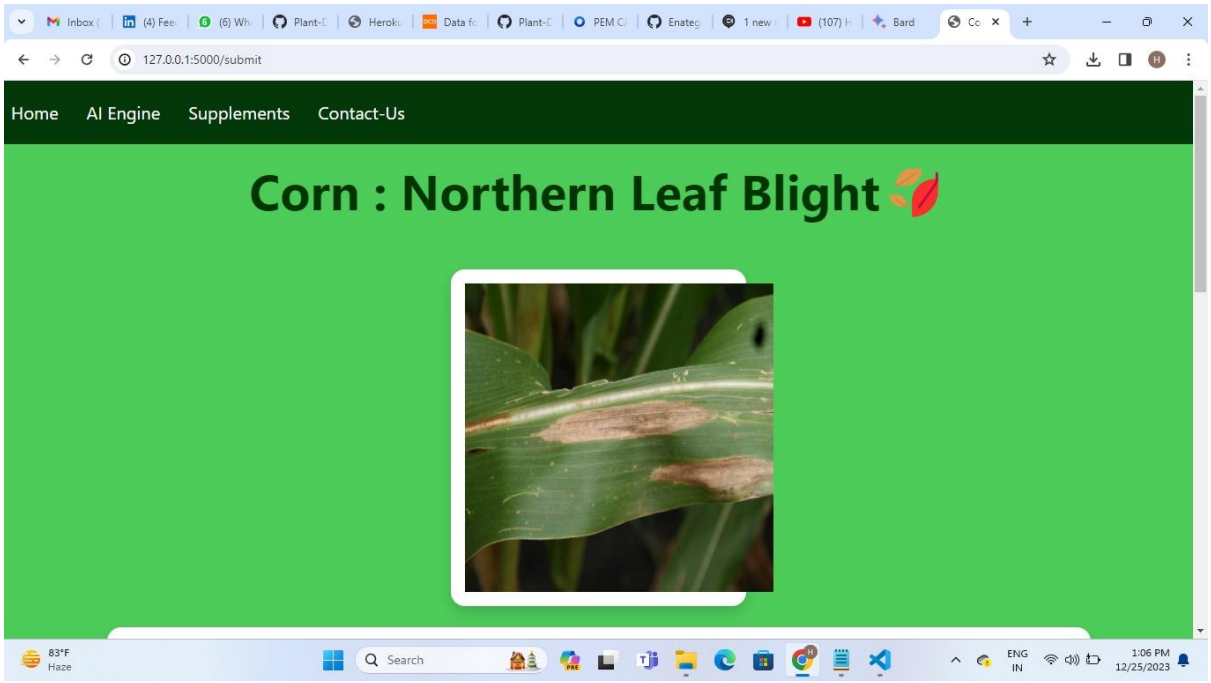

Figure: 8.3 Illustrates the disease of Corn plant (Output) screen

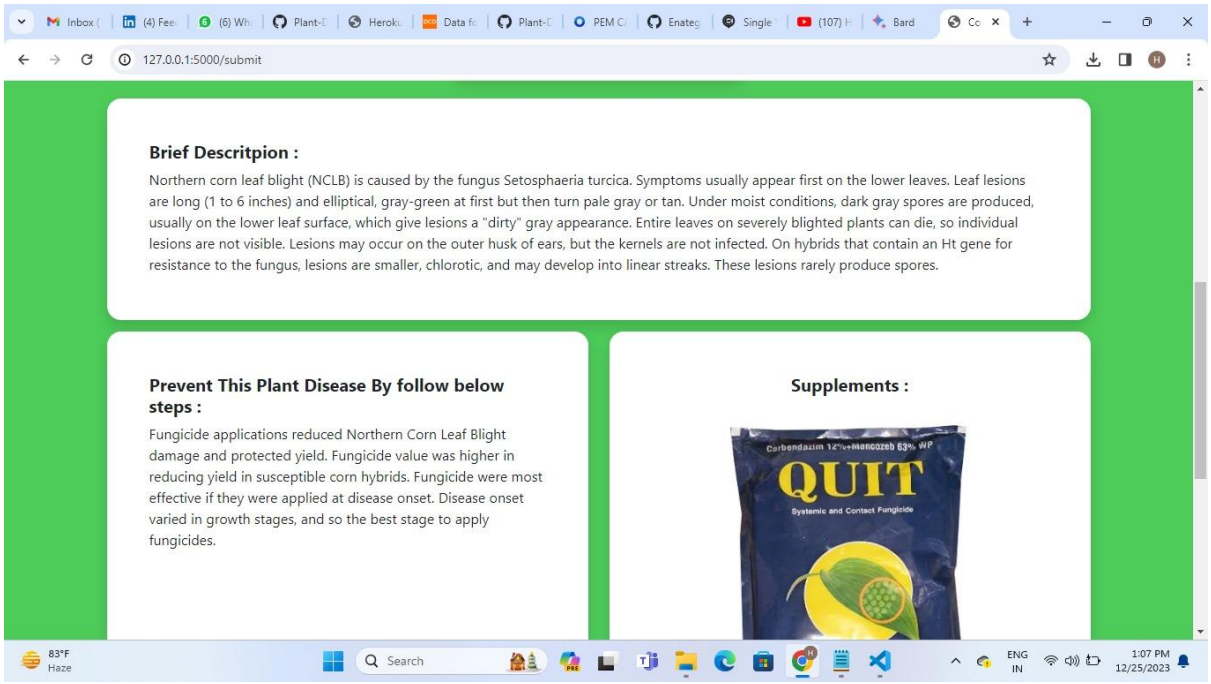

Figure: 8.4 Illustrates the Description about disease and provide Supplements Screen

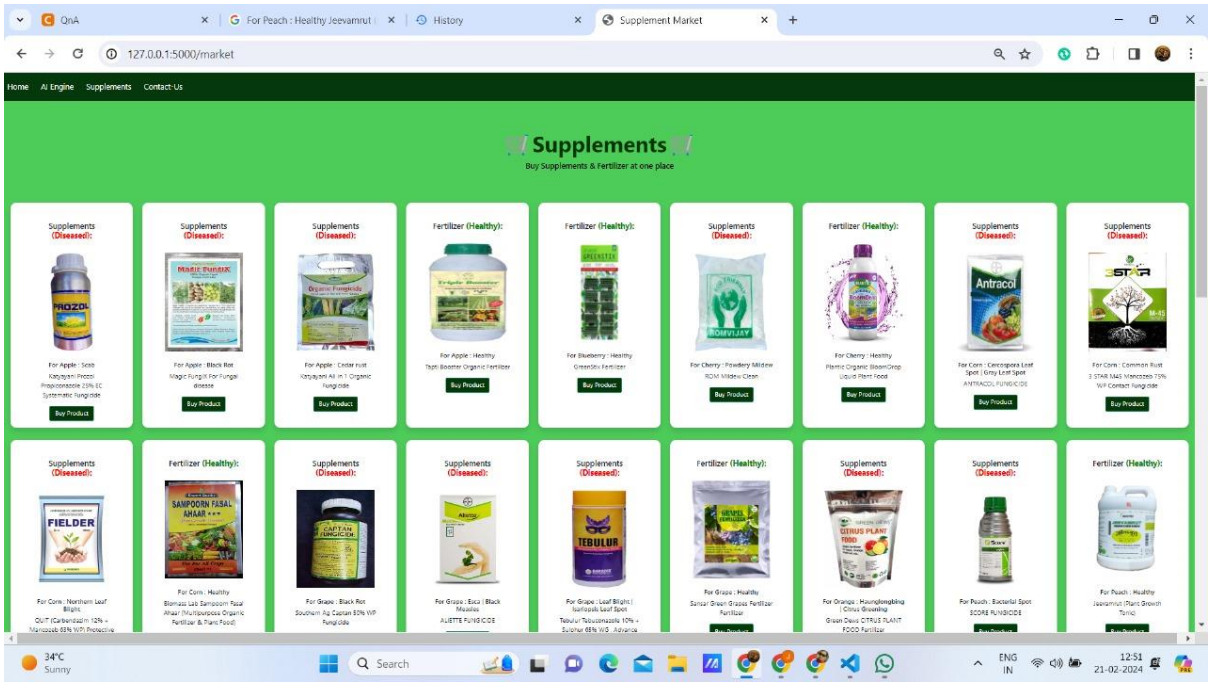

Figure: 8.5 Illustrates the Supplements

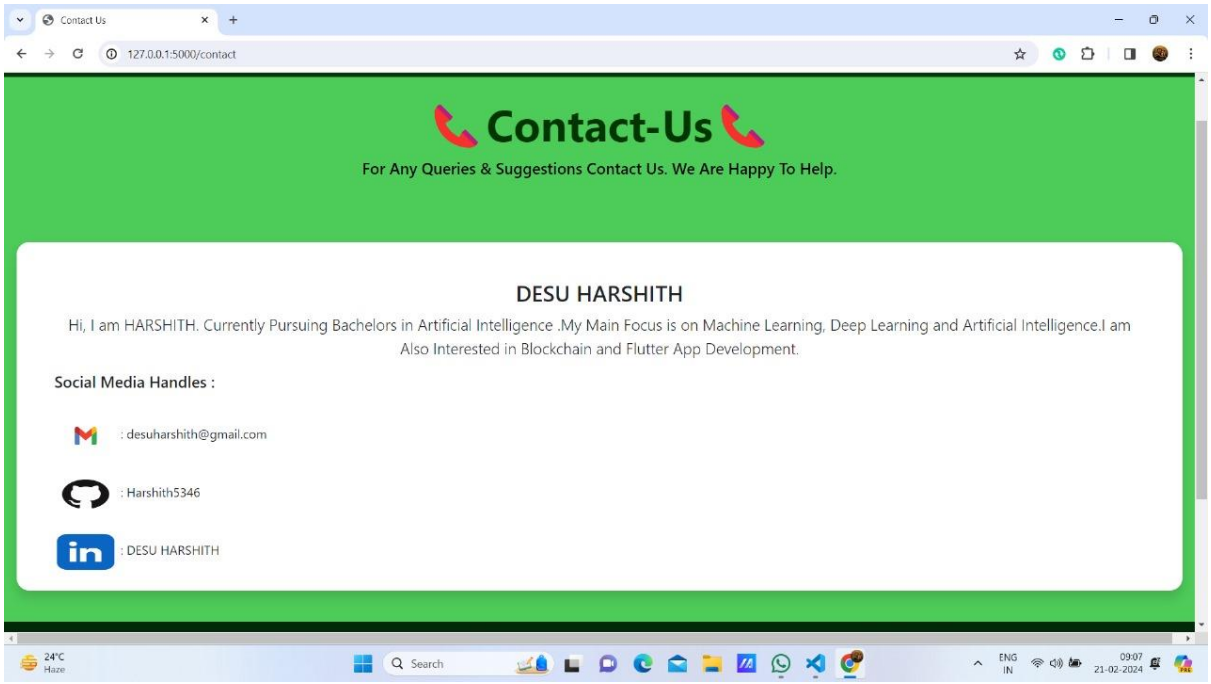

Figure: 8.6 Illustrates Contact-Us Screen

9. CONCLUSION

This study presents an effective approach for detecting and classifying diseases in agriculture plant leaves, termed NFNHDL-based Deep Learning. The authors developed this method, employing RoI extraction in the pre-processing unit to eliminate unwanted noise and distortion from input images. The algorithm isolates damaged areas from the pre-processed output, followed by feature extraction, encompassing statistical, CNN, and textual features, along with data augmentation. Subsequently, a Deep Fuzzy neural network classifies the image as healthy or diseased. In the final step, the NFNHDL classifier identifies specific leaf diseases such as BLB, blast, or brown spot, evaluating the network's effectiveness through metrics like accuracy, recall, and F1-score for each category.

To validate the network's benefits in detecting harmful agriculture plant illnesses, tests assessed the impact of the deep structure, fuzzy inference layer, and overall network structure on recognition accuracy. Optimal combinations of learning rate and batch size were sought to maximize recognition accuracy.

In summary, deploying a deep neuro-fuzzy network for agriculture plant illness diagnosis proves to be a practical, reliable, and outstanding approach with high accuracy. Furthermore, the NFNHDL-based Deep Learning approach demonstrated superior performance in testing accuracy, sensitivity, and specificity, with values of 0.96, 0.97, and 0.96, respectively. Future efforts will focus on optimizing strategies to enhance classification results and evaluating the proposed technique using larger datasets, considering other disorders.

10. FUTURE SCOPE

For agriculture plant disease detection, we used 20,639 images, there are 997 images showing diseased Pepper bell leaves, 2,000 images featuring diseased Potato leaves, and 14,421 images displaying diseased Tomato leaves. Additionally, there are 1,478 images showing healthy Pepper bell leaves, 152 images featuring healthy Potato leaves, and 1,591 images displaying healthy Tomato leaves in our work, we used transfer learning techniques for disease detection which can predict diseased leaves accurately. For this model building we have used three transfer learning techniques those are Inception V3, ResNet50, RestNet152V2. By using these three algorithms we have built three models. Out of these three ResNet152V2 has given most accuracy i.e. 99.6%. Compared to other training models that we have built. In the future, we plan to work with 3D plant images, achieve more efficient agriculture plant diseases detection. Working with a larger dataset will be more challenging in this aspect, and we want to build a dataset emphasizing the abstract with respect to our country which will accelerate the scope of our work.

We can extend the project for classification of different types of agriculture plant diseases, but in order to get more accurate results, we need to have a greater number of items in training dataset. We can make the training set after making the site usable for everyone. As the image uploaded by user for prediction at this stage can be taken into consideration for making them as training set data so that we can train them and make results more accurate.

11. REFERENCES

1. S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 768-771, doi: 10.1109/ICCUBEA.2015.153.
2. S. C. Madiwalar and M. V. Wyawahare, "Plant disease identification: A comparative study," 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 2017, pp. 13-18, doi: 10.1109/ICDMAI.2017.8073478.
3. Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. Computers and Electronics in Agriculture, 137, 52-58.
4. Yin H, Gu YH, Park C, Park J, Yoo SJ (2020) Transfer learning-based search model for hot pepper diseases and pests. Agriculture 10:439
5. Ashwinkumar, S., Rajagopal, S., Manimaran, V., & Jegajothi, B. (2022). Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. Materials Today: Proceedings, 51, 480-487.
6. P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka and E. Hernandez, "Plant Disease Detection Using Hyperspectral Imaging," 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp. 1-8, doi: 10.1109/DICTA.2017.8227476.
7. C. Lu, S. Gao, Z. Zhou, Maize disease recognition via fuzzy least square support vector machine. J. Inf.Comput. Sci. 8(4), 316–320 (2013)
8. V. Barra, J.Y. Boire, Automatic segmentation of subcortical brain structures in MR images using information fusion. IEEE Trans. Med. Imaging 20(7), 549–558 (2001)
9. V. Barra, J.Y. Boire, A general framework for the fusion of anatomical and functional medical images. Neuroimage 13(3), 410–424 (2001)
10. M.J. Hsu, Y.H. Chien, W.Y. Wang et al., A convolutional fuzzy neural network architecture for object classification with small training database. Int. J. Fuzzy Syst. 22(1), 1–10 (2020).
11. G.J. Klir, where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like? Fuzzy Sets Syst. 24(2), 141 160 (1987).
12. S. D.M., Akhilesh, S. A. Kumar, R. M.G. and P. C., "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0645-0649, doi: 10.1109/ICCSP.2019.8698007.

13. G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," 2020 IEEE Applied Signal Processing Conference (ASPCON), 2020, pp. 109-113, doi: 10.1109/ASPCON49795.2020.9276722.
14. Latif, G.; Alghazo, J.; Maheswar, R.; Vijayakumar, V.; Butt, M. Deep Learning Based Intelligence Cognitive Vision Drone for Automatic Plant Diseases Identification and Spraying. J. Intell. Fuzzy Syst. 2020, 39, 8103–8114.
15. J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23(3), 665–685 (1993)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-163

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that **DESU HARSHITH** , NARASARAOPETA ENGINEERING COLLEGE has presented the paper title **DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P.,

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-163

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Noorbasha Janibasha, NARASARAOPETA ENGINEERING COLLEGE has presented the paper title **DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-163

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Papisetti Vishnu Vardhan, NARASARAOPETA ENGINEERING COLLEGE has presented the paper title **DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P.,

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-163

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Badisa Veera Brahmam, NARASARAOPETA ENGINEERING COLLEGE has presented the paper title **DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P.,

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING

Desu Harshith
Student
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
desuharshith@gmail.com

Noorbasha Janibasha
Student
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
janibasha731@gmail.com

Papisetti Vishnu Vardhan
Student
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
vishnuvardhan630sap@gmail.com

Badisa Veera Brahmam,
Assistant Professor
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
veerabrahmam265@gmail.com

Abstract*— *Significant risks to agricultural production come from plant diseases, which have an effect on food yields and the stability of the economy. Conventional illness detection techniques that depend on human observation are frequently expensive, time-consuming, and prone to errors. On the other hand, automatic detection that makes use of image processing techniques provides accurate and timely results, signaling a potentially fruitful path towards improving plant protection in agriculture. This research uses a DCNFN (Deep Convolution Neuro-Fuzzy Network) to provide a novel method for disease recognition in tomato, pepper bell, and potato plants. Our methodology, which makes use of recent developments in computer vision, intends to transform precision agriculture by providing a powerful tool for the detection, evaluation, and classification of plant diseases. The creation of a deep learning framework for the purpose of training and optimizing the DCNFN model, the establishment of a comprehensive collection of plant photos, and the professional evaluation of disease signs are crucial phases in our implementation. One notable aspect of our model is its simplicity: background photos and healthy leaves are smoothly incorporated into the training dataset, making it possible to discern between healthy and diseased foliage. Our findings open the door for the DCNFN model's practical application in agricultural settings by demonstrating its effectiveness in precisely diagnosing and categorizing plant diseases across a wide range of species. Our technique has great potential to improve crop resilience, optimize resource allocation, and ultimately protect global food security, as it provides a comprehensive solution for identifying diseases and evaluating risks.

Keywords: Deep Learning, fuzzy rules, neuro-fuzzy, inference layer, and plant leaves

I. INTRODUCTION

The vital plants used in agriculture are those that give billions of people worldwide access to different types of proteins and vitamins [1]. In Indonesia and India, it serves as the main source of nutrition. Anything that has an impact on the plant crop has an international effect, regardless of its superiority or measure. To effectively address the issue, frequent illness identification and monitoring are therefore crucial. If illnesses are not treated promptly, they could have a negative effect on output [2]. The decline of agricultural crops in recent years has been attributed to a wide range of illnesses and pests. These attacks have the potential to seriously disrupt farms in the absence of adequate management and surveillance, and the issue has only gotten worse recently. Furthermore, the early and precise detection of plant issues reduces losses in terms of money and protects plants from harmful diseases. The management of these illnesses aids in our attempts to maintain a healthy agricultural sector [1].

In the agricultural industry, the capacity to recognise plant diseases automatically from plant leaves is a significant advancement. The efficiency and quality of harvests may be increased by early and precise identification of plant leaf diseases. Stopping disease from spreading across the farm is essential to maximise crop superiority. If a disease's symptoms are identified, its causes of spread are looked into, and control measures are implemented, the disease may be contained. Plants can experience issues at any point in their life cycle due to plant diseases.

Weak plants frequently have wilted, discoloured leaves. In general, abnormalities in a disease's visual presentation can be identified by searching for its distinctive pattern. Since a plant's leaves are typically the first to exhibit symptoms of illness, they are a valuable tool for identifying plant diseases [3]. Recent effective applications of deep learning techniques include semantic segmentation and picture identification. Furthermore, a variety of plant diseases have been diagnosed using this technique [4-6]. When we talk about "Deep Learning," "Deep" refers to the quantity of transformational layers that are applied to data, each of which extracts progressively finer features from the source. The input, hidden, and output layers comprise the three levels of nodes that make up the Deep Learning algorithm. It executes a superb classification process by picking up on more intricate leaf data properties. The convolutional neural network is a full-stack solution; its input is the raw picture data, and its output is the model's learned grouping. Neither expertly designed algorithms nor human input are required in the centre, where learning takes

place. The input, hidden, and output layers comprise the three levels of nodes that make up the Deep Learning algorithm. It executes a superb classification process by picking up on more intricate leaf data properties. The convolutional neural network is a full-stack solution; its input is the raw picture data, and its output is the model's learned classification. Neither expertly designed algorithms nor human input are required in the centre, where learning takes place[11].

Improvements in digital image processing and identification technologies have made it possible to quickly identify sick crops and identify the precise disease that has affected them [12,13]. But AI and ML by themselves won't cut it. As a result, some scholars have suggested utilising cloud services, IoT, surveillance drones, and other technologies to build a comprehensive system that can help farmers save costs and achieve positive results [14]. But the most important element would be a machine learning algorithm, method, or process that is very effective and capable of correctly identifying and diagnosing plant diseases. Because of this, scientists are continuously searching for the best machine learning technique to diagnose plant illnesses. Despite a recent study in this field, researchers are always trying to determine the best and most accurate response, thus they are always making progress in this direction.

An adaptable neuro-fuzzy inference system was developed by Jang et al. [15]. The TSK fuzzy model parameter is computed using a neural network-based technique. Although it is a bit simplistic, this method is a well-liked neuro-fuzzy modelling strategy that concentrates on a single rule set. We propose a deep neuro-fuzzy network for tomato leaf disease detection. This network integrates fuzzy implication rules into a deep framework by fusing deep learning with fuzzy systems. The new fuzzy pooling and fuzzy inference layers enhance the extraction of information from tomato leaf images by offering both fuzzy and clear values for higher recognition accuracy. An end-to-end network with multiple hidden layers ensures better performance and generalisation while processing photos of tomato leaves and extracting relevant information.

II. LITERATURE SURVEY

A prediction is an assertion about what one thinks will happen in the future. Many predictions are made every day. Some are purely conjectural, while others are incredibly important and grounded in mathematics. Anticipating future events, whether they occur in several months, a year, or a decade, can be beneficial in several aspects. The study and analysis of the application of deep learning to improve and detect plant disease is the goal of "Implementation Of Deep Convolution Neuro-Fuzzy Network To Detect Plant Diseases, Risk Assessment And Classification Using Deep Learning". This research scans a leaf image to detect if the plant is disease-free or sick using an automated vision system and image processing technique. The farmer is then given information on how to determine whether the plant is ill and even how to treat it.

III. DESIGN OF PROPOSED MODEL

The Process of detecting the plant disease is shown in Fig: 1. Each and every phase has different step as seen in Fig.

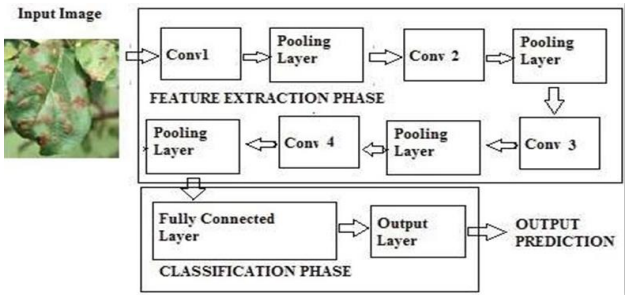

Fig 1: A Synopsis of the Suggested Design Process

Based on Fig. 1, Upon getting an image of a plant, the system will determine whether or not it is infected by comparing it to the training dataset. The system will recommend actions to the farmer if the expected outcome shows a sick plant or leaf. These recommendations mostly concentrate on diagnosing and treating the plant disease. By taking a preventative measure, the farmer may protect their plants from diseases, which will ultimately save time, labour, and money.

Technology Used:

The technology used to develop this project is Deep Learning.

Deep Learning:

Deep Learning is a subfield of artificial intelligence and computer science that aims to increase system accuracy by simulating human learning with data and algorithms. Depending on the issues it tackles, it has been divided into several branches. Deep learning comprises a wide range of methods, each designed for particular use cases and industry difficulties.

Algorithms used in this Project:

In this project we have used several algorithms and techniques would likely be used in combination to create a comprehensive solution for detection of plant diseases , risk assessment, and classification.

A. Convolutional Neural Networks (CNNs)

- CNNs are specialised deep learning models made to handle structured input in the form of grids, such images. Their proficiency in feature extraction and hierarchical representation learning makes them ideal for image classification applications, such as the detection of plant illnesses from leaf photos.
- One kind of deep learning model called convolutional neural networks (CNNs) is made especially for handling structured, grid-like data, like photographs. They are made up of several layers of neurons that carry out functions including convolution, pooling, and nonlinear activation; they are inspired by the human visual system. The fundamental principle of CNNs is their ability to automatically extract hierarchical feature representations from unprocessed input data.
- Convolutional layers, which apply convolution processes to input data, are the fundamental components of CNNs. In order to do these processes,

a tiny filter, is slid across the input image, sometimes called a kernel, and the dot products between the filter and certain regions of the image are calculated. The network can now record spatial hierarchies of features, from lower-level elements like textures and edges to higher-level elements like forms and objects, thanks to this process.

- Convolutional layers in a CNN apply filters, or kernels, to the input image to detect patterns and features such as shapes, edges, and textures.

B. Neuro-Fuzzy Systems

- Neuro-fuzzy systems are hybrid models that can handle both symbolic knowledge representation (fuzzy logic) and numerical data processing (neural networks). They achieve this by fusing neural network and fuzzy logic components. In your project, a neuro-fuzzy system might leverage fuzzy logic reasoning in conjunction with CNN feature extraction power to enhance the interpretability and robustness of the model.
- Fuzzy logic is utilized in a neuro-fuzzy system to simulate linguistic variables, membership functions, and fuzzy rules in order to simulate human-like reasoning. Neural networks, on the other hand, have the computational capacity to learn from data and adaptively modify the fuzzy logic components according to the input-output relationships seen in the training set. This integration allows for the combination of fuzzy logic's interpretability and neural networks' learning capabilities to create neuro-fuzzy systems.

C. Supervised Learning

- The model will be trained using supervised learning methods since we most likely have tagged data pictures of plants labelled with the diseases that correlate to them. Neural network training procedures like backpropagation, which adjusts the model's parameters to lessen the difference between expected and actual outcomes, could be used for this.

- During training, the supervised learning algorithm adjusts its internal parameters using optimisation techniques in an effort to reduce the discrepancy between the expected outputs and the actual labels in the training data. In this process, the difference between the true and predicted values is quantified by iteratively updating the model's parameters using a chosen loss function. Gradient descent and its variations are popular optimisation methods used in supervised learning applications.

- A variety of tasks, such as regression, structured prediction, and classification, are included in supervised learning. Whereas the model predicts continuous numerical values in regression tasks, it predicts discrete class labels for input examples in classification tasks.

IV. DATE SET AND IMAGE PRE-PROCESSING

Data Set:

From this detection of plant disease, The dataset that we use for the detection of plant disease was taken from the Kaggle repository which contain 20,639 images out of which 997 images belong to diseased Pepper bell's leaves, 2,000 images belong to diseased Potato's leaves, 14,421 images belong to diseased Tomato's leaves, 1,478 images belong to healthy Pepper bell's leaves. 152 images belong to healthy Potato's leaves, and 1,591 images belong to healthy Tomato's leaves.

Fig 2 and 3 shows both healthy and diseased pepper bell plant leaves.

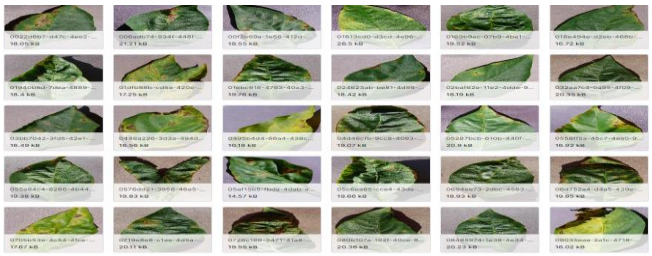

Fig 2: Pepper bell plant Diseased leaves

Fig 3: Pepper bell plant Healthy leaves

Fig 4 and 5 shows both healthy and diseased potato plant leaves.

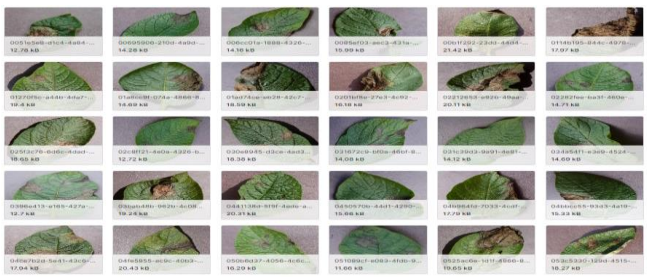

Fig 4: Potato plant Diseased leaves

Fig 5: Potato plant Healthy leaves

Fig 6 and 7 shows both healthy and diseased tomato plant leaves.

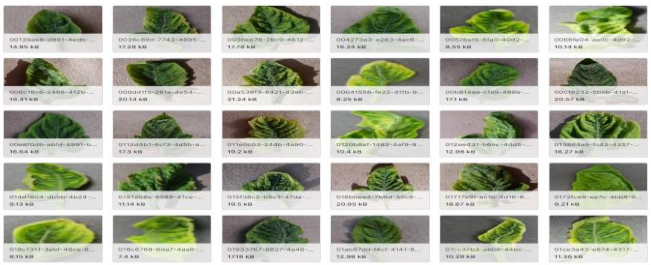

Fig 6: Tomato plant Diseased leaves

Fig 7: Tomato plant Healthy leaves

Image Pre-processing:

Image pre-processing stands as a vital initial phase in both computer vision and image processing applications, where a myriad of techniques are deployed to refine raw images, thereby bolstering their quality and priming them for subsequent analysis or manipulation. Concurrently, techniques such as image normalization aid in standardizing pixel values across images, facilitating fair comparisons and optimizing model performance. Moreover, noise reduction techniques strive to mitigate unwanted distortions or aberrations introduced during image acquisition or transmission, thereby preserving crucial image features. Alongside noise reduction, image enhancement techniques are leveraged to amplify desired characteristics or details within images, enhancing their interpretability or aesthetic appeal. Integral to image pre-processing is the task of edge detection, which involves identifying boundaries **or** transitions between objects within images, thus facilitating subsequent segmentation or feature extraction tasks. In essence, image pre-processing orchestrates a symphony of operations, harmonizing raw inputs into refined data ready for the symposium of computer vision algorithms. Fig. 8 illustrates the image pre-processing procedure.

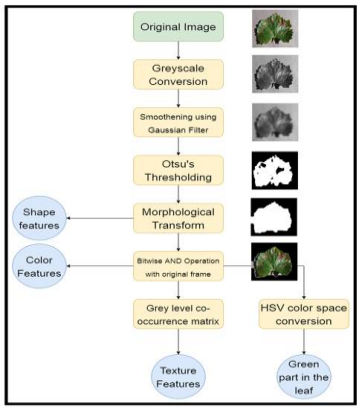

Fig 8: Image Pre-processing

V. RESULTS & PERFORMANCE EVALUATION

A deep neuro fuzzy network's prediction accuracy depends on the batch size and training rate. To find the ideal combination of these attributes, we experimented with various starting learning rates and batch sizes while monitoring changes in the learning rate. Figure 9 displays a line graph titled "Training and Validation loss." On the graph, the blue line signifies the training loss and the red line the validation loss. The label on the x-axis is labelled, but it is not visible; it probably indicates how many epochs or iterations were used to train the machine learning model. The loss value, which shows the model's mistake during training, is represented by the y-axis. The model is learning and getting better over time at using the training data, as shown by the blue line (training loss), which starts at a larger value and gradually trends downward. In addition to showing a decreasing trend, the red line (validation loss) also has notable spikes at specific epochs that may indicate overfitting at those times when the model performs well on training data but badly on unknown validation data.

The graph's x-axis, which has numbers ranging from 0 to 25, indicates that it spans 25 epochs or iterations. The loss values are represented along the y-axis, which has a range of 0 to 1.75.

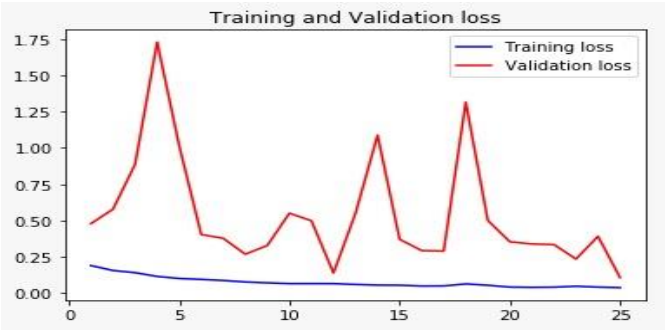

Fig 9: Training and Validation loss

The fig 10 shows a line graph labelled "Training and Validation accuracy," on which two lines are plotted to indicate how accurate a machine learning model is during training. The x-axis represents the epoch number, which runs from 0 to 25, and the y-axis shows the accuracy, which varies from 0.88 to 0.98.

There are two lines on the graph:

A. The blue line represents the training accuracy. It starts off just below 0.98 and remains relatively stable throughout the training process, with only minor fluctuations.

B. The red line represents the validation accuracy. This line starts at approximately 0.91, shows more variability than the training accuracy, dips slightly below 0.90, and then generally trends upward, reaching around 0.96 by the 25th epoch.

The model's performance is assessed using the graph; ideally, the training and validation accuracy should be high and near to one another. A point of examination for more model tuning may be the fluctuations and the difference between the two lines, which could hint to problems like overfitting or underfitting.

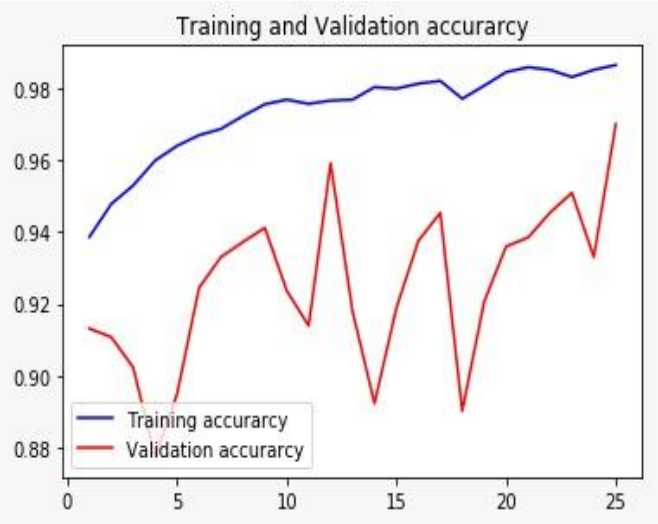

Fig 10: Training and Validation accuracy

TABLE I.

WE USED PERFORMANCE INDICATORS TO COMPARE OUR SUGGESTED NETWORK TO OTHER MODELS.

Model	Precision	Accuracy	F1 Score
Long Short-Term Memory	0.74	0.71	0.77
Support Vector Manchine	0.55	0.57	0.50
Convolutional Neural Network	0.83	0.83	0.83
GoogleNet	0.93	0.94	0.94
DenseNet	0.87	0.85	0.86
AlexNet	0.89	0.90	0.91
Proposed model	0.96	0.97	0.96

A comparison graph with more baseline models is shown in Fig. 11. Among the models that we have proposed, ours has the highest accuracy rate.th

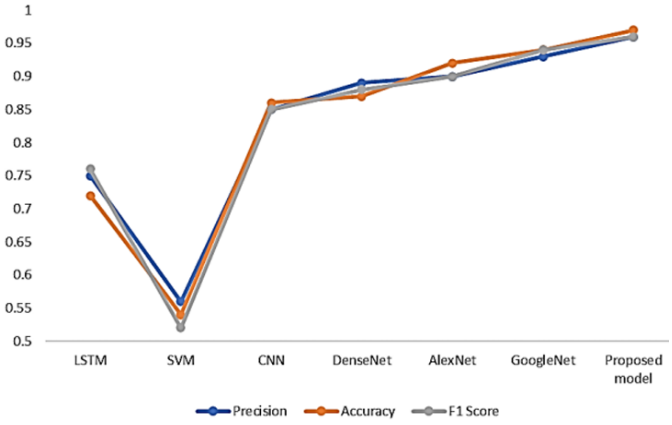

Fig 11: A comparison table for the suggested model

VI. CONCLUSION

This research presents NFNHDL-based Deep Learning, a productive technique for classifying and diagnosing diseases in agricultural plant leaves. The method developed by the authors uses ROI extraction in the pre-processing unit to eliminate unwanted noise and distortion from input images. After removing any damaged portions from the output after preprocessing, the approach extracts features (textual, statistical, and CNN features) and enhances the data. The state of the image is subsequently evaluated by use of a Deep Fuzzy neural network. The NFNHDL classifier determines which particular leaf diseases, such as brown spot, blast, or BLB, to recognise in the last stage depending on variables including For every category, the F1-score, accuracy, and recall.

In conclusion, it turns out that using a deep neuro-fuzzy network to diagnose diseases in agricultural plants is a great, dependable, and highly accurate method. Additionally, with scores of 0.96, 0.97, and 0.96 in evaluating accuracy, sensitivity, and specificity, respectively, the NFNHDL-based Deep Learning technique performed better. Subsequent endeavours will centre around refining tactics to augment classification outcomes and assessing the suggested methodology with more extensive datasets, taking into account additional diseases.

REFERENCES

[1] S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," in 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 768-771, doi: 10.1109/ICCUBEA.2015.153.

[2] S. C. Madiwalar and M. V. Wyawahare, "Plant disease identification: A comparative study," in 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 2017, pp. 13-18, doi: 10.1109/ICDMAI.2017.8073478.

[3] Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. Computers and Electronics in Agriculture, 137, 52-58.

[4] Yin H, Gu YH, Park C, Park J, Yoo SJ (2020) Transfer learning-based search model for hot pepper diseases and pests. Agriculture 10:439

[5] Ashwinkumar, S., Rajagopal, S., Manimaran, V., & Jegajothi, B. (2022). Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. Materials Today: Proceedings, 51, 480-487.

[6] P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka, and E. Hernandez, "Plant Disease Detection Using Hyperspectral Imaging," in 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp. 1-8, doi: 10.1109/DICTA.2017.8227476.

[7] C. Lu, S. Gao, Z. Zhou, Maize disease recognition via fuzzy least square support vector machine. J. Inf. Comput. Sci. 8(4), 316–320 (2013)

[8] V. Barra, J.Y. Boire, Automatic segmentation of subcortical brain structures in MR images using information fusion. IEEE Trans. Med. Imaging 20(7), 549–558 (2001)

[9] V. Barra, J.Y. Boire, A general framework for the fusion of anatomical and functional medical images. Neuroimage 13(3), 410–424 (2001)

[10] M.J. Hsu, Y.H. Chien, W.Y. Wang et al., A convolutional fuzzy neural network architecture for object classification with small training database. Int. J. Fuzzy Syst. 22(1), 1–10 (2020).

[11] G.J. Klir, "Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?" Fuzzy Sets Syst. 24(2), 141–160 (1987).

DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING.pdf

by A a

Submission date: 04-Apr-2024 09:39AM (UTC-0400)

Submission ID: 2329593544

File name: DETECTING_PLANT_DISEASES_WITH_DEEP_CONVOLUTION_NEURO-FUZZY_NETWORKS_USING_DEEP_LEARNING.pdf (644.37K)

Word count: 3330

Character count: 18810

DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING

Desu Harshith
Student
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
desuharshith@gmail.com

Noorbasha Janibasha
Student
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
janibasha731@gmail.com

Papisetti Vishnu Vardhan
Student
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
vishnuvardhan630sap@gmail.com

Badisa Veera Brahmam,
Assistant Professor
CSE(ARTIFICIAL INTELLIGENCE)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh, India
veerabrahmam265@gmail.com

Abstract*— *Significant risks to agricultural production come from plant diseases, which have an effect on food yields and the stability of the economy. Conventional illness detection techniques that depend on human observation are frequently expensive, time-consuming, and prone to errors. On the other hand, automatic detection that makes use of image processing techniques provides accurate and timely results, signaling a potentially fruitful path towards improving plant protection in agriculture. This research uses a DCNFN (Deep Convolution Neuro-Fuzzy Network) to provide a novel method for disease recognition in tomato, pepper bell, and potato plants. Our methodology, which makes use of recent developments in computer vision, intends to transform precision agriculture by providing a powerful tool for the detection, evaluation, and classification of plant diseases. The creation of a deep learning framework for the purpose of training and optimizing the DCNFN model, the establishment of a comprehensive collection of plant photos, and the professional evaluation of disease signs are crucial phases in our implementation. One notable aspect of our model is its simplicity: background photos and healthy leaves are smoothly incorporated into the training dataset, making it possible to discern between healthy and diseased foliage. Our findings open the door for the DCNFN model's practical application in agricultural settings by demonstrating its effectiveness in precisely diagnosing and categorizing plant diseases across a wide range of species. Our technique has great potential to improve crop resilience, optimize resource allocation, and ultimately protect global food security, as it provides a comprehensive solution for identifying diseases and evaluating risks.

Keywords: *Deep Learning, fuzzy rules, neuro-fuzzy, inference layer, and plant leaves*

I. INTRODUCTION

The vital plants used in agriculture are those that give billions of people worldwide access to different types of proteins and vitamins [1]. In Indonesia and India, it serves as the main source of nutrition. Anything that has an impact on the plant crop has an international effect, regardless of its superiority or measure. To effectively address the issue, frequent illness identification and monitoring are therefore crucial. If illnesses are not treated promptly, they could have a negative effect on output [2]. The decline of agricultural crops in recent years has been attributed to a wide range of illnesses and pests. These attacks have the potential to seriously disrupt farms in the absence of adequate management and surveillance, and the issue has only gotten worse recently. Furthermore, the early and precise detection of plant issues reduces losses in terms of money and protects plants from harmful diseases. The management of these illnesses aids in our attempts to maintain a healthy agricultural sector [1].

In the agricultural industry, the capacity to recognise plant diseases automatically from plant leaves is a significant advancement. The efficiency and quality of harvests may be increased by early and precise identification of plant leaf diseases. Stopping disease from spreading across the farm is essential to maximise crop superiority. If a disease's symptoms are identified, its causes of spread are looked into, and control measures are implemented, the disease may be contained. Plants can experience issues at any point in their life cycle due to plant diseases.

Weak plants frequently have wilted, discoloured leaves. In general, abnormalities in a disease's visual presentation can be identified by searching for its distinctive pattern. Since a plant's leaves are typically the first to exhibit symptoms of illness, they are a valuable tool for identifying plant diseases [3]. Recent effective applications of deep learning techniques include semantic segmentation and picture identification. Furthermore, a variety of plant diseases have been diagnosed using this technique [4-6]. When we talk about "Deep Learning," "Deep" refers to the quantity of transformational layers that are applied to data, each of which extracts progressively finer features from the source. The input, hidden, and output layers comprise the three levels of nodes that make up the Deep Learning algorithm. It executes a superb classification process by picking up on more intricate leaf data properties. The convolutional neural network is a full-stack solution; its input is the raw picture data, and its output is the model's learned grouping. Neither expertly designed algorithms nor human input are required in the centre, where learning takes

place. The input, hidden, and output layers comprise the three levels of nodes that make up the Deep Learning algorithm. It executes a superb classification process by picking up on more intricate leaf data properties. The convolutional neural network is a full-stack solution; its input is the raw picture data, and its output is the model's learned classification. Neither expertly designed algorithms nor human input are required in the centre, where learning takes place[11].

Improvements in digital image processing and identification technologies have made it possible to quickly identify sick crops and identify the precise disease that has affected them [12,13]. But AI and ML by themselves won't cut it. As a result, some scholars have suggested utilising cloud services, IoT, surveillance drones, and other technologies to build a comprehensive system that can help farmers save costs and achieve positive results [14]. But the most important element would be a machine learning algorithm, method, or process that is very effective and capable of correctly identifying and diagnosing plant diseases. Because of this, scientists are continuously searching for the best machine learning technique to diagnose plant illnesses. Despite a recent study in this field, researchers are always trying to determine the best and most accurate response, thus they are always making progress in this direction.

An adaptable neuro-fuzzy inference system was developed by Jang et al. [15]. The TSK fuzzy model parameter is computed using a neural network-based technique. Although it is a bit simplistic, this method is a well-liked neuro-fuzzy modelling strategy that concentrates on a single rule set. We propose a deep neuro-fuzzy network for tomato leaf disease detection. This network integrates fuzzy implication rules into a deep framework by fusing deep learning with fuzzy systems. The new fuzzy pooling and fuzzy inference layers enhance the extraction of information from tomato leaf images by offering both fuzzy and clear values for higher recognition accuracy. An end-to-end network with multiple hidden layers ensures better performance and generalisation while processing photos of tomato leaves and extracting relevant information.

II. LITERATURE SURVEY

A prediction is an assertion about what one thinks will happen in the future. Many predictions are made every day. Some are purely conjectural, while others are incredibly important and grounded in mathematics. Anticipating future events, whether they occur in several months, a year, or a decade, can be beneficial in several aspects. The study and analysis of the application of deep learning to improve and detect plant disease is the goal of "Implementation Of Deep Convolution Neuro-Fuzzy Network To Detect Plant Diseases, Risk Assessment And Classification Using Deep Learning". This research scans a leaf image to detect if the plant is disease-free or sick using an automated vision system and image processing technique. The farmer is then given information on how to determine whether the plant is ill and even how to treat it.

III. DESIGN OF PROPOSED MODEL

The Process of detecting the plant disease is shown in Fig: 1. Each and every phase has different step as seen in Fig.

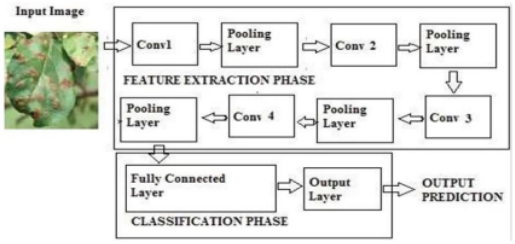

Fig 1: A Synopsis of the Suggested Design Process

Based on Fig. 1, Upon getting an image of a plant, the system will determine whether or not it is infected by comparing it to the training dataset. The system will recommend actions to the farmer if the expected outcome shows a sick plant or leaf. These recommendations mostly concentrate on diagnosing and treating the plant disease. By taking a preventative measure, the farmer may protect their plants from diseases, which will ultimately save time, labour, and money.

Technology Used:

The technology used to develop this project is Deep Learning.

Deep Learning:

Deep Learning is a subfield of artificial intelligence and computer science that aims to increase system accuracy by simulating human learning with data and algorithms. Depending on the issues it tackles, it has been divided into several branches. Deep learning comprises a wide range of methods, each designed for particular use cases and industry difficulties.

Algorithms used in this Project:

In this project we have used several algorithms and techniques would likely be used in combination to create a comprehensive solution for detection of plant diseases , risk assessment, and classification.

A. *Convolutional Neural Networks (CNNs)*

- CNNs are specialised deep learning models made to handle structured input in the form of grids, such images. Their proficiency in feature extraction and hierarchical representation learning makes them ideal for image classification applications, such as the detection of plant illnesses from leaf photos.
- One kind of deep learning model called convolutional neural networks (CNNs) is made especially for handling structured, grid-like data, like photographs. They are made up of several layers of neurons that carry out functions including convolution, pooling, and nonlinear activation; they are inspired by the human visual system. The fundamental principle of CNNs is their ability to automatically extract hierarchical feature representations from unprocessed input data.
- Convolutional layers, which apply convolution processes to input data, are the fundamental components of CNNs. In order to do these processes,

a tiny filter, is slid across the input image, sometimes called a kernel, and the dot products between the filter and certain regions of the image are calculated. The network can now record spatial hierarchies of features, from lower-level elements like textures and edges to higher-level elements like forms and objects, thanks to this process.

- Convolutional layers in a CNN apply filters, or kernels, to the input image to detect patterns and features such as shapes, edges, and textures.

B. *Neuro-Fuzzy Systems*

- Neuro-fuzzy systems are hybrid models that can handle both symbolic knowledge representation (fuzzy logic) and numerical data processing (neural networks). They achieve this by fusing neural network and fuzzy logic components. In your project, a neuro-fuzzy system might leverage fuzzy logic reasoning in conjunction with CNN feature extraction power to enhance the interpretability and robustness of the model.
- Fuzzy logic is utilized in a neuro-fuzzy system to simulate linguistic variables, membership functions, and fuzzy rules in order to simulate human-like reasoning. Neural networks, on the other hand, have the computational capacity to learn from data and adaptively modify the fuzzy logic components according to the input-output relationships seen in the training set. This integration allows for the combination of fuzzy logic's interpretability and neural networks' learning capabilities to create neuro-fuzzy systems.

C. *Supervised Learning*

- The model will be trained using supervised learning methods since we most likely have tagged data pictures of plants labelled with the diseases that correlate to them. Neural network training procedures like backpropagation, which adjusts the model's parameters to lessen the difference between expected and actual outcomes, could be used for this.
- During training, the supervised learning algorithm adjusts its internal parameters using optimisation techniques in an effort to reduce the discrepancy between the expected outputs and the actual labels in the training data. In this process, the difference between the true and predicted values is quantified by iteratively updating the model's parameters using a chosen loss function. Gradient descent and its variations are popular optimisation methods used in supervised learning applications.
- A variety of tasks, such as regression, structured prediction, and classification, are included in supervised learning. Whereas the model predicts continuous numerical values in regression tasks, it predicts discrete class labels for input examples in classification tasks.

IV. DATE SET AND IMAGE PRE-PROCESSING

Data Set:

From this detection of plant disease, The dataset that we use for the detection of plant disease was taken from the Kaggle repository which contain 20,639 images out of which 997 images belong to diseased Pepper bell's leaves, 2,000 images belong to diseased Potato's leaves, 14,421 images belong to diseased Tomato's leaves, 1,478 images belong to healthy Pepper bell's leaves. 152 images belong to healthy Potato's leaves, and 1,591 images belong to healthy Tomato's leaves.

Fig 2 and 3 shows both healthy and diseased pepper bell plant leaves.

Fig 2: Pepper bell plant Diseased leaves

Fig 3: Pepper bell plant Healthy leaves

Fig 4 and 5 shows both healthy and diseased potato plant leaves.

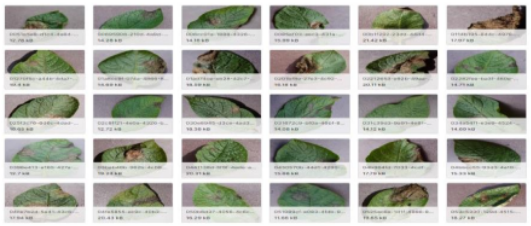

Fig 4: Potato plant Diseased leaves

Fig 5: Potato plant Healthy leaves

Fig 6 and 7 shows both healthy and diseased tomato plant leaves.

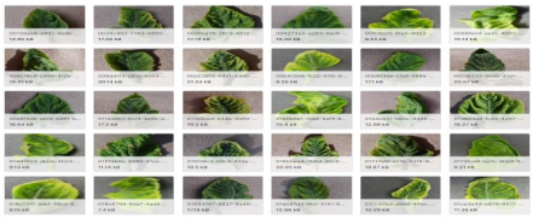

Fig 6: Tomato plant Diseased leaves

Fig 7: Tomato plant Healthy leaves

Image Pre-processing:

Image pre-processing stands as a vital initial phase in both computer vision and image processing applications, where a myriad of techniques are deployed to refine raw images, thereby bolstering their quality and priming them for subsequent analysis or manipulation. Concurrently, techniques such as image normalization aid in standardizing pixel values across images, facilitating fair comparisons and optimizing model performance. Moreover, noise reduction techniques strive to mitigate unwanted distortions or aberrations introduced during image acquisition or transmission, thereby preserving crucial image features. Alongside noise reduction, image enhancement techniques are leveraged to amplify desired characteristics or details within images, enhancing their interpretability or aesthetic appeal. Integral to image pre-processing is the task of edge detection, which involves identifying boundaries **or** transitions between objects within images, thus facilitating subsequent segmentation or feature extraction tasks. In essence, image pre-processing orchestrates a symphony of operations, harmonizing raw inputs into refined data ready for the symposium of computer vision algorithms. Fig. 8 illustrates the image pre-processing procedure.

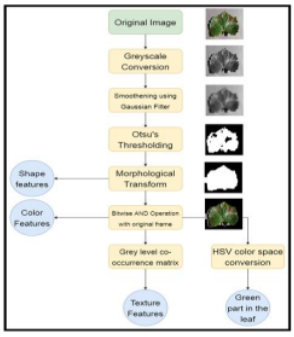

Fig 8: Image Pre-processing

V. Results & Performance Evaluation

A deep neuro fuzzy network's prediction accuracy depends on the batch size and training rate. To find the ideal combination of these attributes, we experimented with various starting learning rates and batch sizes while monitoring changes in the learning rate. Figure 9 displays a line graph titled "Training and Validation loss." On the graph, the blue line signifies the training loss and the red line the validation loss. The label on the x-axis is labelled, but it is not visible; it probably indicates how many epochs or iterations were used to train the machine learning model. The loss value, which shows the model's mistake during training, is represented by the y-axis. The model is learning and getting better over time at using the training data, as shown by the blue line (training loss), which starts at a larger value and gradually trends downward. In addition to showing a decreasing trend, the red line (validation loss) also has notable spikes at specific epochs that may indicate overfitting at those times when the model performs well on training data but badly on unknown validation data.

The graph's x-axis, which has numbers ranging from 0 to 25, indicates that it spans 25 epochs or iterations. The loss values are represented along the y-axis, which has a range of 0 to 1.75.

Fig 9: Training and Validation loss

The fig 10 shows a line graph labelled "Training and Validation accuracy," on which two lines are plotted to indicate how accurate a machine learning model is during training. The x-axis represents the epoch number, which runs from 0 to 25, and the y-axis shows the accuracy, which varies from 0.88 to 0.98.

There are two lines on the graph:

A. The blue line represents the training accuracy. It starts off just below 0.98 and remains relatively stable throughout the training process, with only minor fluctuations.

B. The red line represents the validation accuracy. This line starts at approximately 0.91, shows more variability than the training accuracy, dips slightly below 0.90, and then generally trends upward, reaching around 0.96 by the 25th epoch.

The model's performance is assessed using the graph; ideally, the training and validation accuracy should be high and near to one another. A point of examination for more model tuning may be the fluctuations and the difference between the two lines, which could hint to problems like overfitting or underfitting.

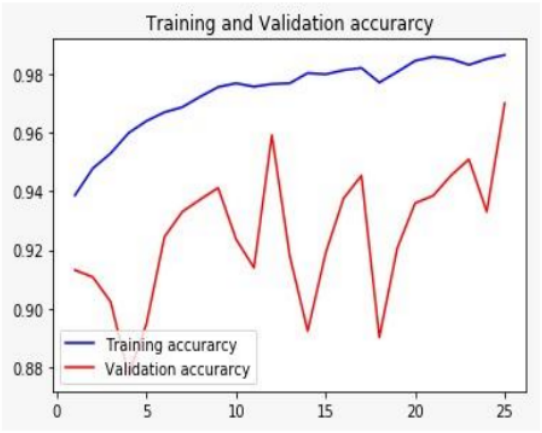

Fig 10: Training and Validation accuracy

TABLE I.

WE USED PERFORMANCE INDICATORS TO COMPARE OUR SUGGESTED NETWORK TO OTHER MODELS.

Model	Precision	Accuracy	F1 Score
Long Short-Term Memory	0.74	0.71	0.77
Support Vector Manchine	0.55	0.57	0.50
Convolutional Neural Network	0.83	0.83	0.83
GoogleNet	0.93	0.94	0.94
DenseNet	0.87	0.85	0.86
AlexNet	0.89	0.90	0.91
Proposed model	0.96	0.97	0.96

A comparison graph with more baseline models is shown in Fig. 11. Among the models that we have proposed, ours has the highest accuracy rate.th

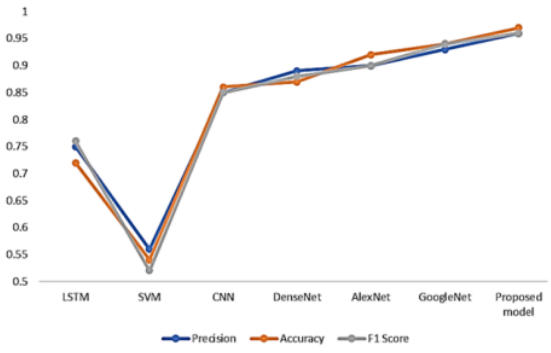

Fig 11: A comparison table for the suggested model

VI. CONCLUSION

This research presents NFNHDL-based Deep Learning, a productive technique for classifying and diagnosing diseases in agricultural plant leaves. The method developed by the authors uses ROI extraction in the pre-processing unit to eliminate unwanted noise and distortion from input images. After removing any damaged portions from the output after preprocessing, the approach extracts features (textual, statistical, and CNN features) and enhances the data. The state of the image is subsequently evaluated by use of a Deep Fuzzy neural network. The NFNHDL classifier determines which particular leaf diseases, such as brown spot, blast, or BLB, to recognise in the last stage depending on variables including For every category, the F1-score, accuracy, and recall.

In conclusion, it turns out that using a deep neuro-fuzzy network to diagnose diseases in agricultural plants is a great, dependable, and highly accurate method. Additionally, with scores of 0.96, 0.97, and 0.96 in evaluating accuracy, sensitivity, and specificity, respectively, the NFNHDL-based Deep Learning technique performed better. Subsequent endeavours will centre around refining tactics to augment classification outcomes and assessing the suggested methodology with more extensive datasets, taking into account additional diseases.

REFERENCES

[1] S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," in 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 768-771, doi: 10.1109/ICCUBEA.2015.153.

[2] S. C. Madiwalar and M. V. Wyawahare, "Plant disease identification: A comparative study," in 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 2017, pp. 13-18, doi: 10.1109/ICDMAI.2017.8073478.

[3] Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. Computers and Electronics in Agriculture, 137, 52-58.

[4] Yin H, Gu YH, Park C, Park J, Yoo SJ (2020) Transfer learning-based search model for hot pepper diseases and pests. Agriculture 10:439

[5] Ashwinkumar, S., Rajagopal, S., Manimaran, V., & Jegajothi, B. (2022). Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. Materials Today: Proceedings, 51, 480-487.

[6] P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka, and E. Hernandez, "Plant Disease Detection Using Hyperspectral Imaging," in 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp. 1-8, doi: 10.1109/DICTA.2017.8227476.

[7] C. Lu, S. Gao, Z. Zhou, Maize disease recognition via fuzzy least square support vector machine. J. Inf. Comput. Sci. 8(4), 316–320 (2013)

[8] V. Barra, J.Y. Boire, Automatic segmentation of subcortical brain structures in MR images using information fusion. IEEE Trans. Med. Imaging 20(7), 549–558 (2001)

[9] V. Barra, J.Y. Boire, A general framework for the fusion of anatomical and functional medical images. Neuroimage 13(3), 410–424 (2001)

[10] M.J. Hsu, Y.H. Chien, W.Y. Wang et al., A convolutional fuzzy neural network architecture for object classification with small training database. Int. J. Fuzzy Syst. 22(1), 1–10 (2020).

[11] G.J. Klir, "Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?" Fuzzy Sets Syst. 24(2), 141–160 (1987).

DETECTING PLANT DISEASES WITH DEEP CONVOLUTION NEURO-FUZZY NETWORKS USING DEEP LEARNING.pdf

ORIGINALITY REPORT

9%	2%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Ashok Kumar Koshariya, Mohd. Shaikhul Ashraf, Rashmi Nigam, T Sampath Kumar, Deepak Kumar. Awasthi, G Elavarasan. "Implementation of Deep Convolution Neuro-Fuzzy Network to Plant Disease Detection, Risk Assessment, and Classification", 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2023 Publication	3%
2	D. Thabitha*, Dr.B. Jhansi Vazram. "Recognition of Position Group Relationship in Dynamic Social Networks", International Journal of Innovative Technology and Exploring Engineering, 2019 Publication	1%
3	Submitted to University of Hertfordshire Student Paper	1%
4	"Computational Vision and Bio-Inspired Computing", Springer Science and Business	1%

Media LLC, 2021
Publication

5	Mir Shakeel Ahmad*, Harmandeep Kaur, Tanika Thakur. "Dual Polarization-Based Transmitter Configuration Improving Q-Factor for a Single-Mode Fiber Link", International Journal of Recent Technology and Engineering (IJRTE), 2020 Publication	1 %
6	Submitted to University of Alabama at Birmingham Student Paper	1 %
7	Submitted to University of Wollongong Student Paper	1 %
8	link.springer.com Internet Source	1 %
9	Submitted to University of Asia Pacific Student Paper	<1 %
10	Sheshang Degadwala, Dhairya Vyas, Sonia Panesar, Divya Ebenezer, Darshanaben D. Pandya, Vandita Dipak Shah. "Revolutionizing Hops Plant Disease Classification: Harnessing the Power of Transfer Learning", 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), 2023 Publication	<1 %

11	www.irjmets.com Internet Source	<1%
12	mail.ijabe.org Internet Source	<1%
13	Studies in Fuzziness and Soft Computing, 2015. Publication	<1%
14	Xiaole Tian, Xiangyan Meng, Qiufeng Wu, Yiping Chen, Jinchao Pan. "Identification of Tomato Leaf Diseases based on a Deep Neuro-fuzzy Network", Journal of The Institution of Engineers (India): Series A, 2022 Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		