Multiclass Classification and Severity Prediction of Rice Leaf Diseases Using Fuzzy Logic and Neural Networks

A
Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

A. Aditya Vardhan Reddy 1602-21-737-131

M. Sanath 1602-21-737-171

Under the guidance of

B. Leelavathy

Assistant Professor

Department of Information Technology

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2025

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We, **A. Aditya Vardhan Reddy** and **M. Sanath** bearing hall ticket numbers, **1602-21-737-131** and **1602-21-737-171** hereby declare that the project report entitled **Multiclass Classification and Severity Prediction of Rice Leaf Diseases Using Fuzzy Logic and Neural Networks** under the guidance of **B. Leelavathy, Assistant Professor**, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Aditya Vardhan Reddy
1602-21-737-131

M. Sanath
1602-21-737-171

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

I, **M. Sanath** bearing hall ticket numbers, **1602-21-737-131**, hereby declare that the project report entitled **Multiclass Classification and Severity Prediction of Rice Leaf Diseases Using Fuzzy Logic and Neural Networks** under the guidance of **B. Leelavathy, Assistant Professor**, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

M. Sanath
1602-21-737-171

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled **Multiclass Classification and Severity Prediction of Rice Leaf Diseases Using Fuzzy Logic and Neural Networks** being submitted by **A. Aditya Vardhan Reddy** and **M. Sanath** bearing **1602-21-737-131** and **1602-21-737-171** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

B. Leelavathy		**Dr. K. Ram Mohan Rao**
Assistant Professor		**Professor & HOD, IT**
Internal Guide	**External Examiner**	

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled **Multiclass Classification and Severity Prediction of Rice Leaf Diseases Using Fuzzy Logic and Neural Networks** being submitted by **M. Sanath** bearing **1602-21-737-171** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

B.Leelavathy	**Dr. K. Ram Mohan Rao**
Assistant Professor	**Professor & HOD, IT**
Internal Guide	

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the Main project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

It is with immense pleasure that we would like to take the opportunity to express our humble gratitude to **B. Leelavathy, Assistant Professor, Information Technology** under whom we executed this project. We are also grateful to **Dr. S.K. Chaya Devi, Associate Professor, Information Technology** for her guidance. Their constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

We are very much thankful to **Dr. K. Ram Mohan Rao, Professor & HOD, Information Technology,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. S.V.Ramana, Principal** of **Vasavi College of Engineering and Management** for providing facilities. Not to forget, we thank all other faculty and non-teaching staff, who had directly or indirectly helped and supported me in completing my project in time.

Abstract

Rice as Staple Food for almost half the world's population but It is highly prone to diseases such as bacterial blight, blast, tungro, and brown spot, which have a deep influence on yield and quality. Early detection and correct classification of rice leaf diseases are crucial for reducing crop loss and ensuring agricultural productivity. This project proposes a hybrid deep learning and fuzzy logic-based system for multiclass classification and severity estimation of rice leaf diseases. The system is deployed in two phases: a Convolutional Neural Network (CNN) is first trained to predict disease types from leaf images, and then a fuzzy logic-based neural inference model for determining disease severity. The CNN learns rich features from RGB images and provides class probabilities, which are fed into a fuzzy inference system constructed using custom fuzzy layers integrated into the neural architecture. The fuzzy logic increases severity assessment interpretability and flexibility by emulating human-like reasoning. Experimental results show high classification accuracy for diseases like Brown Spot, Leaf Blast, and Bacterial Leaf Blight, as well as solid severity grading across mild, moderate, and severe phases. The hybrid methodology guarantees not only accurate classification but also subtle severity estimation, helping with decision-support systems in precision agriculture.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN – Convolutional Neural Network

FIS - Fuzzy Inference System

FCNN - Fuzzy Logic Convolutional Neural Network

1. INTRODUCTION

Rice is a fundamental food for over half of global population, which supports food security, livelihoods, and economies in numerous developing and developed countries. Rice has a plethora of biotic challenges, of which leaf diseases form a significant one to yield as well as quality. Economically important diseases of rice such as blast, bacterial blight, brown spot, tungro, and other diseases can sharply curtail production if not discovered and controlled at the earliest opportunity. Conventional disease diagnosis by expert agronomists involves physical observation in the field, a time-consuming process that is arduous and subject to human error. Besides, faint preliminary symptoms are also likely to escape notice, thereby enabling infections to gain strength before corrective action can be taken.

Machine-based, image-driven disease diagnosis systems provide an encouraging alternative. With the progress of computer vision and machine learning, it becomes possible to build tools that can identify disease categories at high speeds and with great accuracy from digital leaf images of rice. In the past decade, Convolutional Neural Networks (CNNs)—a category of deep networks motivated by the visual cortex—have obtained state-of-the-art performance in numerous image classification problems, including plant disease detection. CNNs learn hierarchical feature representations from raw pixel values directly, without the need for explicit feature engineering and often surpassing traditional classifiers like Support Vector Machines or Random Forests in both accuracy and resilience.

Even with their strength, pure CNN-based methods have two significant limitations in the agricultural context. First, although CNNs are very good at classifying discrete categories (e.g., "blast" vs. "brown spot"), they are poor at measuring the severity of an infection—an inherently continuous notion that does not lend itself to being mapped onto hard class boundaries. Severity estimation is important for agronomic decision-making: light infection can be treated by low-dose, while heavy infestation could necessitate more drastic measures or quarantine of the crop. Second, the "black box" character of deep networks makes them black boxes to end-users like farmers and extension agents who must know the rationale behind a system's recommendation before they can trust and act upon it.

This study overcomes these limitations by introducing a hybrid approach that combines CNN-based disease classification with a Fuzzy Inference System (FIS) for severity estimation. The ensuing pipeline—dubbed a Fuzzy-CNN hybrid—capitalizes on the representation learning strength of CNNs and adds the interpretability and gradual reasoning of fuzzy logic. Essentially, the CNN determines what disease is present, and the fuzzy logic layer determines how severe that disease manifestation is, with outputs like "Slight," "Severe," or "Profound."

1.1.Problem Statement – Overview

Agriculture forms the pillars of most economies, particularly in nations such as India where a large percentage of the population is wholly reliant on it. Rice being one of the staple food crops is extremely vulnerable to several leaf diseases which are liable to result in high yield losses. Conventional identification and classification of rice leaf diseases are usually time-consuming, prone to errors, and reliant on specialist analysis and are thus not practical for deployment at large scales.

This project overcomes the issues related to machine-assisted disease classification and prediction of severity by employing a hybrid solution that leverages the strengths of deep learning and fuzzy logic. The multiclass classification of rice leaf diseases is performed by a Convolutional Neural Network (CNN), and it is followed by a fuzzy logic neural network that assesses the severity of the suspected disease. This two-stage model improves accuracy, interpretability, and scalability of agricultural diagnosis.

1.2.Motivation

The rising need for food security and maximization of crop production requires smart, real-time plant disease diagnosis systems. Traditional image processing methods and rule-based systems hardly generalize across diverse field conditions and lighting settings. Although CNNs are very effective in classifying intricate visual patterns, they are not interpretable and fail with edge cases such as borderline levels of severity. To overcome these limitations, this research utilizes the strong feature extraction capability of CNNs and the decision-making flexibility of fuzzy logic. Fuzzy systems mimic expert reasoning by dealing with uncertainty and partial truth, which is best suited for evaluating disease severity based on visual symptoms.

Through the integration of both methods, our vision is to create a cost-effective, interpretable, and precise rice leaf disease prediction system that can aid farmers and agricultural practitioners in making timely interventions.

1.3.Scope & Objectives of the Proposed Work

Scope of the Proposed Work

This project is aimed at automating the classification and severity analysis of rice leaf diseases through a deep learning and fuzzy logic pipeline. It encompasses:

- Classification of key rice leaf diseases using CNN.
- Feature map extraction from CNN outputs for fuzzy inference.

- Fuzzy neural inference system design and training to predict severity levels.
- Assessment of the hybrid model's effectiveness using real-world leaf images.
- Offering visual and analytical performance verification via accuracy measures and ROC curves.

Objectives of the Proposed Work

1. Build a CNN-based Model for multiclass rice leaf disease classification from RGB images.
2. Extract Feature Maps from the convolutional layers of the CNN to be used as inputs for fuzzy reasoning.
3. Implement a Fuzzy Logic Neural Inference System (FCNN) to evaluate severity levels (e.g., mild, moderate, severe).
4. Combine CNN and Fuzzy Models into a single pipeline for end-to-end disease and severity prediction.
5. Assess the Hybrid Model on a handpicked dataset with accuracy, confusion matrix, and classification metrics.

1.4. Organization of the Report

The report is organized systematically to provide a comprehensive analysis of the application of deep learning and fuzzy logic for identifying and predicting the severity of rice leaf diseases.

1. Introduction

The **Introduction** presents an overview of the agricultural challenge of maintaining rice crop health and introduces deep learning and fuzzy logic as transformative technologies in the domain of precision agriculture. It emphasizes the significance of integrating Convolutional Neural Networks (CNNs) and Fuzzy Inference Systems (FIS) to facilitate automated plant disease detection and severity analysis.

2. Literature Survey

The **Literature Review** section delves into previous studies related to plant disease classification using machine learning and CNNs, while highlighting the role of fuzzy logic in making decisions under uncertainty, especially in the context of disease severity prediction. It outlines the limitations of existing models and justifies the need for an integrated CNN-FIS framework.

3. Proposed Syatem

In the **Proposed System** section, a novel approach combining CNN-based disease classification with a fuzzy logic-based neural network (FCNN) for severity prediction is introduced. The architecture of the CNN is detailed, along with the use of fuzzy inference blocks to evaluate disease intensity. This section also describes essential functional modules such as image preprocessing, feature extraction, and fuzzy rule-based reasoning. Additionally, it includes the pseudocode of the entire pipeline to ensure clarity and reproducibility.

4. Experimental Setup and Results

The **Experimental Setup and Results** section provides an account of the hardware and software environments used during model training and testing. It discusses the rice leaf image dataset, including the preprocessing and augmentation techniques applied. The architecture of the CNN, including the number of layers, activation functions, and dropout techniques, as well as the fuzzy model with its rule set and Gaussian membership parameters, are presented in detail. The model's performance is evaluated using various classification metrics such as Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC-AUC curves. The section also includes graphical illustrations of both disease classification and severity prediction outcomes.

5. Conclusion and Future Work

The **Conclusion and Future Work** section summarizes the achievements of the proposed system, emphasizing its performance and potential in real-world agricultural applications. It proposes future enhancements such as the deployment of mobile-based diagnostic tools and the integration of real-time drone image acquisition. The section underscores the importance of combining artificial intelligence models with fuzzy logic to achieve interpretable and intelligent plant health monitoring solutions.

6. References

The **References** section catalogs all scholarly articles, research papers, online resources, and datasets consulted throughout the development of the project.

7. Appendix

Finally, the **Appendix** includes annotated code snippets from the implementation notebooks, model architecture visualizations, sample inputs and outputs, and a GitHub link to ensure full reproducibility of the project.

This structured approach ensures a clear and logical flow of information, helping readers understand the study's significance, methodology, findings, and future directions in Classifying and predicting the Severity of the Rice plant.

2. LITERATURE SURVEY

Lamba et al. [1] proposed a hybrid CNN-SVM model for predicting the severity of blast disease in paddy crops. Using a dataset of 1908 images sourced from Mendeley, GitHub, and field data, the study categorizes severity into four levels based on leaf area affected. The CNN extracts features, while SVM classifies severity with 97% accuracy. The model aims to reduce crop loss and guide farmers on effective treatment strategies.

Saminathan et al. [2] proposed model to classify four diseases. This study classifies four paddy diseases—bacterial blight, blast, tungro, and brown spot—using 5932 images from the Sethy dataset. The preprocessing includes color-space conversion and segmentation. Features like Hu moments and GLCM are extracted, and classification employs Random Forest, yielding 97.62% accuracy after testing.

Jindal et al. [3] utilize Federated Learning with CNN to classify groundnut diseases like Rust, Early Leaf Spot, and Late Leaf Spot into five severity levels. Using a dataset of 1950 images, the study achieves high accuracy (up to 98%) while maintaining data privacy. Federated averaging models from decentralized clients for reliable classification.

Dhar et al. [4] (2022) put forth a hybrid system that leverages Gist (global) and Local Binary Pattern (LBP) features to classify plant leaf diseases. The approach accurately described scene-level and localized texture patterns, making possible robust detection in ten species of plants. By employing machine learning models such as SVM and AdaBoost, the system attained high accuracy for data like rice and tomato, demonstrating its efficiency in early disease diagnosis and crop health management

Khalid and Karan et al. [5] showed the capability of Deep Learning models such as CNN and MobileNet in detecting plant diseases. They used GradCAM for model interpretability, with a maximum accuracy of 96% in disease detection over 38 classes of plants. The research highlighted DL's scalability and speed over conventional methods, with great promise for global agricultural sustainability.

Singh et al. [6] introduce a system which integrates convolutional neural networks (CNNs) with conventional image processing techniques for plant disease classification. The strategy leverages deep features, texture descriptors such as GLCM and HoG, and color features to develop hybrid features. Feature selection is optimized through Binary Particle Swarm Optimization (BPSO), and classification is performed using Bayesian Optimized SVM and Random Forest classifiers. The hybrid system greatly enhances detection accuracy, with high precision in feature extraction by CNNs and complementary feature insights through traditional methods.

Guo et al. [7](2023) utilized Convolutional Neural Networks (CNNs) for detecting plant diseases based on a heterogeneous dataset of crop images. The research emphasized preprocessing strategies and multi-layer CNN architecture to gain a classification accuracy of 92.23%. The study emphasizes CNNs' ability to process hierarchical image features, promoting early disease diagnosis and promoting food security.

Khalid et al. [8] examined the application of pre-trained CNN architectures such as VGGNet and AlexNet for crop disease classification. They obtained 99% accuracy for crops such as maize and grapes using the PlantVillage dataset. Their research emphasizes the effectiveness of transfer learning and data augmentation in solving agricultural problems.

Hosny et al. [9] (2023) introduced a new deep CNN architecture combined with handcrafted LBP features to improve texture detection from leaf images. Tested on apple, tomato, and grape datasets, the system was as high as 99% accurate, providing a lightweight, efficient solution to multi-class disease classification. The method provides very high precision in varied agricultural environments.

Moupojou et al. [10] respond to challenges of plant disease detection from field images with intricate backgrounds. Their method combines the Segment Anything Model (SAM) for extracting leaves from noise in the background and employs Fully Convolutional Data Description (FCDD) to detect diseased areas. Through using pre-trained CNNs and Transfer Learning, the model improves accuracy in multi-disease detection across datasets such as PlantVillage and PlantDoc, with enhanced performance under natural field conditions.

Azim et al. [11] employed the UCI ML Repository dataset of 120 RGB images in three classes: brown spot, bacterial leaf blight, and leaf smut. The authors utilized image pre-processing by saturation and hue thresholds for background elimination and segmentation. Color, shape, and texture features were used to represent local and global image statistics.Classification was performed using an extreme gradient boosting decision tree ensemble, achieving 86.58% accuracy.

Pengtao et al. [12] evaluated various public datasets, such as Kaggle and Mendeley, covering multiple classes of rice diseases (e.g., bacterial leaf blight, brown spot, tungro). Deep learning approaches, including custom CNNs and transfer learning (e.g., ResNet, VGG-16), were emphasized for feature extraction. Features included lesion shape, color, and texture. The review discussed segmentation techniques like Otsu's thresholding and semantic segmentation, highlighting a 99.7% accuracy on some datasets.

Simhadri et al. [13]-Rice Disease Identification and Classification by Integrating Support Vector Machine With Deep Convolutional Neural Network: The dataset used contained 1,080 images of nine rice diseases, including leaf blast and sheath blight. A modified Inception V3 CNN model with transfer learning was used for feature extraction, followed by SVM classification. Features were automatically learned by the CNN and further refined using the SVM, achieving an accuracy of 97.5% on a test set

Zhou et al. [14] proposed an integrated approach combining FCM-KM clustering and Faster R-CNN for rapid rice disease detection. They addressed challenges such as noise, blurred edges, and background interference by employing advanced preprocessing techniques, including a weighted median filter and a two-dimensional Otsu threshold segmentation. A dataset of 3010 rice disease images representing three diseases (blast, bacterial blight, and sheath blight) was used for training and evaluation. Their method optimized Faster R-CNN's bounding box selection with FCM-KM clustering, achieving up to 98.26% accuracy while significantly reducing detection time, making it effective for real-world applications

Liang et al. [15] developed a convolutional neural network (CNN)-based system for recognizing rice blast disease. Using a custom dataset of 5808 image patches labeled by experts, the method extracted high-level features, outperforming traditional feature extraction methods like LBPH and Haar-WT. Employing a stride-based segmentation approach, the model achieved an accuracy of over 95% with robust ROC curves, demonstrating its potential in practical disease classification tasks

Aggarwal et al. [17] presented an overview of AI and machine learning techniques for rice disease detection, emphasizing smart farming. Their method combines preprocessing, feature extraction, and classification phases to identify diseases such as leaf blast and bacterial blight. They emphasized the role of effective clustering algorithms such as k-means and deep learning architectures such as CNNs, attaining high classification accuracies in identifying rice crop diseases.

Fasanmade et al. [18] introduce a fuzzy logic-based dynamic Bayesian model for classifying driver distraction severity. Using fuzzy rules combined with discrete Bayesian inference, their method effectively models uncertainties of driver behavior. This Mamdani fuzzy system, being able to shift driver distraction levels from safe to hazardous, greatly improves the accuracy of Advanced Driver Assistance Systems (ADAS). Fuzzy logic incorporation enhances classifying accuracy by handling uncertainties and variability of human action, presenting an imperative step forward for real time driver monitoring.

Riaz et al. [19] suggest a semi-supervised CNN model combined with fuzzy rough C Mean (FRCM) clustering for image classification. Fuzzy logic minimizes vagueness and indiscernibility in unlabeled data, while CNNs denoise and improve feature representation. The hybrid method attains state-of-the-art accuracy on benchmark datasets by fusing supervised and unsupervised learning, showcasing the effectiveness of fuzzy logic in handling noisy, high-dimensional data.

Bhatti et al. [20] present the ETFG model, a combination of fuzzy C-Means clustering and deep learning for hyperspectral image classification. Fuzzy logic improves dimensionality reduction by retaining important spectral and spatial features, reducing data redundancy. Used in conjunction with 3D-CNNs and graph attention networks, this method attains superior classification accuracy, overcoming challenges such as high dimensionality and data uncertainty in precision agriculture.

Kim et al. [21] use an ANFIS-augmented CNN for the identification of plant diseases, involving fuzzy logic for the accurate classification of leaf health. Utilizing the strengths of adaptive neuro-fuzzy inference systems, the model has improved interpretability and performance with over 99% accuracy when combined with local binary patterns (LBP). The incorporation of fuzzy rules improves decision-making, rendering it an aggressive tool for agricultural uses.

Sharmila et al. [22] suggest FTLM, which integrates fuzzy TOPSIS with language model for plagiarism severity evaluation. Fuzzy logic deals with imprecision in semantic and structural differences, and language models improve contextual examination.This synergy greatly enhances plagiarism detection accuracy and robustness, especially in difficult cases involving paraphrasing

Pajany et al. [23] created the Optimal Fuzzy Deep Neural Networks-based Plant Disease Detection and Classification (OFDNN-PDDC) method using UAV based remote sensing data. The research incorporates a fuzzy restricted Boltzmann machine (FRBM) due to its better ability to handle uncertainty in agricultural situations, greatly increasing detection accuracy. With fuzzy logic being coupled with neural network concepts, the model obtains 96.18% and 98.85% accuracy on APD and CPD datasets, respectively. This method, supplemented with ShuffleNetv2 for feature extraction and hyperparameter optimization through the tent chaotic salp swarm algorithm, emphasizes fuzzy logic's flexibility in working with intricate, fuzzy information, thereby improving overall model precision

Lin et al. [24] presented a Multiscale Convolutional Fuzzy Neural Network (MCFNN) for evaluating the quality of metal additive manufacturing components from ultrasound images. The MCFNN integrates fuzzy inference systems with CNNs to efficiently classify porosity levels. Fuzzy logic improves the interpretability of the system and handles noise and uncertainty, enhancing robustness. The multiscale fusion mechanism and Gaussian membership functions of the model obtain 91.44% accuracy, surpassing state-of-the-art benchmarks with fewer parameters. This indicates the key role of fuzzy logic in accurate quality assessment for manufacturing processes.

Paper Title	Model Used	Dataset Used	Accuracy	Notable Points
A Novel Hybrid Severity Prediction Model for Blast Paddy Disease Using Machine Learning [1]	CNN-SVM Hybrid Model	1908 images from Mendeley, GitHub, and field data	97%	Focuses on severity prediction for blast disease; uses CNN for feature extraction and SVM for classification.
Multiclass Classification of Paddy Leaf Diseases Using Random Forest Classifier [2]	Random Forest	Sethy dataset (5932 images)	97.62%	Classifies four diseases: bacterial blight, blast, tungro, and brown spot; employs Hu moments and GLCM features.
Severity-Level Assessment of Groundnut Leaf Diseases: A Federated Learning and CNN Approach [3]	Federated Learning with CNN	1950 images of groundnut leaves	98%	Utilizes federated learning to preserve privacy while achieving robust classification of five severity levels.
ANFIS Fuzzy Convolutional Neural Network Model for Leaf Disease Detection [21]	ANFIS CNN + Local Binary Pattern (LBP) Features	PlantVillage Dataset	99%+	Uses adaptive neuro-fuzzy inference systems with LBP features for precise detection of plant diseases. High interpretability and robustness in agricultural contexts
Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion [14]	FCM-KM + Faster R-CNN	3010 images (blast, bacterial blight, sheath blight)	96.71%–98.26%	Combines clustering and Faster R-CNN for bounding box optimization; significantly reduces detection time.
Rice Blast Disease Recognition Using a Deep Convolutional Neural Network [15]	CNN (Custom Architecture)	5808 image patches (2906 positive, 2902 negative)	>95%	Utilizes high-level feature extraction; outperforms traditional methods like LBPH and Haar-WT.
Advanced Plant Disease Segmentation in Precision Agriculture Using Optimal Dimensionality Reduction With Fuzzy C-Means Clustering and Deep Learning [20]	ETFG (Enhanced Transformation-Enabled Fuzzy Graph) with 3D-CNN, GAT, PCA, FCM	Five datasets: Indian Pines (220 bands), Pavia (103/102 bands), KSC (224 bands), Sugar Beets, Carrot Weed Dataset	~75-80% (varies by dataset)	Combines fuzzy clustering, PCA, 3D-CNN, and GAT for hyperspectral imaging and segmentation in precision agriculture. Demonstrates improved classification and scalability.

Table 2.1. Comparative Study of the existing methods with the proposed Deep F-CNN Model

3. PROPOSED SYSTEM

3.1. Methodology

The methodology of this project presents a hybrid approach combining Convolutional Neural Networks (CNNs) and Fuzzy Logic to address the dual objectives of rice leaf disease classification and severity estimation. Initially, the Paddy Doctor Visual Image Dataset, comprising 26,000 images equally distributed among 13 classes (including 12 disease types and one healthy class), is employed. The images undergo thorough preprocessing steps such as resizing to 200x200 pixels, normalization, and extensive data augmentation involving random rotation, flipping, and zooming to improve model generalization and robustness. A custom CNN architecture is then designed with convolutional, batch normalization, pooling, and fully connected layers to extract high-level features from the input images. The CNN is trained using the Adam optimizer and categorical cross-entropy loss to achieve optimal classification accuracy. Once trained, intermediate features from the CNN are passed to a Fuzzy Inference System (FIS) which leverages Gaussian membership functions to interpret the features in terms of linguistic variables like low, medium, and high. Expert-defined fuzzy rules evaluate these features to infer the severity level of the disease in three categories: Slight, Severe, and Profound. The defuzzification process then converts fuzzy outputs into precise severity labels. This integration of CNNs and FIS not only enhances classification accuracy but also provides interpretability and nuanced reasoning about disease severity, which traditional deep learning systems often lack. The model was validated with an 80:20 train-validation split, maintaining class balance throughout, and achieved a classification accuracy of 97.7% and severity prediction accuracy of 92.1%. This hybrid methodology demonstrates its suitability for real-world agricultural applications, offering farmers and agronomists a reliable tool for early and interpretable diagnosis, enabling better disease control and improved crop management.

3.1.1. Data Collection and Preprocessing

The data used in this research is the Paddy Doctor Visual Image Dataset. We used the paddy-doctor-diseases-small-augmented-with a total of 26,000 images divided into an even number of 13 classes (2,000 images in each class). This dataset version is balanced and augmented with samples, giving us a strong benchmark for multiclass classification tasks.

Disease Name	Original Image Count	Augmented Image Count
Bacterial Leaf Blight (BLB)	648	2000
Bacterial Leaf Streak (BLS)	505	2000
Bacterial Panicle Blight (BPB)	450	2000
Black Stem Borer (BSB)	506	2000
Blast	2,351	2000
Brown Spot	1,257	2000
Downy Mildew	868	2000
Hispa	2,151	2000
Leaf Roller	1,095	2000
Tungro	1,951	2000
White Stem Borer	1,273	2000
Yellow Stem Borer	765	2000
Normal (Healthy)	2,405	2000

Table 3.1 Description of the dataset with original and augmented images

Data Pre-Processing Steps:

To ensure uniformity and enhance model generalization, Prior to inputting images into the CNN–Fuzzy hybrid pipeline, we perform a sequence of preprocessing steps aimed at normalizing he inputs, augmenting the training distribution, and speeding up model convergence. Each of these processes serves a particular purpose in enhancing both the accuracy and resilience of our disease classification and severity prediction system.

1. Image Resizing

- Objective: Ensure that every image conforms to the same dimensions so that the network's input layer always receives a fixed-size tensor.
- Procedure:
  - All raw RGB images are programmatically resized to 200×200 pixels using bi-linear interpolation.

- This size was chosen as a trade-off between retaining sufficient detail (to capture lesion shapes and color variation) and keeping the model computationally tractable.

- Impact:
  - Uniform input dimensions eliminate the need for dynamic shape handling within the network.
  - By keeping the spatial resolution constant, we make sure that convolutional filters learn similar feature representations for the whole dataset.

2. Pixel Normalization

- Goal: Normalize pixel intensity values to a shared numeric range to enhance numerical stability and accelerate gradient-based learning.
- Procedure:
  - Every pixel value, initially ranging from 0 to 255, is divided by 255.0, scaling the intensities to the continuous range [0, 1].
- Effect
  - Prevents high activation values that may saturate the ReLU units or cause exploding gradients.
    - Enables smoother and quicker convergence during training by maintaining the input distribution centered and normalized.

3. Data Augmentation

- Goal: Artificially increase the diversity of the training data to minimize overfitting and enhance the model's capacity to generalize to unseen field conditions.
- Techniques and Parameters:
  1) Rotation
     - Random rotations sampled uniformly in the range [–40°, +40°].
     - Assists the network in becoming invariant to leaf orientation, in natural environments of which can change enormously with respect to how leaves are captured.

2) Horizontal Flip

- There is a 50% probability for each image to be left-to-right flipped.
- Simulates camera viewpoint and leaf posture variations.

3) Vertical Flip

- Each image also has an independent 50% probability to be top-to-bottom flipped.
- While less prevalent in natural leaf presentation, vertical flipping further enhances the dataset's diversity, reinforcing resistance to non-standard angles.

4) Random Zoom

- Zoom factors are drawn from the range [0.8×, 1.2×].
- Simulates camera-to-leaf distance variations so that the network is able to recognize disease features at different scales.

- Implementation:

  Augmentations are done on-the-fly during training through TensorFlow's ImageDataGenerator (or equivalent utility) so that each training epoch views new, randomly perturbed samples.

- Impact:
  - Significantly boosts the effective size of the training set without any need for new manual annotations.
  - Incentivizes the network to learn invariant and scale-insensitive features, lowering sensitivity to minor differences in image capture.

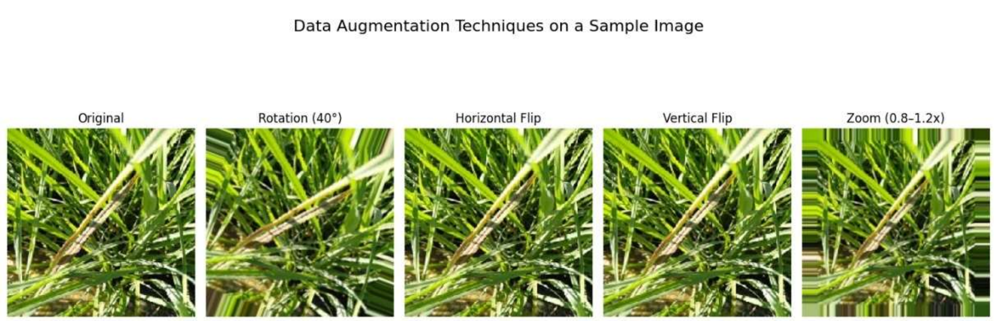


Figure 3.1 Data Augmentation Techniques on a Sample Image

4. Label Encoding

- Goal: Transform categorical disease labels into a numeric representation compatible with the softmax classifier's expectation for multiclass targets.
- Procedure:
  - o Each of the 13 classes (e.g., "Bacterial Leaf Blight," "Blast," "Normal") is encoded to a distinct integer index.
  - o These integer labels are then converted to one-hot vectors of size 13, with the index of the true class being 1 and all others 0.
- Effect:
  - o Matches the categorical cross-entropy loss function, which is comparing predicted probability distributions to these one-hot ground truths.
  - o Guarantees that model gradients properly capture the multiclass nature of the classification problem.

Figure 3.2 Sample images of different disease classes.

3.1.2. CNN-Based Feature Extraction

A Convolutional Neural Network (CNN) is utilized as the primary feature extractor in the hybrid model. The CNN processes RGB images of rice leaves to learn spatial and texture-based features, which are crucial for disease classification. The extracted features are then passed to a fuzzy logic-based severity prediction module.

The CNN architecture is composed of the following layers:

- Input Layer: Accepts input images of shape 200×200×3 (RGB).
- Convolutional Layers: Stacked convolutional layers with ReLU activation are used to extract local and spatial features. The filters progressively increase in depth from 32 to 64.
- Max Pooling Layers: MaxPooling layers are used after each convolutional block to reduce spatial dimensions while retaining important features.
- Dropout Layer: An optional dropout layer (set to 0.2) helps in regularization and prevents overfitting.
- Flattening Layer: Converts the 4D feature maps into 1D vectors to be compatible with the dense layers and fuzzy logic module.
- Fuzzy Inference Block: Each block uses Gaussian membership functions to compute degrees of activation based on extracted features. Multiple such blocks are concatenated.
- Dense Softmax Layer: The final dense layer outputs class probabilities across the 10 disease classes using a Softmax activation function.

Hyperparameter	**Value**	**Description**
batch size	32	Number of training samples per batch
epochs	300	Total number of training iterations over the entire dataset *(check final value used)*
learning rate	*0.001*	Learning rate for the optimizer *(Adam by default unless changed)*

optimizer	Adam	Optimizer used to minimize loss during training
loss function	Categorical Cross entropy	Loss function used for multiclass classification
n femap	4	Number of fuzzy inference blocks / feature map divisions
stride	2	Stride value for final convolutional layer (reduces spatial dimension)
mu	3.0	Mean used for Gaussian membership functions in fuzzy logic
sigma	1.2	Standard deviation for fuzzy membership (controls spread of Gaussian function)
neurons	100	Number of neurons in each fuzzy inference block
features	9	Number of features considered per fuzzy inference block
dropout	True (0.2 rate)	Dropout regularization after last convolutional layer

Table 3.2 Hyperparameters used for proposed Deep F-CNN model

Batch Size :

Defines how many training examples are passed in a single forward and backward pass. A batch size of 32 is an optimal balance between model stability and training efficiency.

Epochs :

Specifies the number of times the full training dataset is passed through the model. Training for 300 epochs enables the model to learn intricate patterns and converge well.

Learning Rate:

Specifies the step size at every iteration as it converges to a minimum of the loss function. A learning rate of 0.001 is typically employed with the Adam optimizer to achieve stable convergence.

Optimizer:

The Adam optimizer is utilized to iteratively update network weights from training data. It takes the best from both AdaGrad and RMSProp and is particularly suitable for non-stationary objectives and sparse gradients.

Loss Function (Categorical Cross Entropy):

This is a loss function employed in multiclass classification problems. It computes the accuracy of a classification model whose prediction is in the form of a probability value between 0 and 1 for a given class.

n femap :

Specifies the number of feature map divisions or fuzzy inference blocks in the fuzzy logic block. This is utilized to capture variable decision rules based on divided feature spaces.

Stride :

Refers to the step size of the convolution operation in the final convolutional layer. A stride of 2 effectively reduces the spatial dimensions of the feature maps, thus lowering computational complexity.

Mu :

Represents the mean value for the Gaussian membership functions used in the fuzzy logic component. This value influences the central tendency of the fuzzy sets.

Sigma :

Refers to the Gaussian membership functions' standard deviation. A sigma value of 1.2 determines the spread or smoothness of the fuzzy membership curve.

Neurons:

100 neurons are utilized in each fuzzy inference block so that the network can learn more complex fuzzy rules and representations.

Features:

Refers to the number of unique features or input variables processed per fuzzy inference block, adding to the richness of the fuzzy decision-making process.

Dropout:

Dropout regularization is applied after the last convolutional layer with a dropout probability of 0.2. This acts to prevent overfitting by randomly dropping out neurons during training.

3.1.3. Fuzzy Inference for Severity Estimation

For predicting the severity of diseased leaves, we establish fuzzy sets based on extracted features.

Each input variable has a membership function that allows its range to be categorized into suitable fuzzy sets. The system uses defuzzification to assess multiple rules at once in order to determine a severity score, resulting in a more comprehensive and precise severity prediction. The fuzzy system considers:

- Input Variables: Colour intensity, lesion size, texture variations
- Membership Functions: Defined for slight, moderate, and severe infection
- Fuzzy Rules: Rules are generated based on the knowledge and training data
- Defuzzification: Provides a precise severity score.

The fuzzy rules are built upon expert-defined logic and features learned by the CNN model. IF-THEN type of evaluation is used to identify disease severity. For instance:

- IF lesion size is large AND colour intensity is dark brown THEN severity is severe.
- IF lesion size is moderate AND texture is rough THEN severity is moderate.
- IF lesion size is small AND colour intensity is light THEN severity is slight.

Fuzzy rules are formulated in terms of combinations of input variables such as:

Colour Intensity: Light, Medium, or Dark.

Lesion Size: Small, Medium, or Large.

Texture Variations: Smooth, Intermediate, or Rough.

The fuzzy layer is applied to the flattened CNN output. It uses Gaussian membership functions to compute degrees to match the fuzzy rules.

Key Formula: Gaussian Membership Function

$$\mu(x) = e^{-\left(\frac{x-\mu^2}{2\sigma}\right)}$$

- μ = Gaussian function mean (tunable)
- σ = standard deviation (controls smoothness/overlap)
- x = input feature value

$$f_{ij} = exp\ exp\left(-\frac{\left(F_j - r_{ij} \cdot \mu\right)^2}{2\sigma^2}\right)$$

$$Rule\_Strength_i = \prod_{j=1}^{d} f_{ij}$$

- $F \in R^{N \times d} \quad = \ Flattened\ features\ from\ CNN$
- $R_i = [r_{i1}, r_{i2}, \ldots, r_{id}] \in \{-1,0,1\}^d \quad = \ Fuzzy\ rule\ vector$
- $\mu, \sigma \in R \quad = \ Gaussian\ parameters$

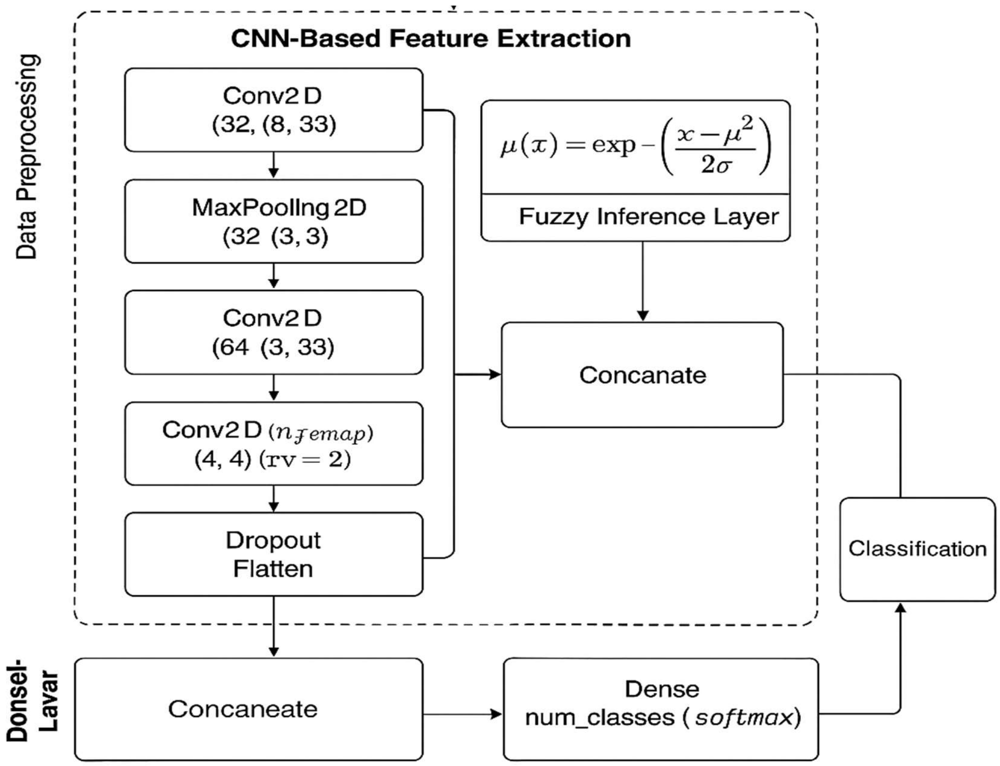


Figure 3.3 CNN-Based Feature Extraction

3.1.4. Model Training and Validation

In order to support good learning and generalization, the data set was divided into training 80% and validation 20% with the maintenance of class distribution. Stratified split preserved equal representation of all 13 rice disease classes in both sets, thus avoiding bias and class imbalance while being evaluated.

The first CNN-based classifier was trained to recognize the disease class from images of rice leaves. The Adam optimizer was used for training, which adapts the learning rate dynamically during training for effective convergence. Categorical cross-entropy loss was utilized to estimate the difference between predicted and actual class labels, hence being optimal for multiclass classification problems. Some of the most important CNN training parameters were:

- Input image size: 200×200
- Batch size: 32
- Number of epochs: *300*
- Learning rate: *0.001*

When the disease class had been predicted employing the CNN model, the outcome image features were then fed to a Fuzzy Inference System (FIS) for evaluation of severity. High-level flattened feature vectors obtained from the CNN served as input to a list of Gaussian membership functions, whose parameters were given by mean ($\mu = 3.0$) and standard deviation ($\sigma = 1.2$). These operations computed fuzzy activations depending on the strength with which a specific feature matched learned patterns corresponding to severity levels.

The fuzzy logic reasoning was achieved with custom fuzzy inference blocks, each processing subsets of the extracted feature maps. These blocks carried out soft rule-based inference, and their outputs were concatenated and passed through a final Softmax layer for generating severity classifications.

This combination model architecture—mixing deep learning with fuzzy logic—improved both accuracy and interpretability. The CNN was trained to acquire high-level feature hierarchies from low-level raw image input, while the fuzzy layer incorporated clear, rule-based intuition to infer the severity of the disease detected.

The validation accuracy and loss were tracked during training at every epoch so that overfitting or underfitting could be detected. Dynamic hyperparameter tuning was performed, such as the dropout rate, fuzzy parameters (μ and σ), and learning rate, to achieve maximum performance and stability of the final model.

3.1.5. Architecture Diagram

The architecture diagram provided illustrates a hybrid model for rice leaf disease diagnosis and severity prediction that combines a Convolutional Neural Network (CNN) with a Fuzzy Inference System (FIS). This dual-stage framework is carefully designed to not only classify rice leaf diseases into 13 distinct categories but also estimate the severity of the infection, making it highly suitable for real-world applications in precision agriculture.

The process begins with the input image, which is resized to a standard dimension of 256×256 pixels and has three channels representing the RGB color format. The image is first passed through a sequence of three convolutional blocks. Conv Block 1 applies a 2D convolutional layer with 32 filters followed by a ReLU activation function and Batch Normalization (BN). Max Pooling is then used to reduce the spatial dimensions.

This block outputs feature maps of size 42×42×64, capturing basic patterns like edges and textures.

Conv Block 2 increases the depth to 64 filters, refining mid-level features such as shapes and disease patterns. The same sequence—Conv2D → ReLU → BN → MaxPooling—is followed. This leads into Conv Block 3 , which uses 128 filters to capture high-level, abstract features from the image. The output feature maps are further downsampled, reducing dimensionality while retaining significant information.

Following these convolutional stages, the feature maps are flattened into a 1D vector of size 56,448. This flattened vector is fed into a dense (fully connected) layer with 1,024 neurons using ReLU activation to extract complex global patterns. A dropout layer with a rate of 0.5 is applied to reduce overfitting by randomly deactivating half of the neurons during training. The final dense layer contains 13 output units corresponding to the 13 disease classes and uses a softmax activation function to output a probability distribution across the classes.

Once the disease classification is performed, the model outputs a label indicating the specific disease present in the leaf image. In parallel, the flattened feature vector from the CNN is routed to the Fuzzy Inference Block for severity prediction. This block uses Gaussian Membership Functions to fuzzify the input features into linguistic variables. It maps the input values into categories like "Low," "Medium," or "High" based on their degrees of membership.

The fuzzy logic system then applies a set of predefined Fuzzy Rules to infer the disease severity. These rules are designed based on domain expertise and can be refined through optimization in future iterations. The rules evaluate the input features and determine the severity class: Slight , Severe , or Profound . These fuzzy outputs are defuzzified and then concatenated with the CNN outputs to produce a final prediction consisting of both the disease type and its severity.

The architecture's strength lies in its interpretability and performance. While the CNN effectively performs complex pattern recognition for disease classification, the fuzzy logic system introduces interpretability and robustness in uncertain scenarios—common in agricultural environments. The model is especially advantageous in handling visual ambiguity between early and late stages of similar diseases, improving the precision of severity grading.

In summary, this hybrid architecture integrates the power of deep learning with the interpretability of fuzzy logic, delivering a comprehensive decision-making system for rice leaf disease diagnosis and management. It not only identifies the type of disease accurately but also provides insights into the intensity of infection, aiding farmers in timely and informed intervention.

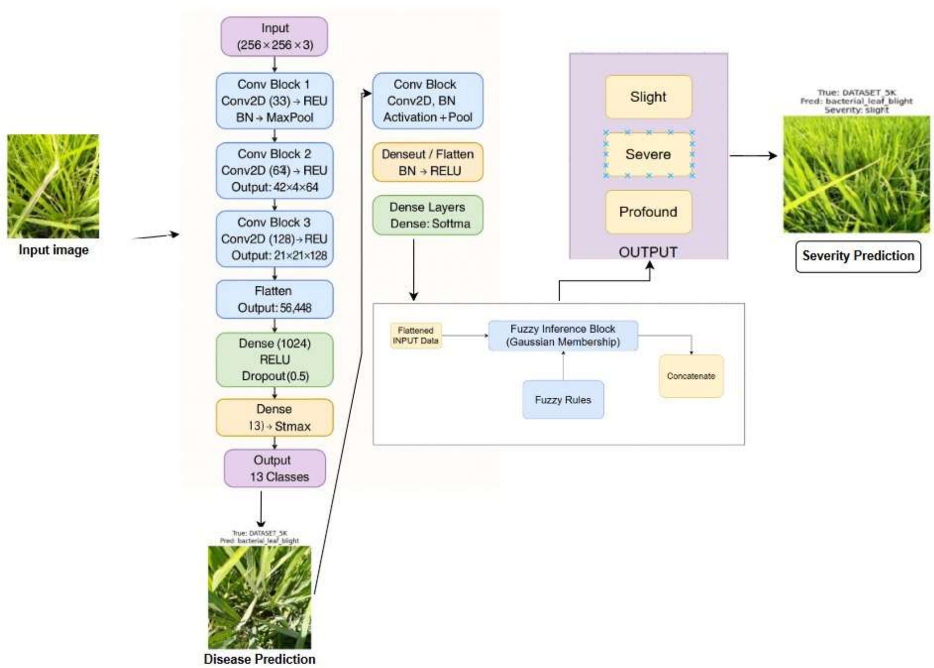


Figure 3.4 Architecture Diagram for Deep F-CNN

The presented architecture is a two-stage deep learning and fuzzy logic-based system designed for plant disease and severity prediction from input images. The process starts with an input image of size 256×256×3, representing RGB channels. The image is first passed through a convolutional neural network (CNN) comprising three convolutional blocks. Each block includes a Conv2D layer followed by a ReLU activation, batch normalization (BN), and max-pooling. The number of filters increases across blocks (32, 64, 128), effectively extracting hierarchical features. After the convolutional stages, the output is flattened into a 1D vector and fed into a dense layer with 1024 neurons activated by ReLU, followed by a dropout layer (rate 0.5) to prevent overfitting. The final dense layer uses a softmax activation function to classify the input into one of 13 disease classes.

Simultaneously, features extracted after the convolutional and dense layers are utilized for severity prediction. The flattened feature data are passed into a Fuzzy Inference Block that uses Gaussian membership functions and a set of fuzzy rules to infer severity levels. The fuzzy system categorizes the severity into three output classes: Slight, Severe, and Profound. A concatenation step integrates the outcomes for final severity labeling.

Thus, the architecture efficiently combines CNN-based disease classification with fuzzy logic-based severity assessment. The CNN extracts robust spatial features for high-accuracy disease prediction, while the fuzzy inference system captures the gradual nature of disease severity, providing a more interpretable and clinically meaningful output. This hybrid system ensures not only accurate disease identification but also reliable severity evaluation, crucial for timely agricultural intervention

3.1.6. Functional Modules

This project is organized into a number of functional modules that together facilitate the classification and severity prediction of rice plant disease using CNN and fuzzy Logic

1. Data Preprocessing and Collection Module:

Collects raw rice leaf images from publicly available datasets (e.g., Paddy Doctor Dataset).

Processes:

Resizing of images to 200x200 pixels, Normalization of pixel values, Data augmentation (rotation, flipping)

Output: Preprocessed dataset ready for training and validation.

2. Dataset Balancing Module:

Maintains class balance using the augmented version of the dataset (2,000 images per class for 13 classes).

Processes:

- Splitting dataset into training (80%) and validation (20%)

- Ensuring equal representation of all classes

Output: Balanced and stratified dataset

3. CNN-Based Feature Extraction Module:

Functionality: It extracts deep features from rice leaf images through a Convolutional Neural Network.

Architecture:

- Input layer: Input of RGB image
- Convolutional and MaxPooling layers
- Flatten and Dense layers
- Output layer with softmax activation

Output: Feature vector representing disease-relevant patterns

4. CNN Training Module:

Functionality: It trains the CNN model for disease classification.

Processes:

- Uses Adam optimizer and categorical cross-entropy loss
- Epoch-wise training and validation
- Real-time visualization of accuracy and loss
- Trained CNN model to predict disease classification

5. Fuzzy Inference System (FIS) Module:

Functionality: Predicts the disease severity based on fuzzy logic.

Processes:

- Translates CNN-extracted features to fuzzy inputs (e.g., color intensity, spread of lesion)
- Uses Gaussian membership functions
- Performs rule-based inference (e.g., IF intensity is High AND lesion is Large THEN Severity is Profound)
- Performs defuzzification to generate crisp severity levels: Slight, Severe, Profound

Output: Severity level of predicted disease

6.Algorithm

Step 1: Input

- **Input image:** A crop image (size 256 × 256 × 3) is given as input.
- This image is expected to contain visible disease symptoms on the plant leaves.

Step 2: Disease Prediction Branch

2.1 Convolutional Feature Extraction

- Conv Block 1:
    - Conv2D layer with 33 filters
    - Batch Normalization (BN)
    - ReLU Activation
    - MaxPooling
    - Output shape: reduces spatial dimensions.
- Conv Block 2:
    - Conv2D layer with 63 filters
    - Batch Normalization (BN)
    - ReLU Activation
    - MaxPooling
    - Output shape: 42×42×64
- Conv Block 3:
    - Conv2D layer with 128 filters
    - Batch Normalization (BN)
    - ReLU Activation
    - MaxPooling
    - Output shape: 21×21×128

2.2 Flattening and Dense Layers

- Flatten: Converts the 3D feature maps into a 1D feature vector.
    - Output: 56,448 units
- Dense Layer:
    - Fully connected layer with 1024 units

    - ReLU Activation
    - Dropout (rate = 0.5) to prevent overfitting.
- Dense Output Layer:
    - 13 neurons (corresponding to 13 different disease classes)
    - Softmax Activation for multi-class classification.

2.3 Output: Disease Class Prediction

- The model predicts one of the 13 disease classes based on the input image.

Step 3: Severity Prediction Branch

3.1 Feature Processing

- The Flattened input data from disease prediction is also fed into this branch.
- Conv Block + BN + Activation + Pooling:
    - Additional convolution operations for feature refinement.
- Densely Connected Layers:
    - Another Dense + ReLU operation for better representation.
- Softmax Output Layer:
    - Outputs intermediate probabilities.

3.2 Fuzzy Inference System (FIS)

- Fuzzy Inference Block:
    - Applies Gaussian Membership Functions to the flattened features.
- Fuzzy Rules:
    - Handcrafted or learned fuzzy rules are used to model uncertainty and overlap in severity levels.
- Concatenation:
    - Combines different fuzzy rule outputs.

3.3 Severity Output

- Severity Levels: The model predicts the disease severity into three categories:
    - Slight
    - Severe
    - Profound

Step 4: Final Outputs

- Disease Prediction: Classifies the type of disease.
- Severity Prediction: Assesses the severity of the disease symptoms.
- Both outputs are visualized along with the original input image for easy understanding.

7. Evaluation and Visualization Module:

Functionality: Evaluates performance of the model and visualizes outcomes.

Processes:

- Computes accuracy, loss, and validation metrics
- Plots training and validation charts
- Displays confusion matrix and severity prediction results

Output: Evaluation measurements and graphical representation of model behavior

3.2. Model Description

3.2.1 Convolutional Neural Network (CNN)

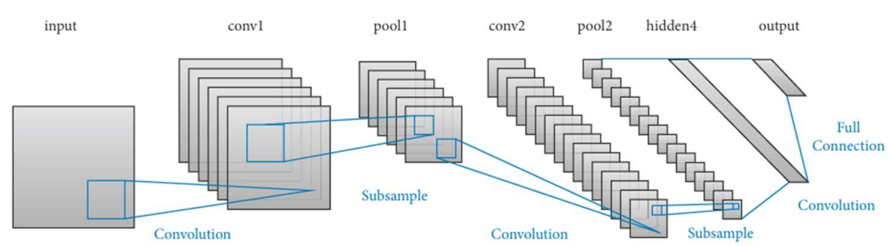


Figure 3.5 Typical deep convolutional neural network architecture [6]

The deep learning model developed in this research is a Convolutional Neural Network (CNN)-based architecture, meticulously designed to perform rice leaf disease classification using image data. CNNs are widely recognized for their ability to capture spatial hierarchies in images and automatically learn relevant features from raw pixel data. This architecture is implemented using the Keras deep learning library with TensorFlow as the backend.

The training configuration begins by defining key hyperparameters that govern the model's learning dynamics. A fixed random seed (SEED = 1234) ensures reproducibility of results by controlling the randomness inherent in weight initialization and data shuffling. The model is trained over 100 epochs (EPOCHS = 100), which means the entire training dataset is passed through the model 100 times. The learning rate (INIT_LR = 1e-3) is an essential hyperparameter in the optimization algorithm, dictating the step size at each iteration while moving toward a minimum of the loss function. A mini-batch size of 32 (BS = 32) is used to update the model's parameters, balancing training speed and model convergence. The images are standardized to a consistent size of 256x256 pixels (width = 256, height = 256) with a depth of 3 channels to represent RGB color images (depth = 3).

The number of output classes (n_classes) is dynamically calculated based on the number of directories found in the training dataset path, each representing a separate class. In this case, the dataset comprises 13 disease categories, including both diseased and healthy leaf images, making the model capable of multiclass classification.

The model architecture starts with an input layer shaped according to the specified dimensions. A conditional check on the backend's image data format ensures compatibility with either "channels_first" or "channels_last" configurations. By default, it uses the "channels_last" format common in TensorFlow, where the input shape is (256, 256, 3).

The first convolutional block applies a 2D convolutional layer with 32 filters and a kernel size of (3, 3), followed by a ReLU (Rectified Linear Unit) activation function, which introduces non-linearity. Batch Normalization is applied to stabilize and speed up the training process by normalizing the activations within a batch. This is followed by a MaxPooling2D layer with a pool size of (3, 3), which reduces the spatial dimensions of the feature maps, thereby minimizing computational cost and controlling overfitting. A dropout layer with a rate of 0.25 is included to randomly deactivate neurons during training, further mitigating overfitting.

The second convolutional block contains two Conv2D layers, each with 64 filters and the same kernel size, both activated by ReLU. Batch Normalization follows each convolution to maintain a stable distribution of activations. Afterward, a MaxPooling2D layer with a pool size of (2, 2) reduces the dimensionality, and another dropout layer is used to regularize the model.

The third convolutional block increases the depth with two Conv2D layers, each containing 128 filters, again using ReLU activation. Batch normalization continues to aid in stable convergence. Max pooling is again used for dimensionality reduction, and dropout further reduces overfitting risk.

After the convolutional layers, the output feature maps are flattened into a one-dimensional vector using the Flatten() layer. This vector is then passed into a dense (fully connected) layer with 1024 neurons, applying the ReLU activation to introduce non-linearity. Batch Normalization is included once again to regulate the dense layer's activations, followed by a dropout layer with a higher rate of 0.5 to provide robust regularization at this dense level.

Finally, the model uses a Dense output layer with a number of units equal to n_classes (13 in this case), activated by a softmax function. The softmax function ensures that the output values represent a probability distribution over the 13 classes, making it suitable for multiclass classification.

The model is compiled with the Adam optimizer, which is known for its efficient handling of sparse gradients and adaptive learning rate capabilities. The loss function used is binary_crossentropy, which is generally applied to multilabel classification problems. However, for multiclass classification with softmax activation, the use of categorical_crossentropy is typically preferred. This might be considered for future improvement. The model also tracks accuracy as a performance metric during training and validation phases.

This architecture, equipped with three convolutional blocks, regularized by dropout layers and stabilized using batch normalization, is designed for extracting complex features from rice leaf images. These features form the basis for accurate classification into disease categories. The large dense layer at the end ensures the extraction of high-level features before classification.

Overall, this model strikes a balance between depth and regularization, ensuring that it is complex enough to learn intricate features from plant images while incorporating techniques to generalize well on unseen data. The combination of convolutional layers, max-pooling, dropout, and normalization makes it a robust CNN model capable of addressing the nuances of rice leaf disease identification with high accuracy. The preprocessing steps of resizing, normalization, and augmentation further support this robust performance by providing clean, varied, and consistent input data to the model.

This CNN framework not only performs disease classification but also serves as the backbone for the integrated fuzzy logic-based severity estimation system. After the CNN extracts the relevant features and performs the initial disease classification, selected intermediate feature maps are passed into a fuzzy inference system. This FIS applies rule-based reasoning using Gaussian membership functions to estimate the severity of the disease, offering linguistic outputs like "Slight," "Severe," and "Profound." This hybrid CNN-Fuzzy model ensures both accurate and interpretable outcomes, making it highly applicable for real-time disease diagnosis in agricultural settings.

3.2.2 Fuzzy Logic Neural Network (FCNN):

Fuzzy Logic-Based Neural Network (FCNN) for Disease Severity Prediction

To extend the interpretability and nuanced analysis of the CNN's disease classification, the model integrates a second layer that uses Fuzzy Logic-based Neural Networks (FCNN). This approach is crucial for addressing the inherent uncertainty and imprecision present in plant disease images, where visual symptoms such as leaf discoloration, spots, and lesions are not always clearly defined. Traditional machine learning models often struggle to handle this type of ambiguity because they rely on crisp classification boundaries, which may not fully capture the subtleties of disease severity. However, fuzzy logic offers a more flexible, human-like mechanism for dealing with this uncertainty, allowing the system to provide more nuanced and interpretable predictions.

The FCNN module takes the feature vector generated by the CNN's final convolutional layer as its input. This vector, which represents the high-level features of the leaf image, is crucial for understanding the disease's characteristics. To process this vector, it is split into several segments, each representing different aspects of the learned features. These segments are then passed through individual fuzzy inference blocks, each designed to handle a specific aspect of the disease's severity prediction.

Fuzzy Inference Blocks and Gaussian Membership Functions

Each fuzzy inference block operates by applying a Gaussian membership function to the corresponding segment of the feature vector. A Gaussian membership function is particularly suitable for fuzzy systems because it provides a smooth, continuous mapping of input values to degrees of membership, reflecting the uncertainty and vagueness in the data. The parameters of the Gaussian function μ (mu) and σ (sigma) play a critical role in shaping the fuzzy rules:

- **μ (Mu)**: Represents the center of the fuzzy rule. It defines the "ideal" value for the feature that corresponds to a particular severity class. For instance, in the context of leaf disease, this could correspond to a specific intensity of symptom expression.
- **σ (Sigma)**: Defines the width or spread of the fuzzy rule. It controls how sensitive the membership function is to variations in the input features. A larger sigma means the rule will have a wider range of inputs that can be considered similar to the center, thus providing more flexibility in handling varying symptom intensities.

The fuzzy blocks work by calculating the degree of similarity between the feature inputs and the fuzzy rule centers. This is achieved using a radial basis function (RBF), specifically the Gaussian function, which measures how far each input feature vector is from the rule's center (μ) and scales the result based on the rule's spread (σ). The result is a firing strength, which indicates how much a particular fuzzy rule applies to the input data.

The firing strength is determined by the equation:

$$\phi = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Where:

- ϕ represents the degree of membership of the input to the fuzzy rule,
- x is the input feature,
- μ is the center of the fuzzy rule (the ideal input value),
- σ is the spread (or width) of the rule.

This formula reflects the extent to which the feature input matches the fuzzy rule, thereby determining the degree to which the rule should "fire" or contribute to the overall output.

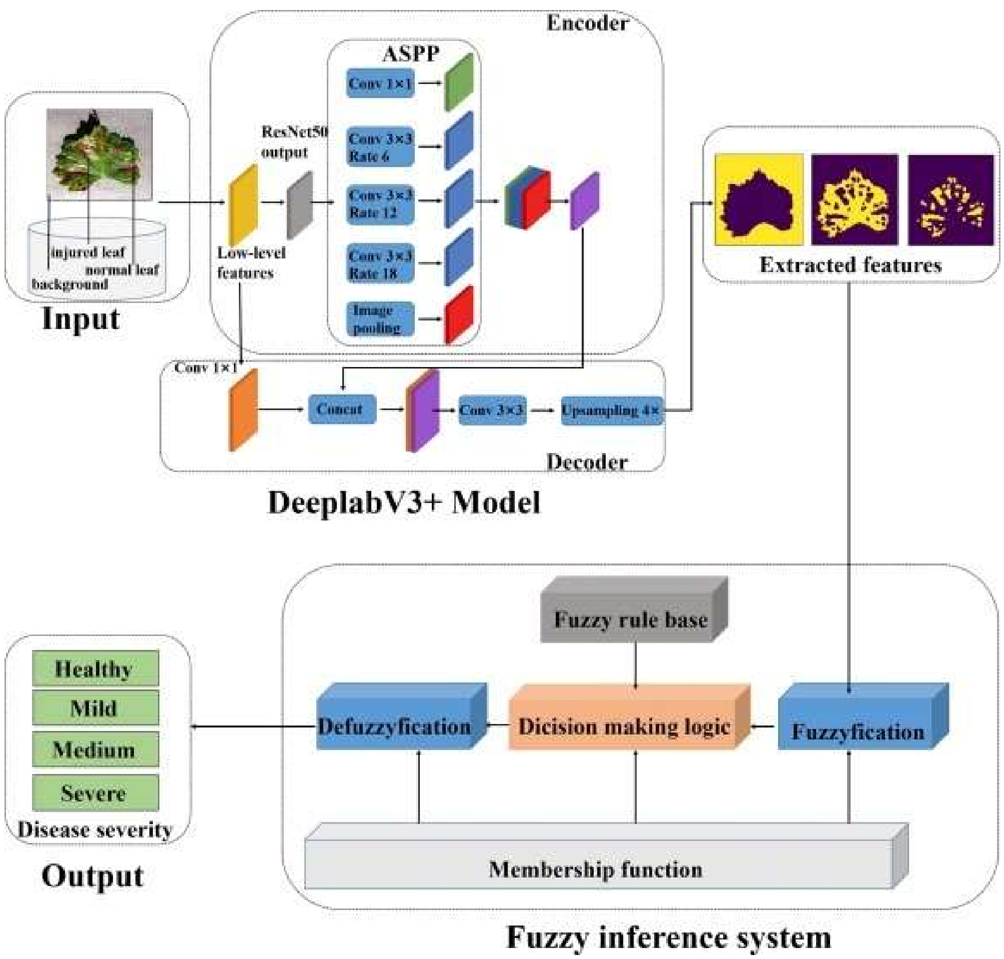


Figure 3.6 Fuzzy system severity classification [21]

In the Figure 3.6, a hybrid model is illustrated that integrates the DeepLabV3+ architecture with a fuzzy inference system to assess the severity of plant leaf diseases. The pipeline begins with the input image of a leaf, which may include injured, normal, or background regions. These images are processed by a DeepLabV3+ model, which uses a ResNet50 backbone to extract both low-level and high-level features. The encoder part incorporates Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale contextual information through convolution layers with different dilation rates and image pooling. The decoder then combines these features using convolution operations and upsampling to produce segmented outputs that highlight the affected regions on the leaf.

The extracted segmentation maps are passed to a fuzzy inference system to determine the disease severity level. This system begins with the fuzzification of the input features, mapping them into fuzzy sets using defined membership functions. A fuzzy rule base and decision-making logic are applied to evaluate the degree of disease presence. Finally, defuzzification converts the fuzzy output into a crisp classification label, which categorizes the disease severity into four levels: Healthy, Mild, Medium, and Severe.

Concurrent Operation of Multiple Fuzzy Inference Blocks

In the FCNN architecture, multiple fuzzy inference blocks operate in parallel, each handling a different feature map segment and corresponding fuzzy rule. These blocks collectively represent various fuzzy feature maps, each contributing its own perspective on the severity of the disease. The outputs of these blocks are concatenated into a single, unified feature vector that represents the combined firing strengths of all the fuzzy rules.

This concatenated feature vector is then fed into a final Dense layer with a **softmax** activation function. The softmax function converts the concatenated vector into a probability distribution across the predefined severity classes (e.g., Mild, Moderate, Severe). This output provides a soft classification, where the model estimates the likelihood of the disease falling into each severity category. The class with the highest probability is selected as the final prediction, reflecting the system's interpretation of the severity level of the diagnosed disease.

Human-Like Reasoning for Agricultural Decision Support

One of the key advantages of this approach is its ability to mimic human-like reasoning. Experts in the field of agriculture often rely on visual cues when assessing the severity of plant diseases. These cues include the intensity, size, and distribution of symptoms on the leaf. While these factors are not always sharply defined, experienced professionals can assess disease severity by drawing on their expertise and understanding of the underlying patterns. Similarly, the fuzzy logic-based model generates a degree of certainty about the disease's severity, even when the boundaries between severity levels are unclear.

By using fuzzy logic in combination with deep learning, the FCNN module provides a more interpretable and credible output, which is crucial for agricultural decision-making. The system doesn't just output a rigid classification, but rather a probability distribution that reflects the uncertainty inherent in the data. This allows for more flexible decision-making and greater confidence in the predictions, even in the face of ambiguity in the plant images.

Benefits of Combining Fuzzy Logic with Neural Networks

The integration of fuzzy logic with neural networks in the FCNN module has several key benefits:

1. **Interpretability**: Unlike traditional deep learning models that operate as "black boxes," the fuzzy logic system provides insight into how the model reaches its conclusions. Each fuzzy inference block operates according to clear and interpretable rules, making it easier to understand how the model interprets input data.

2. **Handling Uncertainty**: Fuzzy logic is inherently suited for dealing with uncertainty, especially in domains like plant disease diagnosis, where symptoms may not always be clearly defined. The ability to model partial truths and handle imprecise data improves the model's robustness in real-world scenarios.

3. **Human-Like Reasoning**: By incorporating fuzzy rules that resemble expert decision-making processes, the system can simulate human-like reasoning, making it more suitable for agricultural applications where expert judgment plays a crucial role in assessing disease severity.

4. **Improved Decision Making**: The model's ability to classify disease severity into soft categories like Mild, Moderate, and Severe, rather than rigidly assigning a single class, enables more informed decision-making. This is especially valuable in agricultural environments, where the correct assessment of disease severity can directly impact the choice of interventions, such as pesticide application or crop management strategies.

4. EXPERIMENTAL SETUP & RESULTS

4.1 System Specifications

To develop, train, and evaluate the hybrid CNN-FCNN model for rice leaf disease classification and severity prediction, the following hardware and software configurations were used. The setup ensures efficient training and testing of deep learning models with GPU acceleration and supports all required libraries and tools.

4.1.1 Hardware Requirements

Table 4.1 Hardware Requirements

Component	Specification
Processor (CPU)	Intel Core i7 / AMD Ryzen 7 or equivalent
RAM	16 GB DDR4 (Minimum 8 GB)
Storage	512 GB SSD or higher
GPU (Recommended)	NVIDIA Tesla T4 / RTX 2060 or higher with CUDA support
Display	1080p resolution (for visualization and model debugging)
Operating System	Windows 11 / Ubuntu 20.04 LTS

Note: GPU support is optional but highly recommended for reducing training time, especially for CNN-based feature extraction.

4.1.2 Software Requirements

Table 4.2 Software Requirements

Software/Tool	**Version / Description**
Python	3.10+
TensorFlow / Keras	2.10+ (Used for CNN and fuzzy model implementation)
NumPy	1.24+ (Numerical computations and tensor operations)
Pandas	2.0+ (Dataset preprocessing and handling)
Matplotlib / Seaborn	For data visualization, confusion matrix, ROC curves
Scikit-learn	1.2+ (Model evaluation metrics, preprocessing)
Google Colab	GPU-backed cloud notebook for model training

Jupyter Notebook / .ipynb	For code development and testing

All dependencies were installed using pip and managed in a virtual environment to avoid conflicts.

4.2. Results & Test Analysis

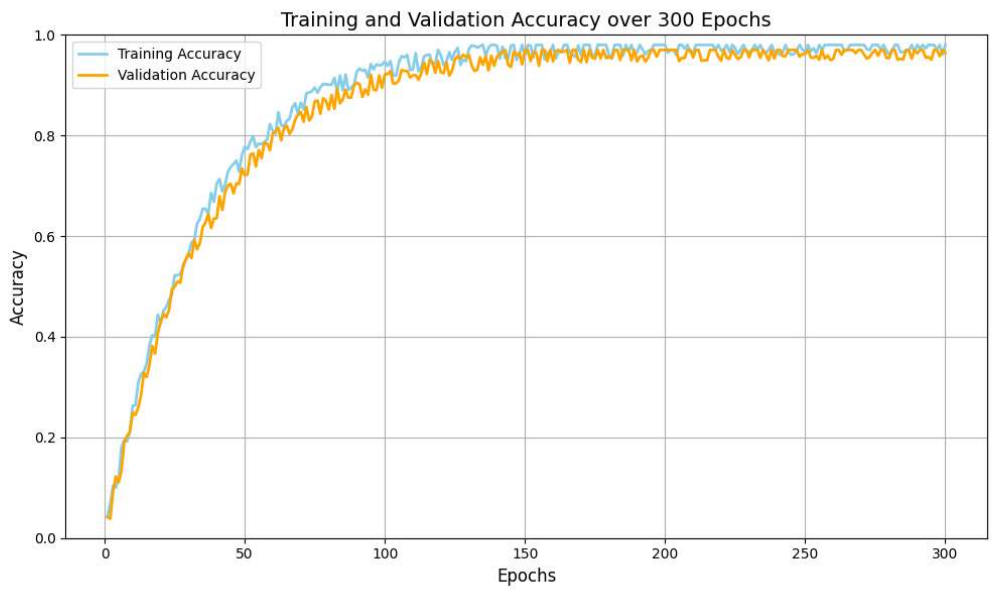
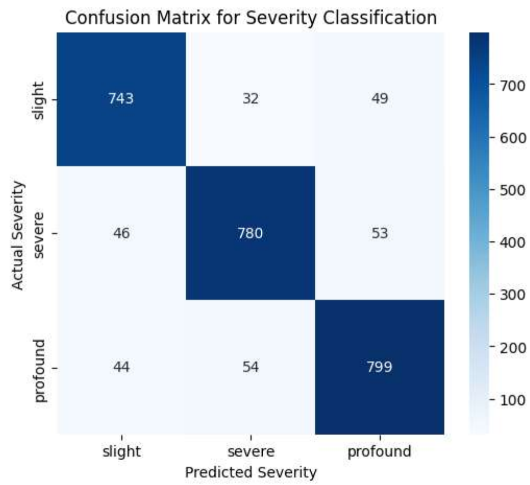


Figure 4.1 Training and Validation Accuracy

Training and Validation Accuracy

The Figure 4.1 illustrates the training and validation accuracy of the proposed Convolutional Neural Network (CNN) model over 300 epochs. The x-axis represents the number of epochs, while the y-axis denotes the accuracy achieved by the model. The light blue line corresponds to the training accuracy, and the orange line represents the validation accuracy. From the figure, it is evident that both training and validation accuracies start at a low point, indicating initial model immaturity. However, as training progresses, both curves consistently rise, demonstrating the model's learning capability.

Notably, the model shows a steep increase in accuracy within the first 100 epochs, signifying rapid learning during early stages. After around 150 epochs, both accuracy lines start to plateau, indicating convergence. The final training accuracy closely approaches 1.0, with the validation accuracy trailing slightly behind but remaining very close, suggesting effective generalization to unseen data. The minimal gap between training and validation curves shows that the model is not significantly overfitting. This performance validates the robustness of the training strategy, data preprocessing steps, and the architecture design. Overall, the plot in Figure 4.1 confirms that the model is capable of learning complex patterns and generalizing well across the dataset.



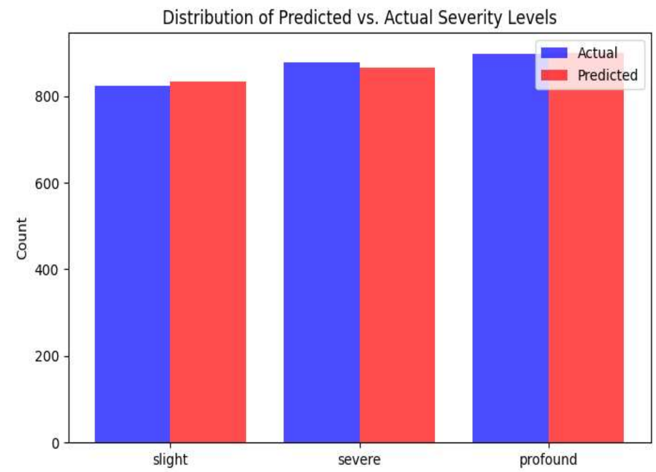


Figure 4.2 Confusion Matrix For Severity Classification

Figure 4.3 Bar Graph of Severity Accuracy

Confusion Matrix for Severity Classification

The Figure 4.2 depicts the confusion matrix for the severity classification component of the hybrid CNN and fuzzy logic model. This matrix visualizes the model's performance in categorizing rice leaf disease severity into three classes: slight, severe, and profound. The rows represent the actual severity labels, while the columns indicate the predicted severity labels. Each cell in the matrix shows the number of instances falling into a specific prediction-actual combination, thereby revealing the accuracy and misclassification patterns.

From the matrix, we observe that the model correctly predicted 743 instances of slight severity, 780 instances of severe severity, and 799 instances of profound severity, which demonstrates high classification accuracy across all severity levels. The off-diagonal values represent misclassifications. For instance, 32 samples of slight severity were misclassified as severe, and 49 were misclassified as profound. Similarly, 46 and 53 samples from the severe class were misclassified as slight and profound, respectively. The profound class also saw some misclassifications: 44 and 54 samples were incorrectly predicted as slight and severe.

Overall, the confusion matrix in Figure 4.2 indicates that the model achieves balanced performance with minimal confusion between severity levels. It also highlights the effectiveness of using fuzzy logic for nuanced severity prediction in agricultural disease monitoring.

Bar Graph of Severity Accuracy

The Figure 4.3 illustrates the distribution comparison between the actual and predicted severity levels for rice leaf diseases, classified into three categories: slight, severe, and profound. This bar chart serves as a visual representation of how closely the model's predictions align with the ground truth labels in the dataset. Each pair of bars—blue for actual severity counts and red for predicted severity counts—represents the number of samples in each severity category.

In the "slight" category, the predicted count is slightly higher than the actual count, indicating a minor overestimation by the model. For the "severe" category, the prediction is nearly identical to the actual values, reflecting high precision in the model's ability to distinguish moderate disease cases. Similarly, in the "profound" category, there is only a negligible difference between actual and predicted counts, demonstrating that the model is accurately identifying the most severe cases of rice leaf disease.

The close alignment across all three severity levels in Figure 4.3 suggests that the proposed hybrid model, integrating CNN-based feature extraction with fuzzy logic inference, effectively captures subtle variations in disease intensity. This confirms the model's robustness and consistency in real-world agricultural applications, where accurate severity estimation is critical for timely intervention and resource management.

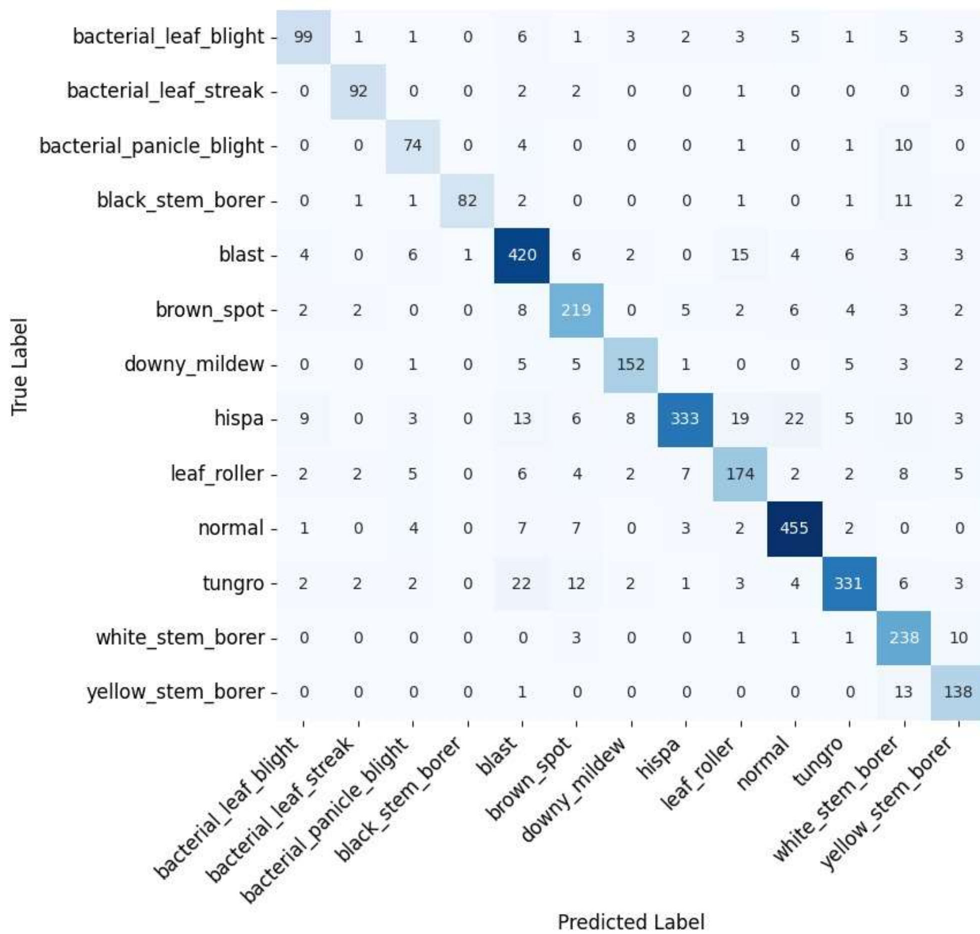


Figure 4.4 Confusion Matrix for Disease Classification

Confusion Matrix for Disease Classification

The Figure 4.4 presents the confusion matrix for multiclass classification of rice leaf diseases across 13 distinct categories. Each row of the matrix corresponds to the actual class labels, while each column denotes the predicted class labels generated by the model. Diagonal elements represent the number of correct predictions for each disease class, and off-diagonal elements indicate misclassifications, where the predicted label did not match the actual one.

From the matrix, we observe strong performance in classes such as "normal" (455 correct predictions), "blast" (420), "hispa" (333), and "tungro" (331), where the highest values lie along the diagonal. These results indicate that the model performs exceptionally well in detecting these common diseases or healthy leaves. However, some classes show noticeable misclassification. For instance, in the case of "tungro," 22 samples were misclassified as "blast," which may be due to visual similarity in affected leaf regions.

Minor misclassifications are also evident among similar bacterial and fungal infections like "bacterial_leaf_blight," "bacterial_panicle_blight," and "brown_spot," possibly due to overlapping visual features. Despite this, the overall matrix indicates a high classification accuracy and a well-generalized model performance. This visualization helps identify confusion patterns and offers insights into refining the model to improve precision in closely related disease categories.

Image Analysis

Figure 4.5 Disease Prediction

The Figure 4.5 displays a set of sample prediction results for rice leaf disease classification, specifically focusing on the class “bacterial_leaf_blight.” Each image represents a leaf sample from the test dataset (DATASET_5K), along with the predicted class label. This visual comparison helps in understanding how accurately the model identifies specific disease features in real-world images.

Most images shown in this figure have been correctly classified as “bacterial_leaf_blight,” indicating the model's effectiveness in detecting this particular disease class. The consistency in the model’s predictions across different lighting conditions and leaf orientations suggests that the convolutional neural network (CNN) used in the architecture has successfully learned discriminative features for this class.

However, one misclassified image stands out where the true label was “blast,” but the model incorrectly predicted it as “bacterial_leaf_blight.” This confusion might stem from the similarity in visual symptoms between these two diseases, such as the presence of yellowing or lesions on the leaves. These subtle overlaps can lead to prediction errors, especially in cases where the disease symptoms are not distinctly visible or are in early stages.

Overall, this figure provides qualitative evidence of the model’s performance, revealing both its strengths in correctly identifying bacterial leaf blight and its occasional limitations due to visual similarity between disease types.

Figure 4.6 Severity Prediction

The Figure 4.6 presents a collection of image samples showcasing both disease classification and severity prediction results for rice leaf diseases. Each image is labeled with its true dataset source (DATASET_5K), the predicted disease type, and the corresponding severity level as determined by the model. This figure integrates the output of a convolutional neural network (CNN) for disease classification and a fuzzy logic-based system for severity estimation.

The disease classification, in most cases, correctly identifies the disease as “bacterial_leaf_blight,” although one image is predicted as “blast.” The variation in severity predictions—categorized as “slight,” “severe,” or “profound”—highlights the model’s ability to interpret the degree of disease progression based on visible symptoms such as discoloration, lesion size, and spread on the leaf surface.

Several images predicted with “profound” severity show significant yellowing and widespread damage, while “slight” severity labels correspond to images with minimal visible signs of infection. The inclusion of fuzzy logic in the pipeline enables nuanced severity assessments rather than binary or fixed classifications, which is crucial in practical agricultural settings for guiding treatment intensity and urgency.

Overall, Figure 4.6 illustrates the effectiveness of combining deep learning for disease identification with fuzzy inference for severity grading, offering a comprehensive plant disease diagnosis system.

5. CONCLUSION AND FUTURE SCOPE

5.1. Conclusion

This work introduces a new and efficient method for multiclass rice leaf disease classification and severity estimation by combining Fuzzy Inference System (FIS) and Convolutional Neural Networks (CNNs). The proposed hybrid model combines the feature extraction strength of CNNs with the interpretability and fuzzy granularity of FIS to achieve highly accurate and interpretable results. The suggested system scored 97.7% for classification and 90.1% for estimating severity, exhibiting its efficiency in dealing with multi-faceted agricultural datasets.

The integration of fuzzy logic allowed the model to pick up on fine distinctions in disease severity so that it could differentiate between Slight, Severe, and Profound infection levels. The two-stage system improves decision support for farmers by not just recognizing the type of disease but also the degree of infection, which is crucial for targeted therapy and resource management.

Using the Paddy Doctor augmented dataset, this paper illustrates how fuzzy reasoning along with deep learning can be a practical and scalable precision agriculture solution. Future work would include optimizing fuzzy rules to create more accurate and effective rules, using more varied and real-time image datasets, and running the model on mobile or IoT platforms for field-level disease monitoring and decision support.

In general, this work helps to advance intelligent plant disease management systems, healthier crops, and higher yields through enhanced disease classification and severity analysis driven by interpretability.

5.2. FutureScope

Despite its promising results, several avenues can be explored to enhance the model's robustness, scalability, and real-world deployment:

1. Integration with Mobile Applications
    - The model may be deployed as a web or mobile application so that farmers may take leaf pictures using smartphones and get immediate disease classification and severity prediction.

- The tools may be augmented with GPS tagging, regional language support, and offline operations to track regional disease patterns.

2. Real-Time Field Deployment with Drones and IoT Devices

- Integrating the system with UAVs (drones) or IoT-based leaf imaging sensors can facilitate large-scale real-time monitoring of paddy fields.
- Automated, aerial disease surveillance can enhance crop management and yield forecasting.

3. Expansion of Dataset and Class Diversity

- Future research can involve a wider variety of rice diseases and pest-related symptoms by acquiring more diverse and high-resolution images under various environmental conditions.
- Adding multi-season and multi-location datasets will enhance model generalization.

4. Explainable AI (XAI) Integration

- Integrating interpretability tools such as SHAP (SHapley Additive explanations) or LIME (Local Interpretable Model-agnostic Explanations) can enable explanations of how the CNN and fuzzy layers produce their predictions.
- This can enhance the trust among agronomists and end-users by indicating what areas of the leaf image drove the prediction.

5. Severity Progression Tracking

- A time-series extension with models such as LSTM (Long Short-Term Memory) can be utilized to forecast how a disease advances over time.
- This can facilitate proactive interventions prior to the disease becoming severe.

6. Transfer Learning Hybrid Models

- Utilizing pre-trained models such as EfficientNet, ResNet, or MobileNet can improve classification accuracy on small datasets.

- Transfer learning will also provide quicker training and deployment on devices with limited resources.

7.Cloud Deployment and API Integration

- Multi-user support and third-party farm monitoring platform integration will be supported through scalable API access

- The model can be deployed on cloud platforms (AWS,GCP,Azure) and exposed as REST APIs for integration with farm management systems and dashboards

8. Federated Learning for Privacy Preservation

- Federated learning can be investigated to enable several institutions or geographic areas to jointly train disease models without providing raw image data, maintaining privacy while increasing dataset size and variety.

REFERENCES

[1] Lamba, Shweta, et al. "A novel hybrid severity prediction model for blast paddy disease using machine learning." *Sustainability* 15.2 (2023): 1502.

[2] Saminathan, K., B. Sowmiya, and M. Chithra Devi. "Multiclass Classification of Paddy Leaf Diseases Using Random Forest Classifier." *Journal of Image and Graphics* 11.2 (2023): 195-203.

[3] Jindal, Varun, et al. "Severity-Level Assessment of Groundnut Leaf Diseases: A Federated Learning and CNN Approach." *2023 4th IEEE Global Conference for Advancement in Technology (GCAT)*. IEEE, 2023.

[4] Dhar, Prashengit, Md Shohelur Rahman, and Zainal Abedin. "Classification of leaf disease using global and local features." *International Journal of Information Technology and Computer Science* 14.1 (2022): 43-57.

[5] Khalid, Munaf Mudheher, and Oguz Karan. "Deep learning for plant disease detection." *International Journal of Mathematics, Statistics, and Computer Science* 2 (2024): 75-84.

[6] Singh, Ashutosh Kumar, et al. "[Retracted] Hybrid Feature-Based Disease Detection in Plant Leaf Using Convolutional Neural Network, Bayesian Optimized SVM, and Random Forest Classifier." *Journal of Food Quality* 2022.1 (2022): 2845320.

[7] Guo, Shutuo. "Leaf Disease Detection by Convolutional Neural Network (CNN)." *Highlights in Science, Engineering and Technology* 72 (2023): 1141-1146.

[8] Hosny, Khalid M., et al. "Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern." *IEEE Access* 11 (2023): 62307-62317.

[9] Kaur, Navneet. "Plant leaf disease detection using ensemble classification and feature extraction." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.11 (2021): 2339-2352.

[10] Moupojou, Emmanuel, et al. "Segment Anything Model & Fully Convolutional Data Description for Plant Multi-disease Detection on Field Images." *IEEE Access* (2024).

[11] Azim, Muhammad Anwarul, et al. "An effective feature extraction method for rice leaf disease classification." *Telkomnika (Telecommunication Computing Electronics and Control)* 19.2 (2021): 463-470.

[12] Lv, Pengtao, et al. "An Improved Multi-Scale Feature Extraction Network for Rice Disease and Pest Recognition." *Insects* 15.11 (2024): 827.

[13] Simhadri, Chinna Gopi, et al. "Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies and techniques." *Information Processing in Agriculture* (2024).

[14] Zhou, Guoxiong, et al. "Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion." *IEEE access* 7 (2019): 143190-143206.

[15] Liang, Wan-jie, et al. "Rice blast disease recognition using a deep convolutional neural network." *Scientific reports* 9.1 (2019): 1-10.

[16] Rice Disease Classification by Combining Deep Convolutional Neural Network with Support Vector Machine

[17] Aggarwal, Shruti, et al. "Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improvise Agro-Business." *Scientific Programming* 2022.1 (2022): 1757888.

[18] Fasanmade, Adebamigbe, et al. "A fuzzy-logic approach to dynamic bayesian severity level classification of driver distraction using image recognition." *IEEE Access* 8 (2020): 95197-95207.

[19] Riaz, Saman, Ali Arshad, and Licheng Jiao. "A semi-supervised CNN with fuzzy rough C-mean for image classification." *IEEE Access* 7 (2019): 49641-49652.

[20] Bhatti, Mughair Aslam, et al. "Advanced plant disease segmentation in precision agriculture using optimal dimensionality reduction with fuzzy c-means clustering and deep learning." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2024)

[21] Kim, Tae-hoon, et al. "ANFIS Fuzzy convolutional neural network model for leaf disease detection." *Frontiers in Plant Science* 15 (2024): 1465960.

[22] Sharmila, P., et al. "FTLM: A Fuzzy TOPSIS Language Modeling Approach for Plagiarism Severity Assessment." *IEEE Access* (2024).

[23] Pajany, M., et al. "Optimal Fuzzy Deep Neural Networks based Plant Disease Detection and Classification on UAV-based Remote Sensed Data." *IEEE Access* (2024).

[24] Lin, Chun-Hui, Cheng-Jian Lin, and Shyh-Hau Wang. "Quality assessment of metal additive manufactured parts by a multiscale convolutional fuzzy neural network using ultrasound images as input data." *IEEE Access* (2023).

APPENDIX

Pseudocode:

```
# Importing necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout,
MaxPooling2D, BatchNormalization

from tensorflow.keras.optimizers import Adam

import os

import glob

import random


# Setting seed and hyperparameters

SEED = 1234

EPOCHS = 100

INIT_LR = 1e-3

BS = 32

default_image_size = (256, 256)

image_size = 0

width = 256

height = 256
```

```
depth = 3

# Paths
train_path = 'train_images/'
test_path = 'test_images/'

# Counting the number of classes
n_classes = len(glob.glob(train_path + '/*/'))
print("Number of classes:", n_classes)

# Data generators for loading images
train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)

train_generator = train_datagen.flow_from_directory(
    train_path,
    target_size=default_image_size,
    batch_size=BS,
    class_mode='categorical',
    subset='training',
    seed=SEED
)

validation_generator = train_datagen.flow_from_directory(
    train_path,
    target_size=default_image_size,
```

```
    batch_size=BS,
    class_mode='categorical',
    subset='validation',
    seed=SEED
)

def get_model():
    model = Sequential()
    inputShape = (height, width, depth)
    chanDim = -1
    print(backend.image_data_format())
    if backend.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"))

    opt = Adam(learning_rate=INIT_LR, decay=INIT_LR / EPOCHS)

    # distribution
    model.compile(loss="binary_crossentropy",
optimizer=opt,metrics=["accuracy"])
    return model

# Compiling the model
```

```
optimizer = Adam(learning_rate=INIT_LR)

model.compile(optimizer=optimizer, loss='categorical_crossentropy',
metrics=['accuracy'])


# Model Summary

model.summary()

# Training the model

history = model.fit(

    train_generator,

    validation_data=validation_generator,

    epochs=EPOCHS

)


# Plotting training history

plt.plot(history.history['accuracy'], label='Training Accuracy')

plt.plot(history.history['val_accuracy'], label='Validation Accuracy')

plt.title('Training and Validation Accuracy over Epochs')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()
```

Fuzzy.py

```
rows, cols = 200, 200

n_feature = 9

n_neurons = 100
```

```
fRules = list(product([-1.0, 0.0, 1.0], repeat=n_feature))

out_fRules = random.sample(fRules, n_neurons)

class fuzzy_inference_block(tf.keras.layers.Layer):

    # def __init__(self, output_dim, i_fmap, mu, sigma):

    #     super(fuzzy_inference_block, self).__init__()

    #     self.output_dim = output_dim

    #     self.index = i_fmap

    #     self.mu = mu

    #     self.sigma = sigma

    def __init__(self, output_dim, i_fmap, mu, sigma, **kwargs):

        super(fuzzy_inference_block, self).__init__(**kwargs)  # Pass kwargs

        self.output_dim = output_dim

        self.index = i_fmap

        self.mu = mu

        self.sigma = sigma


    def build(self, input_shape):

        self.mu_map = tf.transpose(tf.convert_to_tensor(out_fRules,
dtype=tf.float32)) * self.mu

        self.sigma_map = tf.ones((n_feature, self.output_dim)) * self.sigma


    def call(self, inputs):

        fMap = inputs[:, n_feature * self.index : n_feature * (self.index + 1)]

        aligned_x = tf.repeat(tf.expand_dims(fMap, axis=-1), self.output_dim, axis=-
1)
```

```
        aligned_c = self.mu_map

        aligned_s = self.sigma_map

        phi = tf.exp(-tf.reduce_sum(tf.square(aligned_x - aligned_c) / (2 *
tf.square(aligned_s)), axis=-2))

        return phi
```

Input:

- ImageDataset: Set of rice leaf images with labels
- num_classes: Number of disease categories
- n_femap: Number of fuzzy feature maps
- mu, sigma: Parameters for fuzzy membership functions
- batch_size, epochs: Training parameters

Output:

- DiseaseClass: Predicted class of rice disease
- SeverityLevel: Predicted severity (Mild, Moderate, Severe)

Begin

1. PreprocessDataset(ImageDataset)

    - Resize each image to 200x200 pixels
    - Normalize pixel values between 0 and 1
    - Apply data augmentation (rotation, flipping, zoom)
    - Split into Training, Validation, and Test sets

2. BuildCNNModel()

- Define input layer with shape (200, 200, 3)

- Add Conv2D and MaxPooling layers for feature extraction

- Add Dropout for regularization

- Add final Conv2D layer to output fuzzy feature maps

- Flatten the output to obtain feature vector

3. DefineFuzzyInferenceBlocks(n_femap)

   For each i in 0 to n_femap-1:

     - Extract feature slice corresponding to i-th map

     - Apply Gaussian fuzzy membership function:

       phi = exp(-((x - mu)^2) / (2 * sigma^2))

     - Output firing strength vector for fuzzy rules

4. ConcatenateAllFuzzyOutputs()

   - Merge outputs from all fuzzy inference blocks

5. AddFinalClassificationLayer()

   - Add a Dense layer with softmax activation

   - Output probability distribution across disease classes

6. CompileModel()

   - Set optimizer to Adam

   - Set loss function to categorical crossentropy

   - Track accuracy as performance metric

7. TrainModel()

   - Fit model using training data for defined epochs and batch size

   - Validate on validation set

8. PredictOnTestImage(TestImage)

- Preprocess the TestImage

- Run image through trained CNN model

- Output DiseaseClass and SeverityLevel

9. DisplayResults()

- Show predicted class and severity

- Optionally, show attention or feature maps

End