scientific reports

Check for updates

OPEN

# A hybrid deep learning and fuzzy logic framework for robust tomato disease detection and classification

Satrughan Kumar[1], Yogesh Kumar Sharma[1], Munish Kumar[1], Umesh Kumar Lilhore[2]✉, Sultan Mesfer A. Aldossary[3], Sarita Simaiya[4], Md Monish Khan[5]✉, Shimaa A. Hussien[6] & Ehab seif Ghith[7]

**Early and precise diagnosis of diseases in tomato plants is critical in ensuring productivity in agriculture and reducing losses caused by diseases. Classical classification approaches, however, are often limited by different image quality, different lighting, low resolution, and imbalanced classes. In order to confront these issues, this paper suggests a hybrid ensemble model that integrates the advantages of deep learning, fuzzy logic, and a generative model to classify diseases successfully. The suggested approach combines three strong convolutional neural networks, ResNet-50, EfficientNet-B0, and DenseNet-121, into the adaptive ensemble system. Individual models are combined to form the final decision depending on the predictive accuracy and confidence level. The use of fuzzy logic to refine intelligently the decision-making process provides more flexibility than the decisions of the static ensemble methods. A Conditional Generative Adversarial Network (C-GAN) is used to alleviate the problem of class imbalance and overfitting through the production of multiple synthetic images of high quality. This, in turn, significantly enhances the generalization of models and even gives a balanced representation of the disease's classes. The hybrid structure achieved a classification accuracy of 99.19% when tested on the PlantVillage dataset and outperformed the traditional ensemble and classical techniques. The results highlight the potential of the hybrid approach for real-world agricultural applications. This offers a scalable, accurate, and intelligent solution for automated plant disease diagnosis. This study contributes a novel, interpretable, and performance-driven model that can support sustainable agriculture through timely and precise disease management in tomato crops.**

**Keywords** Tomato Disease Classification, Hybrid Ensemble Learning, Fuzzy Logic, Conditional GAN (C-GAN), Deep Convolutional Neural Networks (CNNs), Agricultural Image Analysis

Tomatoes are one of the most extensively cultivated and commercially essential food crops, as they are rich in nutritional potential and can be utilized in multiple ways in cooking[1]. On the other hand, tomato plants are highly prone to a plethora of foliar diseases early blight (EB), late blight (LB), leaf mould (LM), spotted spider mites (SSM), septoria leaf spot (SPS), mosaic virus (MV), target spot (TS), bacterial spot (BS), yellow leaf curl (YLC), Powdery Mildew (PM), Nitrogen deficiency (ND), Magnesium deficiency (MD), Potassium deficiency (PD), and Spotted Wilt Virus (SWS), Leaf Miner (LM). These diseases inflict numerous striking patterns on the leaves, which can nearly kill the plant and significantly reduce yield. As observed, late blight is considered the most devastating disease due to the fact that in cool and damp conditions, the disease may destroy the crops completely[2]. These diseases require speedy diagnosis and management in this digital era to secure sustainable productivity and cost savings[3]. The traditional methods of categorizing diseases affecting tomato leaves are very reliant on human decision, which is subjective and can be time-consuming and inaccurate most of the time. This kind of evidence is a clear indication of the necessity to have an automated identification system that is efficient and reliable[4]. Machine learning (ML) and computer vision technologies have contributed significantly

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, A.P. 522502, India. [2]School of Computer Science and Engineering, Galgotias University, Greater Noida, UP, India. [3]Department of Computer Engineering and Information, College of Engineering in Wadi Alddawasir, Prince Sattam University, Al-Kharj, Saudi Arabia. [4]Department of Computer Application and Technology, Galgotias University, Greater Noida, UP, India. [5]Institute of R&D, Arba Minch University, Arba Minch, Ethiopia. [6]Electrical Engineering Department, Faculty of Engineering, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, 11671 Riyadh, Saudi Arabia. [7]Department of Mechatronics, Faculty of Engineering, Ain Shams University, Cairo 11566, Egypt. ✉email: umeshlilhore@gmail.com; drkumacse@gmail.com

to the creation of automated systems, but they encounter challenges. Most methods do not work with stratified data, environmental factors that affect image quality, as well as overlapping fuzzy disease symptoms[5,6]. Besides, salient features, including edges, textures, and color histograms, are computationally expensive to differentiate in high-dimensional feature spaces. In addition, it reduces their usability and effectiveness in the context of various datasets.

Transfer learning is useful in classification tasks as the models already comprehend basic image characteristics such as boundaries, textures, and shapes. With prior training on general datasets, these models quickly adapt to plant disease image classification during fine-tuning. Better performance is achieved relative to baseline models trained from scratch[7,8]. In addition, ensemble models emphasize improvements for effective systems used in classifying diseases. An individual can considerably enhance the efficiency of the whole image classification system by assembling a large number of classifiers and training them to predict. This method is more stable, as the errors of any single model are reduced, which increases overall accuracy[8]. In an ensemble network, there may be a model or models that work well in a certain situation and fail in others. An example is that one CNN might be suitable to handle the low-light conditions, whereas another might be better suited to handle the accurate processing of blurred or partially obscured images.

The combined models address these issues by incorporating outcomes to enhance reliability and attain better decisions. This, in effect, minimizes errors in image quality and incomplete symptom visibility. As a result, the system is more accurate and dependable even in the actual farming settings where images are not ideal. Ensemble methods are performed using different techniques, which include bagging, boosting, voting, and stacking. Ensemble performs well when the models with different architectures are trained on various subsets of data. This diversity helps reduce overfitting and captures more useful information from the data features. In voting-based ensembles, a simple or weighted majority vote is preferred the most. The selection and combination of CNNs within a model architecture must be done carefully because it leverages multiple advantages while compensating for the mistakes made by one model.

In agriculture, collecting large and labeled datasets is challenging. As seen, CNNs need diverse training data to perform well. They learn patterns like color, texture, and shape from the images. If the training data only shows similar types of images, the CNN may not learn enough. It may do well on known data but fail on new or different images (such as low resolution, varying lighting, and noisy conditions). GANs are frequently applied to generate new images for training purposes[9,10]. However, standard GANs produce images in an uncontrolled fashion. They do not take into account the class of the image being produced. This complicates efforts to balance certain disease classes. C-GANs create images based on the class labels, such as "early blight" or "bacterial spot". This allows for more effective creation of images in the case of rare disease categories. It addresses the issue of imbalance in classes. C-GANs are also capable of generating images of varying lighting, noise, and resolutions. This increases the diversity of the dataset. Such diversified data is better learned by the CNN models. Consequently, they are effective even with low quality or odd.

In spite of the fact that these single models do a commendable job in feature recognition, fuzzy logic deals with uncertainty. Fuzzy logic helps manage uncertainty and noise. It improves decision-making, accuracy, and interpretability. It makes the hybrid model more robust for unclear or overlapping disease features[6,7]. The hybrid methodology increases their effectiveness in managing different disease symptoms[11]. They adjust well to changes in the lighting, weather, and types of crops that make them applicable in the field[12]. These systems can be applied in a wide variety of agricultural settings since they are applicable to various types and species of diseases. For instance, in[13] authors applied the Sugeno fuzzy integral to fuse CNN models for cervical cytology images, improving decision reliability through weighted fusion. Similarly, fuzzy ensemble modeling is used for breast cancer detection and has been found to help manage uncertainty and boost diagnostic performance[11,14]. In agriculture, an adaptive ensemble with exponential moving average fusion was developed for classifying diseases in tomato leaves with high accuracy[15]. The deep neuro-fuzzy network showed not only good but also high accuracy in handling vague or imprecise visual features[16]. In[17], the authors used a neural network-based model for identifying early and late blight, but did not include any fuzzy reasoning or ensemble strategies.

Fuzzy ensemble techniques applied to medical image classification tend to come from agricultural environments that are more controlled than adaptable to the real world[13,14]. These approaches may not be suitable for inconsistent lighting, diverse image quality, and other highly variable conditions. Likewise, adaptive ensemble models for fuzzy logic uncertainty plant modeling have been explored but not fully embraced due to fuzzy logic uncertainty reasoning capabilities[15]. Moreover, plant disease detection using deep neuro-fuzzy networks could apply fixed fuzzy structures that do not have the dynamic adjustment of weights per instance needed for fluctuating input quality[16]. Single neural network models lack robustness, collective intelligence, and other qualities unique to ensembles[17]. Additionally, manually defined fuzzy measures[13,14] introduce subjectivity and require domain expertise, making these approaches less scalable. Previous works used fuzzy with deep learning for better accuracy, but had high complexity and fixed fuzzy rules. Earlier models struggled with scalability and required expert tuning for fuzzy parameters. My model reduces manual tuning and enhances robustness under uncertain or new crop conditions.

The suggested framework enhances strength using three combined mechanisms. Initially, the data augmentation module introduced is the C-GAN, which creates a variety of synthetic samples that model real-world variations. Second, the fuzzy inference system is used to reduce the effects of uncertain or inconsistent predictions made by the ensemble decision-making. Third, the combination of three deep learning models, namely ResNet-50, EfficientNet-B0, and DenseNet-121, is dynamic to enable multi-level and extensive feature extraction[19–21]. In the proposed hybrid model, fuzzy logic adjusts the contribution of each CNN model based on classification accuracy and a per-sample confidence score derived from SoftMax[18]. In this setup, ResNet-50 is concerned with deep spatial features learning, EfficientNet-B0 introduces scale-adaptive features, and DenseNet-121 encourages feature reuse to improve generalization. These modules together allow the correct

and reliable classification of diseases even in problematic imaging conditions. Our contributions to the hybrid framework are as follows:

- *Hybrid Classification Framework Development:* A deep learning hybrid model was developed to detect tomato leaf disease. The model is successful in dealing with issues like the imbalance in classes, image quality variation, and overfitting, which makes the learning stable in various samples.
- *Combination of Fuzzy Logic and Adaptive Ensemble Learning:* An adaptive fuzzy logic mechanism was used to integrate the predictions of ResNet-50, EfficientNet-B0, and DenseNet-121. The ensemble weights are optimized during training using accuracy and confidence measurements, which allow dynamic adjustment and enhance the overall robustness.
- *Data Augmentation Using Conditional GANs:* C-GANs were utilized to artificially increase the training data by producing natural tomato leaf images that reproduce real-world variations and environmental distortions.
- *Benchmark Performance:* The proposed framework was able to achieve an accuracy of 99.19 and a recall of 99.2 using the PlantVillage dataset, which is higher than the results of similar state-of-the-art methods.
- *Comparative Analysis:* A critical comparative analysis was made with a number of the existing methods, which proved that the suggested framework is better in terms of performance, flexibility, and reliability in the diagnosis of tomato plant diseases through the automated method. We have also employed it on different diverse datasets and observed a notable accuracy.

This paper is arranged as follows: Section "Literature survey" presents a survey of the existing literature on leaf disease classification methods. Section "Materials and methods" illustrates the proffered ensemble architecture based on a new approach to weighting and integration of the models. Section "Experimental results" presents the investigational outcomes and analyzes the efficiency of the proposed approach. Section "Conclusion and future directions" concludes the findings and synthesizes the research contributions on tomato leaf disease classification.

## Literature survey

This part examines the necessary work done by various researchers in tomato classification. In subsection "Traditional and intelligent learning approaches in crop disease detection", we cover the classical and contemporary techniques pertaining to crop disease diagnostics. Subsection "Design evaluation and strategic enhancements in the proposed disease classification system" discusses the gaps identified in existing systems. It presents the new hybrid model incorporating convolutional neural networks, fuzzy logic, and generative adversarial networks with enhanced precision, adaptability, and dependability for tomato disease detection.

### Traditional and intelligent learning approaches in crop disease detection

Two pre-trained models, namely Inception V3 and Inception ResNet V2, were applied for a classification task[22]. Their experiments on different dropout rates between 5 and 50% showed that Inception V3 performed best at 50% dropout, while Inception ResNet V2 did well with 15% dropout. They achieved an impressive accuracy with a low loss of 0.03. In a similar approach, a lightweight pipeline incorporates three enhanced CNNs employing transfer learning and hybrid feature selection. The kNN and SVM achieved the highest accuracy of 22–24 selected feature range[23]. Authors in[24] designed a CNN that features four convolutional layers, max-pooling, dropout, and Softmax output. It achieved an accuracy of 96%, which is significantly higher than the results yielded by AdaBoost (52%) and Random Forest (71%). In[25], a custom CNN was able to classify ten varieties of tomato leaf disease and achieved an accuracy of 96%.

Study in[26] used color and edge histograms to extract shape, color, and texture features. K-means segmentation isolated diseased regions in images. In[27], a custom AlexNet-style CNN gave strong results after full preprocessing. Its performance was limited by the absence of feature fusion and uncertainty control. Study[28] fused SVM with Random Forest using hand-crafted color, shape, and texture features and achieved 95.58% accuracy. In[29], lightweight CNNs with transfer learning from VGG16 and VGG19 classified eleven categories with 95% precision and recall. Authors in[30] evaluated CNN, ResNet50, and MobileNet on six diseases and healthy leaves. ResNet50 achieved 98.7% accuracy. The study showed that deeper models performed better than shallow or non-pretrained CNNs.

The authors applied deep neural networks with genetic algorithms in metaheuristic optimization and achieved 98.8% classification accuracy on tomato disease[32]. In[33], the authors employed segmentation to reduce overfitting by focusing on infected regions. ResNet50 achieved 98.4% accuracy when compared to VGG16, ResNet50, and InceptionV3 with image segmentation. In[34], a standard convolutional neural network (CNN) was enhanced by a hybrid preprocessing, adaptive segmentation, and classification module to boost the precision of lesion localization and spatial mapping in precision agriculture. The integrated CBAM with YOLOv6 and BiRepGFPN for multi-scale fusion achieved 92.9% precision, 95.2% recall, 94.0% F1-score, and 93.8% mAP on the PlantDoc and tomato datasets, which exceeds the baseline by 12% mAP[35]. The authors in[36] integrate a capsule network to preserve the spatial hierarchy of features, while the attention mechanism helps the model focus more effectively on the relevant disease regions. Researchers started implementing GANs to create synthetic disease images in order to rectify data inadequacies[36–39].

In[31], the authors introduced a hybrid model that comprises ResNet50 with Intuitionistic Fuzzy Random Vector Functional Link Classifier (IFRVFLC) for plant leaf detection. ResNet50 extracted deep visual features. However, the IFRVFLC is employed to manage uncertainty in classification. The model achieved an accuracy of 94.8% and improved performance against noisy inputs. However, its hybrid design increased training complexity and demanded expert tuning. Simplifying the fuzzy architecture could enhance scalability. The authors proposed a deep neuro-fuzzy network for tomato leaf disease detection. The model divided input images into patches[16].

These patches are processed through fuzzy inference and pooling layers using TSK rules. The previous model was able to model the accurate and ambiguous features of the disease with greater interpretability. It achieved an accuracy of 94.19% in eight tomato leaf disease classes. Nevertheless, it was not as efficient or adaptable because of high computation requirements and the use of manually defined fuzzy sets. In a study, authors used a Sugeno fuzzy integral-based ensemble to classify cervical cytology using the results of InceptionV3, DenseNet161, and ResNet34[13]. Their approach employed the fuzzy decision scoring technique in which adaptive weights were obtained based on the prediction confidence, and constant fuzzy densities were obtained using the accuracy values. This was better than simple averaging, but was not very flexible since parameter tuning was not optimized dynamically.

In the same manner, authors suggested a Choquet fuzzy integral-based ensemble of five deep learning models to identify tomato leaf disease[5]. The contributions of each model were measured in terms of fuzzy measures in this work based on the analysis of the Shapley value. The fuzzy measures remained static after training, reducing adaptability in uncertain conditions.

Authors in[6] proposed a fuzzy aggregation-based deep ensemble using five CNNs: VGG16, InceptionV3, ResNet152V2, DenseNet201, and MobileNetV2 for classification[6]. The model used a fuzzy Max–Sum integral instead of the MIN operator for smoother fusion. It achieved superior accuracy over single models. However, adaptation under new conditions or class imbalance was limited due to fuzzy measures, which were static and set manually.

In[11], the authors developed a fuzzy logic–CNN framework for multi-crop leaf disease classification 11. The fuzzy module carried out image enhancement and segmentation to visualize the features in the region of interest. However, the AlexNet and GoogleNet were used to perform feature extraction. The fuzzy logic improved image quality, but used fixed rules and membership functions. The static fuzzy component reduced adaptability to unseen crop conditions. In[44], the authors proposed an Integrated Fuzzy and Deep Learning Model (IFDM) for automated coconut maturity detection. An object segmentation approach based on Mask R-CNN was used to segment coconut maturity, and a Combined Fuzzy System (CFS) was employed to classify. In this context, the fuzzy inference unit examined the main features of color, texture, and geometric shape based on the rule-based evaluation. A Probability-Based Fuzzy Integration (PFI) mechanism and a Decision-Making Fuzzy (DMF) module were used to obtain the final decision and allow maturity grading in real time with an accuracy of 86.3%. Nevertheless, the model did not perform well in different illumination and background scenarios because the fuzzy rule base was fixed, which restricted the ability to adapt to new visual variations.

In a different research, authors suggested a fuzzy contrast-enhanced convolutional neural network to identify potato blight[45]. Before training, the contrast adjustment was done using fuzzy logic to enhance the visibility of infected regions. Several CNNs that are trained with different optimization methods were then fused together through a Bayesian-based Learned Optimized Weighted Fusion scheme. The final ensemble had an accuracy of 97.94% which is higher than that of individual CNNs. However, the framework was computationally expensive, and it took a lot of fine-tuning to achieve optimal performance.

The recent advancements in Vision Transformers (ViTs) have demonstrated a high potential for plant disease detection by partitioning images into small patches to perform analysis of features in parallel. Their application is, however, limited by the requirement of large datasets and high-performance computing[40,41]. The Swin Transformer architecture was able to overcome this weakness by using a shifted window method that effectively captures both local and global dependencies at a low cost and residual connections to enhance local feature extraction, resulting in increased recognition accuracy in plant disease datasets.

In addition, an experiment in[43] introduced a light YOLOv8s network to detect tomato leaf disease quickly. The model had an average Precision (mAP) of 92.5% on the PlantVillage dataset, and operated at about 121.5 FPS, 3 times faster and with a higher average Precision (mAP) than both YOLOv5 and Faster R-CNN.

## Design evaluation and strategic enhancements in the proposed disease classification system

The effectiveness of classification systems is determined by several critical factors. The amount and quality of training data serve as the foundational building blocks. Table 1 summarizes plant disease classification models by comparing their accuracy, methods, limitations, and fusion techniques. As demonstrated, noise and blur weaken models, while samples that are in short supply yield insufficient model performance. Biased models often stem from imbalanced classes. Also, the chosen architecture of a given model holds considerable importance relative to a CNN's capacity to learn discriminative features. Moreover, challenges such as overfitting or undergoing a domain shift tend to impact model performance in unpredictable conditions greatly. Applying transfer learning (TL) with InceptionV3 or InceptionResNetV2 allows for great TL results coupled with high accuracy when dropout settings are adjusted optimally[21,22]. Balanced and clean datasets yield the best results from these models. Lightweight pipelines using compact CNNs and feature selection reduce model size and improve speed[23]. However, such smaller architectures tend to perform poorly on extremely complex or detailed problems. On the other hand, deeper CNNs, such as VGG16 or more recent massive architectures, have the ability to learn a lot[29]. These networks still may encounter the vanishing gradient problem. Some methods use only 22–24 features to classify diseases.

However, many of these rely on single CNNs. They lack flexibility when image quality varies. Several custom CNN architectures also reached over 96% accuracy. Custom-designed CNN frameworks achieved success with over 96% accurate results. These models, in most cases, rely on a single CNN architecture or use predetermined fusion strategies. This may restrict adaptability to different imaging conditions[25,26]. Additionally, if not configured properly, overfitting can still occur while fine-tuning a large pre-trained model with a small and specific dataset. In plant disease detection, multiple CNNs integrated into an ensemble model are essential for robust and accurate decision-making. Their deep learning complexities and variability are crucial in providing a valuable understanding through real agricultural images. By drawing from different types of CNN strengths, the

| No | Model (Ref) | $A_c$ (%) | Core Mechanism | Design Constraints | Data Diversity Strategy | Fusion/Fuzzy |
|---|---|---|---|---|---|---|
| 1 | Mezenner, Amine, et al. (2025)[6] | 98.17% | Fuzzy Aggregation-Based Deep Ensemble | Requires fuzzy membership tuning, High computation | Limited Dataset Diversity | Fuzzy aggregation fusion using membership weighting |
| 2 | Deng et al. (2025)7] | 97.9% | FCMNet (Multimodal Fusion) | No uncertainty handling, | Standard Processing | Fuzzy C-means–based multimodal fusion |
| 3 | Mishra et al., 2025[31] | 94.8% | Hybrid CNN + intuitionistic fuzzy random vector functional link classifier | Complex parameter optimization | Multi-crop leaf dataset under varied conditions | Intuitionistic fuzzy feature enhancement |
| 4 | Ghous et al., (2024)[11] | 98.6–99% | Pre-trained CNN feature extraction + fuzzy decision layer | Limited explainability in the fuzzy rule base | Standard Processing | Simple fuzzy logic-based aggregation |
| 5 | Hang et al., (2024)[5] | 97–99.8% | Deep ensemble of CNNs with Choquet fuzzy integral fusion | Overfitting risks. Verified only on 4 classes | Standard Processing only to 4 classes | Choquet fuzzy integral–based score fusion |
| 6 | Javidan et al. (2024)[28] | 95.6% | Weighted ensemble + features | Heavy due to the handcrafted focus | Relief feature selection | Ensemble, no fuzzy |
| 7 | Tian et al. (2022)[16] | 94.4% | CNN + fuzzy inference | Fixed fuzzy, single CNN | Standard images | Fuzzy integration |
| 8 | Zhang & Ren (2024)[35] | 97.2% | Transformer + federated learning | High compute, data needs | Patch-based training | No fusion/fuzzy |
| 9 | Pandiyaraju et al. (2024)[15] | 99% | EMA fusion + weighted-gradient | Large ensemble, complex fusion | Augmented + multi-sourced | No fuzzy |
| 10 | Khan et al. (2024)[17] | 97,4% on the PlantVillage dataset | Inception V3 | Overfitting risks. Verified only on three classes | Preprocessing, No augmentation | No fusion/fuzzy |
| 11 | Paul et al. (2023)[29] | 95% | Real-time TL design | Smaller feature space | Data augmentation & resizing | No fuzzy |
| 12 | Abbas et al. (2021)[38] | 97% | C GAN + transfer learning | No ensemble | C GAN augmentation | No adaptive fusion |

**Table 1**. Comparative Analysis of Existing Plant Disease Classification Models: Accuracy, Mechanisms, Limitations, and Fusion Strategies.

ensemble provides better generalization and robustness. Architectures such as ResNet-50, EfficientNet-B0, and DenseNet-121 offer deep, scalable, and reusable features for efficient, accurate tomato disease classification. Each CNN covers gaps left by others in their vision-restricted classifications, which allows the combined approach model to be consistently dependable in visually volatile environments. Therefore, ensemble models enable exact classification and more advanced diagnostic results across diverse agricultural settings by capturing numerous patterns of tomato leaf diseases. Traditional methods of image processing, including color histograms as well as k-means segmentation, assist in the identification and isolation of disease regions[26].

These techniques are effective for shallow models. Nevertheless, their lack of ability to extract deep features diminishes their effectiveness in more complicated scenarios. Solutions provided by ensemble models with SVM and Random Forest classifiers do perform reasonably well. However, these models often rely heavily on hand-crafted features, which tend to provide very low accuracy when the symptoms of a disease are subtle or when the images are of low quality. Some works use GANs for richer data but do not combine them with model fusion. C-GANs have the ability to simulate changes in lighting, add noise, and alter resolution. However, many existing methods using C-GANs focus only on generating data. They do not combine it with adaptive learning systems[10,28]. The effectiveness of fuzzy ensemble methods in medical imaging has been documented[13,14]. Still, these approaches may not be applicable on the farm where illumination and image capture parameters fluctuate. Some models focused on plant disease diagnosis utilize adaptive ensembles but do not seem to adequately employ fuzzy logic that would resolve uncertainty underscoring prediction[15].

Deep neuro-fuzzy models, as outlined by[16], tend to utilize static fuzzy rules, which result in changes in inputs. Moreover, single-model CNN approaches do not possess the flexibility and resilience of ensemble methods. Many studies have improved feature learning using attention and residual connections[17]. For instance, models with attention modules focus better on disease areas. Swin Transformer captures both local and global features but needs high computational power[41]. Only a few models include fuzzy logic. A few use ensemble voting, yet they do not adapt model weights dynamically during prediction. Real-time confidence or uncertainty is not considered in fusion strategies. Unlike earlier fuzzy ensemble models in agriculture and medical imaging, which used static TSK-type fuzzy rules within a single CNN. Some employed non-adaptive Choquet fuzzy integration, and others relied on fixed fuzzy measures in Max–Sum aggregation. This shows a major gap in adaptive, intelligent fusion under uncertain conditions[5–7].

Previous works used fuzzy with deep learning for better accuracy, but had high complexity, fixed fuzzy rules, and limited adaptability. Earlier models also struggled with scalability and required expert tuning for fuzzy parameters. The proposed framework unifies C-GAN–based augmentation, multi-CNN feature fusion, and a dual-metric dynamic fuzzy inference system for tomato leaf disease classification. Unlike previous fuzzy ensembles in agriculture and medical imaging that use fixed or confidence-based weights, the proposed fuzzy rule base adaptively computes model weights using both accuracy and confidence through T-norm reasoning. This development allows instantaneous adjustments to low-confidence predictions or conflicting predictions and protects against overfitting during generalization over severe, varying conditions, as well as disproportionate class distribution, and other constraints. Thus, the combination of C-GAN and fuzzy dynamic fusion represents a bound improvement over the older static or single-metric fuzzy ensemble methods.

## Materials and methods

This section describes the proposed methodology for the accurate identification of tomato illnesses. To increase the strength of the proffered classifier as well as the validity of the classification, the proposed method uses three pre-trained CNN models, which are ResNet50, DenseNet121, and EfficientNetB0. As seen, the classification problem is sensitive to class imbalance, changing light conditions, low image quality, and shape changes of the leaves. To alleviate these issues, preprocessing techniques are introduced, and the color and contrast are adjusted to reveal the symptoms, which in turn help provide an accurate diagnosis. We also use data augmentation to feed our model with pictures to solve the generalization issue. It can generalize to complex images. Further, while ensembling, we apply fuzzy rule-based weight aggregation, which allows such individual models to work cohesively. The subsequent subsection explains the implementation of the proposed work.

### Data pre-processing

As seen in Fig. 1, the tomato dataset comprises images of unhealthy tomatoes taken under multiple backgrounds, lighting, and camera settings. Adjusting color corrections and contrast helps in improving the clarity of disease symptoms in images. Preprocessing ensures the model focuses on the leaf region, not background clutter.

We employed preprocessing to enhance the quality of the images. Gamma correction enhances visibility in dark areas by increasing the intensity of the mid-tones, whereas adaptive gamma correction changes brightness dynamically depending on the characteristics of the scenery. The color enhancement enhances chromatic balance, which helps in detecting discoloration of the disease, like yellowing or brown on the surface of the leaves. All of these steps of enhancement enhance the uniformity and sharpness of the image, which allows the model to better reflect disease-specific features in this study. Adaptive Gamma Correction, Histogram Equalization (HE), and color enhancement techniques are used. Adaptive methods are favored over a fixed gamma value since they cannot effectively correct all images with varying brightness. Histogram equalization is also done in the HSI color space, with the saturation component (S) multiplied by a factor of 1.2 to 1.5. Once improved, the transformed HSI image is again converted back to the RGB color space to yield the final visually enhanced image that can be used in feature extraction. This is obtained as follows:

$$\mathrm{I}_1(\mathrm{x},\mathrm{y}) = \mathrm{T}\left(\mathrm{H}(\mathrm{x},\mathrm{y}), \alpha \cdot \mathrm{S}(\mathrm{x},\mathrm{y}), \mathrm{I}(\mathrm{x},\mathrm{y})\right) \tag{1}$$

In Eq. (1), 'T' employs a transformation from HSI to RGB.

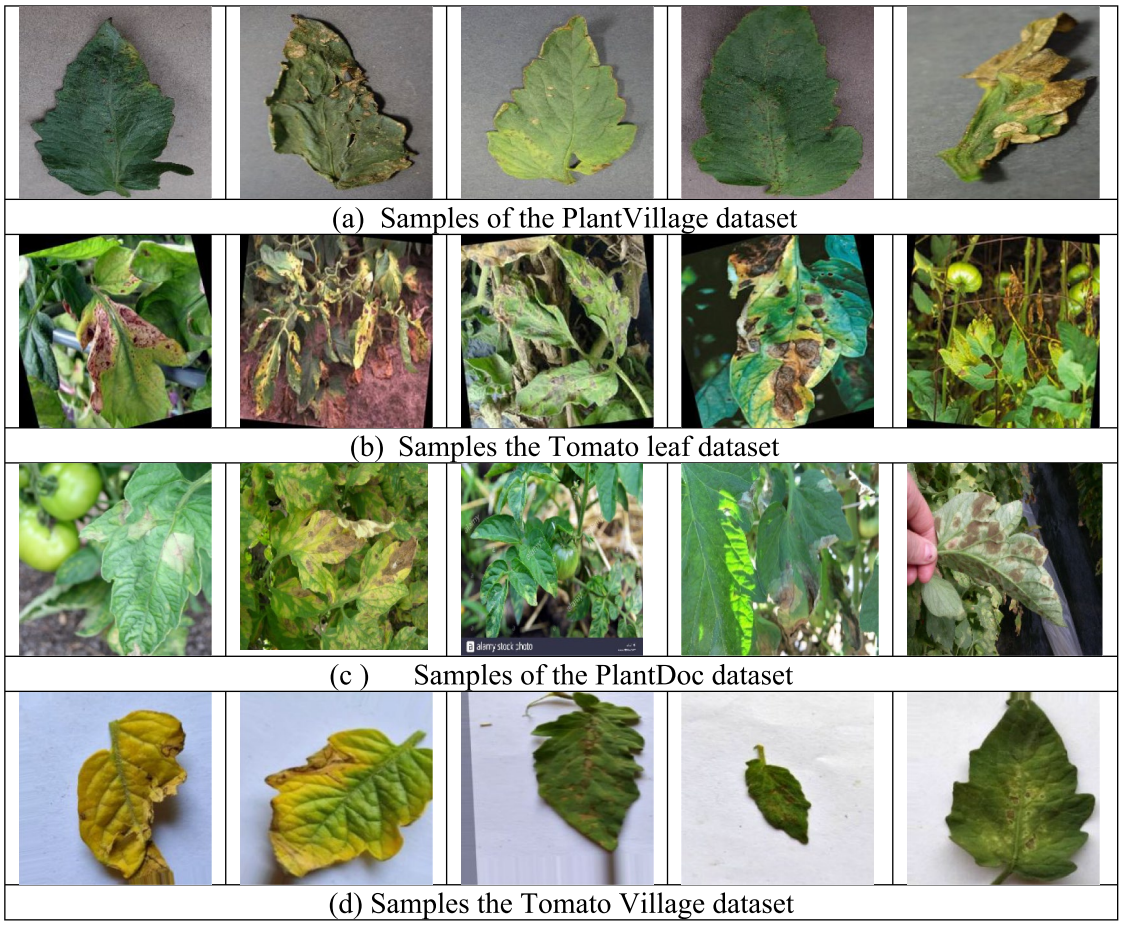



**Fig. 1**. Tomato leaf dataset with unhealthy images.

In the final output '$I_{final}(x,y)$', the intensity produced after applying the gamma correction, histogram equalization, and color enhancement techniques is combined and labeled as the output of the final enhancement step. This output is the result of applying all enhancement functions and is expressed as follows:

$$I_{final}(x,y) = \lambda_1 \cdot c \cdot I_{in}(x,y)^{\gamma} + \lambda_2 \cdot \left( (I_{max} - I_{min}) \cdot \sum_{j=0}^{L} p(j) + I_{min} \right) + \lambda_3 . I_1 \quad (2)$$

In Eq. (2), $\gamma = \log(M)/\log(128)$ and $c = 255/\max(I_{in})$. The normalization constant 'c' ensures the output stays within [0, 255]. $I_{in}(x, y)$ is the input pixel intensity at position (x,y). In Eq. (2), '$I_{max}$' and '$I_{min}$' are the maximum and minimum pixel intensities in the image. The term 'p(j)' represents the probability density function of the pixel intensities, and 'L' is the number of intensity levels. $\lambda_1$, $\lambda_2$, $\lambda_3$ are weights that control the contribution of each method, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. We set empirically 0.3, 0.3, and 0.4 for $\lambda 1$, $\lambda_2$ and $\lambda 3$ respectively for this experiment.

This configuration enhances colors, making it ideal for disease detection and classification in plants where colors change significantly.

## Image data augmentation technique

In general, an increase in samples will improve learning, but the samples must be diverse. Diversity ensures that features irrelevant to the disease are not learned, while cues that are dependent on the disease are learned. In terms of specific features such as shape, size, texture, background, and symptom intensity, diversity is very important. It is found that augmentation methods that improve diversity contribute positively towards developing a more robust and generalizable classifier. In datasets such as PlantVillage, there exists some class imbalance where certain diseases have excess samples while others are undersampled. The model becomes biased towards learning from majority samples, resulting in poor performance on minority ones.

These kinds of underrepresented images can be generated using C-GAN algorithms, which balance the dataset and improve recall alongside F1-score exponentially for those minority classes. We employ extensive image data augmentation approaches to overcome the obstacles caused by class imbalance and unpredictability in characteristics such as illumination, image quality, and leaf shape. These techniques are requisite for increasing the model's generalization capacity across diverse environments. We used both traditional augmentation and augmentation based on C-GAN. Variety can be improved using older techniques such as rotation, flipping, and brightness adjustment. They are quick and straightforward to implement. The addition of C-GAN produces advanced diversity. This combination allows the model to be better trained for practical scenarios involving bad lighting or concealed portions of leaves, as well as blurred imagery. This makes the model stronger, more balanced, and more accurate for real agricultural use.

*Conventional augmentation methods*

Table 2 shows the types and settings of conventional image augmentation techniques used to improve the robustness of the tomato leaf disease classification model. Standard image augmentations prepare deep-learning models to meet common challenges found in real-world photographs of tomato leaf diseases.

In this work, we augment our training set by applying rotation from −20 to 20, flipping horizontal and vertical (0.5), zoom equal to 0.2, and height shift equal to 0.2. This process ensures that models are robust to various orientations, scales, and illuminance. These increase the training data's variability, exposing the model to a wide range of variances and improving its generalization capability[25]. Horizontal and vertical flips train the model to identify a leaf regardless of its orientation. Controlled rotation and shear distortions mimic the appearance of leaves captured at an oblique angle. Width and height translations accustom the system to leaves that are partially visible because of obstruction. Brightness and contrast variations build resilience to images taken in bright sunlight or deep shade. Zoom-in crops combined with Gaussian blur address the problem of low-resolution images that have been aggressively magnified. When applied together, these strategies strengthen prediction accuracy across diverse lighting, occlusion, and resolution scenarios.

*Augmentation using GANs*

In order to augment this dataset, C-GANs, a specific type of GAN, are used[46,47]. Certain classes of diseases pose a problem because they have significantly fewer images than others. To resolve this issue, C-GANs produce synthetic samples for these weaker classes to help balance the dataset and avoid bias towards the majority class.

| Data Augmentation | Range |
|---|---|
| Image Horizontal/Vertical Flip | True |
| Image Rotation | ± 20° |
| Image Width/Height Shift | 0.2 |
| Image Zoom/Shear Range | 0.2 |
| Brightness Range | (0.8, 1.2) |
| Contrast Adjustment | (0.8, 1.2) |
| Gaussian Blur | Kernel size 3 × 3 |

**Table 2**. Conventional Image Augmentation Settings.

While traditional augmentation techniques like flipping and rotation only modify existing samples, C-GANs create wholly new, unique samples. This technique increases the variability within a dataset, improving the model's generalized performance on unseen data. Combining C-GAN image synthesis algorithms with more traditional augmentation methods helps to improve both precision and generalization of classification, making the model more effective in practical applications for tomato leaf disease detection.

The most commonly known GAN is formed by training two neural networks: a generator (G'e) and a discriminator (D'i), alternately using the minimax approach in the training process. The generator then produces realistic synthetic data, and the discriminator's task is to decide whether the data generated is real or fake. The objective function of a conventional GAN is given as:

$$\mathrm{F}\left(\mathrm{D}\prime_{\mathrm{i}}, \mathrm{G}\prime_{\mathrm{e}}\right) = \mathcal{E}_{\mathrm{x_i}\sim \mathrm{pdata(x_i)}}\left[\log\left(\mathrm{D}\prime_{\mathrm{i}}\left(\mathrm{x_i}\right)\right] + \mathcal{E}_{\mathrm{z_i}\sim \mathrm{p_z(z_i)}}[\log\left(1-\mathrm{D}\prime_{\mathrm{i}}\left(G\prime_{e}\left(z_i\right)\right)\right)\right] \tag{3}$$

In Eq. (3), $F(D\prime i, G\prime e)$ denotes a minimax game $(MING\prime e, MAXD\prime i)$. In this case, the generator and discriminator are competing with each other in this minmax approach. The discriminator aims to achieve the maximum classification of accurate data from fake data. Conversely, the generator seeks to minimize this capacity by creating highly realistic synthetic data. In Eq. (1), the discriminator's output for genuine data x is $\prime D\prime i(xi)\prime$ and the generator's output for input noise $\prime zi\prime is\prime G\prime e(zi)\prime$.'Ɛ' signifies the expected value. The variables pdata $(\mathrm{x_i})$ and $\prime pz(zi)\prime$ represent the data distribution and noise, respectively. The proposed work entails implementing the C-GAN architectures outlined in[47] for both $\prime G\prime e\prime and\prime D\prime i\prime$. In a conditional GAN, both $\prime G\prime e\prime and\prime D\prime i\prime$ are dependent on the additional information '$\mathrm{y_i}$'. This conditioning enables the model to create data that meets the specified criteria '$\mathrm{y_i}$'. The C-GAN defines its objective functions as follows:

$$F_1\left(D\prime_i, G\prime_e\right) = \varepsilon_{x_i\sim pdata(x_i/y_i)}\log\left(D\prime_i\left(\frac{x_i}{y_i}\right)\right) + \varepsilon_{z_i\sim p_z(z_i)}\left[\log\left(1-D\prime_i(G\prime_e\left(\frac{z_i}{y_i}\right))\right)\right] \tag{4}$$

In Eq. (4), the variables $\prime xi\prime, \prime yi\prime$, and '$\mathrm{z_i}$' represent real data samples, conditional information or labels, and noise vector inputs to the generator, respectively. The term $\prime pdata\left(xi|yi\right)\prime$ denotes the real data distribution conditioned on $\prime yi\prime.\prime D\prime i(xi \mid yi)\prime$ represents the conditional probability of the discriminator. It refers to '$\mathrm{x_i}$' as an actual data point, given the condition. $\prime yi\prime$. On the other hand, $\prime G\prime e(zi \mid yi)\prime$ represents the synthetic data generated by the generator, given the noise '$\mathrm{z_i}$' and condition '$\mathrm{y_i}$'[48]..

As shown in Fig. 2, the initial layer of the generator concatenates a noise vector with conditional information. In this experiment, we treat the conditional information as class labels. This input is subsequently subjected to a series of fully connected layers, which are followed by reshaping and upsampling layers to produce a coherent image representation. The last layer, which often uses 'tanh' activation, produces the synthesized image. However, the discriminator combines the relevant conditional information at the input to create either a real or fake image. We use some convolutional layers that employ ReLU activations to evaluate the input image and extract features. The output layer generally generates a single probability value to determine the authenticity of the input image. This configuration is the foundation for the proposed study by improving synthetic image generation and categorization. Figure 3 shows the samples of synthetic images using C-GAN.

The Fréchet Inception Distance (FID) is an important indicator to assess the efficacy of images produced by GANs and other generative models. FID evaluates real and generated images by estimating their distributions in the feature space of a pretrained Inception v3 network. It quantifies how close the synthetic images are to the

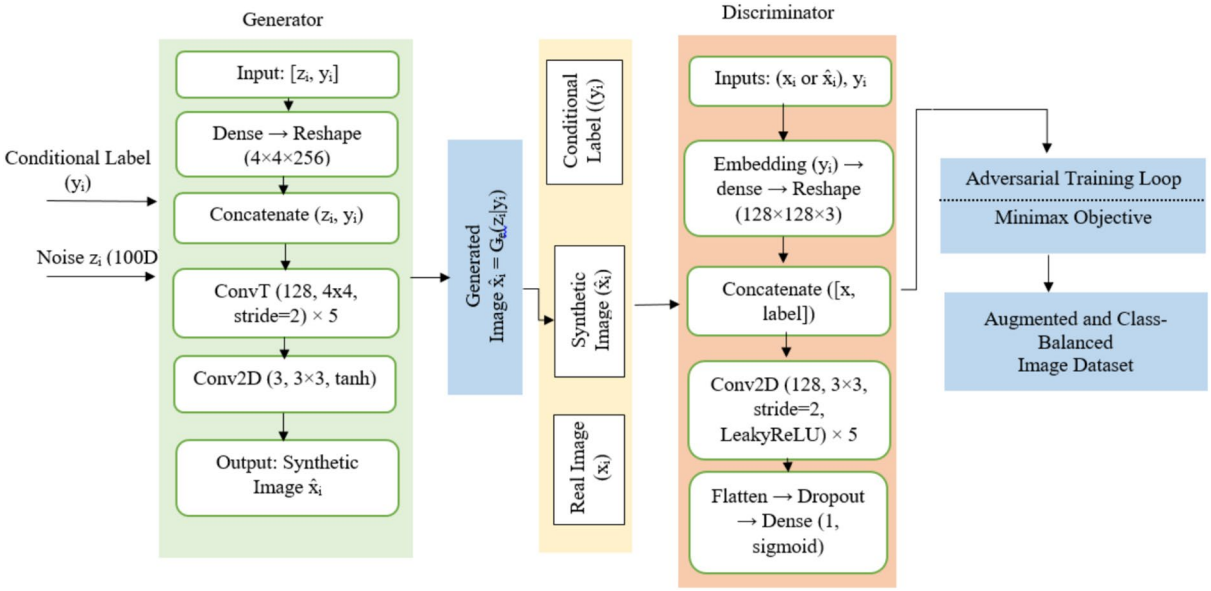



**Fig. 2**. C-GAN Framework Comprising Generator and Discriminator for Synthetic Sample Generation.

real ones in terms of high-level visual features. Prior studies have confirmed that moderately realistic GAN-generated images can significantly boost classification performance, even with FID scores above the ideal threshold. Authors demonstrated the usefulness of DCGAN and CGAN-based augmentation for improving the accuracy of tomato disease classification. They did not mention how close the synthetic images are to the real ones[10,38].

In[37], the authors reported FIDs for their synthetic samples for tomato leaves ranging from 69 to 245. In the same way, authors in[5] achieved an FID of 67 with a Progressive WGAN-GP and noted concomitant improvements in convolutional-network performance on a small rice-disease dataset. Taken together, these results indicate that even with moderate FID values, the diversity of features and correction of class imbalance are enhanced. These findings support the use of GANs in our work, where synthetic images preserved essential disease features such as lesion shape, color variation, and texture disruption. In plant disease detection, key features such as color gradients, lesion shapes, and texture variations are more important than perfect visual detail. We obtained a FID score of 69. The generated images in our study successfully preserved these characteristics. Furthermore, the empirical improvement confirms that the generated images are practically valuable and effective for training robust classification models.

## Proposed hybrid ensemble architecture

The need to distinguish tomato leaf diseases is still a critical and tricky issue because of the visual similarity of the lesions, variation in the distribution of classes, and changes in illumination at the time of taking the image. To overcome these complexities, a hybrid ensemble architecture is proposed that combines three pre-trained convolutional neural networks, ResNet-50, EfficientNet-B0, and DenseNet-121. The models have different strengths in terms of representation: ResNet-50 focuses on hierarchical spatial features, EfficientNet-B0 offers scale-aware representations, and DenseNet-121 allows using features again to achieve better generalization. Their joint use can be used to overcome the drawbacks of single models that are usually unable to identify overlapping or fine-grained patterns of disease due to limited receptive fields.

The ensemble combines multi-scale feature information, which increases the discriminative power of the classification process. Figure 4 shows that the overall workflow of the work is divided into two large modules. The initial module, data preprocessing and augmentation, uses standard procedures like rotation, flipping, and brightness control, as well as artificial images created by C-GAN, to address the issue of class imbalance and minimize the impact of noise. The second module-hybrid ensemble learning, combines the scores of the three CNNs in terms of their prediction with an adaptive fuzzy rule-based weighting strategy. This fuzzy system is the dynamically adjusted contribution of each model to the final decision depending on its confidence and accuracy, which guarantees balanced decision-making and enhanced adaptability to various imaging conditions.

With fuzzy logic weighting and data generation, the model demonstrates improved performance on noisy, imbalanced, and heterogeneous datasets, making it a reliable solution for advanced agricultural disease diagnosis. The SoftMax layer of CNNs obtains the confidence score, which represents the probability of each class. It shows how specific the model is in terms of its prediction. The accuracy metric indicates the proportion of correct predictions made by the model during testing or validation. However, accuracy alone does not convey the model's confidence when making those predictions.

Ensemble strategies evaluate both confidence and accuracy in deciding how to merge the outputs of respective models. With traditional approaches, the confidence score is useful as a prediction weight; however, accuracy acts passively by reflecting performance without influencing the combination of predictions[13,18]. Our proposed ensemble architecture comprises three pretrained convolutional models: ResNet50, DenseNet121, and EfficientNetB0. These models have been successfully applied to large image datasets and trained for image classification tasks, and form the backbone of our ensemble model. In this paper, we outline the aforementioned pre-trained models as well as describe their architecture, including hyperparameter values, fine-tuning steps undertaken, and advantages realized from using an ensemble approach with these pre-trained networks.

**Fig. 3**. Sample of Synthetic Images using C-GAN.

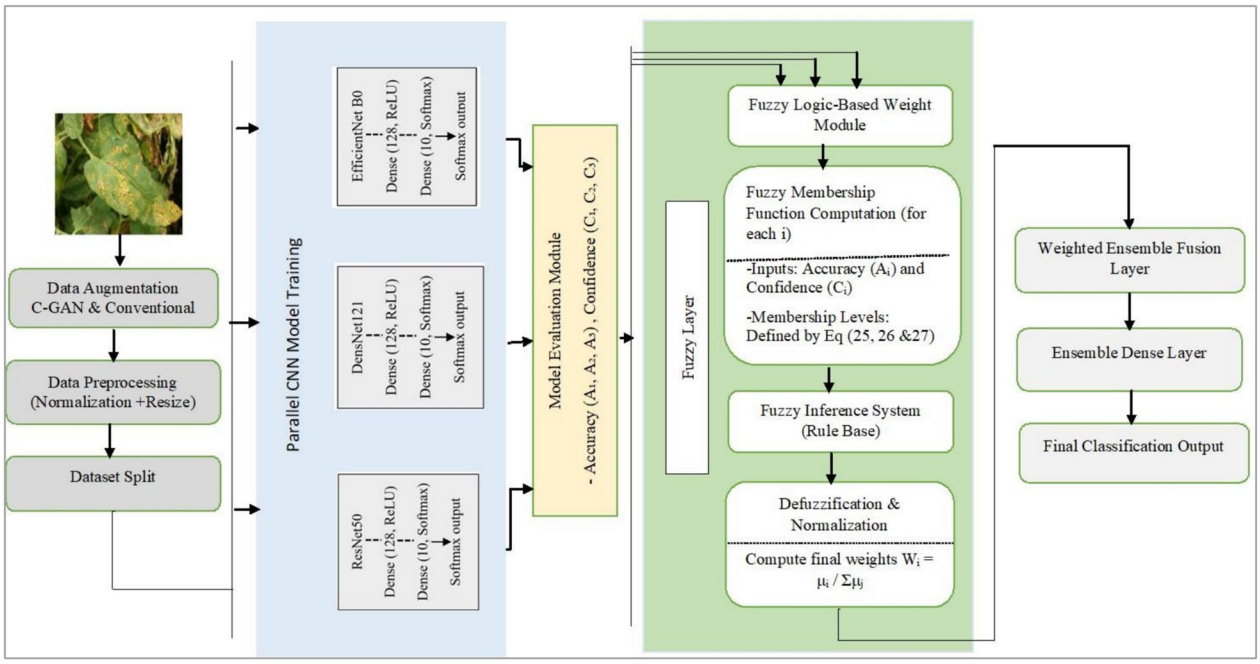



**Fig. 4**. Block diagram of the proposed fuzzy logic-based ensembling model for tomato crop classification.

*The ResNet50 model*

ResNet50 is a recognized architecture due to its residual learning topology, which allows efficient training of exceptionally deep networks[20,49]. The architecture initially starts with a '7 × 7' convolution layer consisting of 64 filters and a $'3x3'$ max pooling layer. A stride of 2 is kept for both the aforementioned layers to reduce the spatial dimensions of the input. The ResNet50 is organized into four stages, each containing a distinct residual block. Within each block are three convolutional layers: a '1 × 1', a '3 × 3', and another '1 × 1' convolution. The '1 × 1' convolutions allow for efficient computation. It first reduces and then restores the features. However, the $'3x3'$ convolution analyzes the spatial highlights. The shortcut connection plays a crucial role in these remaining blocks. It enables the input to bypass the convolutional layers and directly contribute to the output. The realization of the residual block is mathematically represented as follows:

$$y_a = F(x_a, \{W_i\}) + x_a \tag{5}$$

In Eq. (5), '$x_a$' is the input and 'Wi' stands for model's parameter. $'F(x_a, \{W_i\})'$ is the residual mapping, and $'ya'$ is the output. The residual blocks are the fundamental component in ResNet50, which comprises of three convolutional layers. This residual block generalizes the convolution layer as follows:

$$Y3 = \mathrm{ReLU}(\mathrm{BaN}(W * x_a + b)) \tag{6}$$

In Eq. (6), '$x_a$' represents the input data,' BaN' refers to the batch normalization function, and 'Y3' represents the output after the three convolutional layers. The final output at the residual block is represented as follows:

$$Y = \mathrm{ReLu}(Y_3 + x_p) \tag{7}$$

In Eq. (7), $'x_p'$ refers to the previous stage input to the current block. The utilization of identity mapping allows gradients to propagate directly throughout the network. This helps to alleviate the vanishing gradient problem. Moreover, Batch normalization streamlines training by normalizing inputs after each convolutional layer, therefore improving performance both during and after training. The application of ReLU activations not only adds non-linearity, but also average pooling reduces each feature map to one value, which summarizes a spatial region's contribution. After the flattened and fully connected layers, a SoftMax classification layer is used to produce a probability distribution for the different classes.

In this study, the ResNet50 model has been fine-tuned on a tomato dataset using transfer learning. Important hyperparameters are set and optimized so that the model accurately captures the underlying patterns within the data, particularly for the tomato leaves. The training begins with an initial learning rate of 0.001, which in turn is reduced by a factor of 0.1 if validation loss stagnates or does not decrease sufficiently. A batch size of 32 is chosen to optimize both memory utilization and training speed. We added a global pooling layer before the Softmax layer to reduce the dimensionality of the feature vector globally rather than locally. The network was trained for 50 epochs with an Adam optimizer under a dropout setting of 0.2, which allowed convergence without overfitting due to regularization.

*DenseNet121 model*

The architecture of the dense convolutional network comprises 121 layers, and every layer receives inputs from all previously processed layers[19]. In DenseNet121, each layer also has connections to all the subsequent layers in the network, which facilitates effective utilization of already computed features for subsequent computations as well as smooth gradient back propagation. A dense block is defined in the following way:

$$z_l = H_l([c_0, c_1, c_2, \ldots c_{l-1}]) \tag{8}$$

In Eq. (8), where '$z_l$' refers to output at the $l^{th}$ layer, the term [$c_0$, $c_1$, $c_2$…$c_{l-1}$] represents the concatenated features provided to function '$H_l$' from all preceding layers. The function '$H_l$' performs three successive operations:

$$H_l = (\mathrm{Conv}_{3\times3}(\mathrm{Relu}(\mathrm{BaN}([c_0, c_1, c, \ldots c_{l-1}])))) \tag{9}$$

In Eq. (9), initially the batch normalization (BaN) is applied, followed by ReLU activation, and finally 3 × 3 convolution (or Conv) to extract feature from the concatenated vector. The previously mentioned dense block features are flattened and fed through a Global Average Pooling (GAP) layer to the classification layer. The following layers, GAP and SoftMax features, are illustrated as follows:

$$Z = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j} \tag{10}$$

$$P = \mathrm{Softmax}(W_f Z + b_f) \tag{11}$$

In Eq. (10), the terms '$W$' and '$H$' represent the width and height of the feature map, while '$X_{ij}$' denotes the feature at position (i, j). The '$W_f$' refers to weight matrix, '$b_f$' is the bias term in Eq. (11), and 'Softmax' is a function that offers classification probabilities. This architecture alleviates the vanishing gradients problem by encouraging feature reuse and significantly improving gradient flow. Transition layers follow dense blocks to control the model's complexity. These layers employ batch normalization, a 1 × 1 convolution operation, and a pooling layer. The 1 × 1 convolution layer reduces the feature map, while the average pooling layer reduces spatial dimensions. The growth rate, typically between 12 and 32, balances the model's complexity and the computation's efficiency. A pooling layer after the convolutional layer further reduces the spatial resolution. It helps to capture intricate visual features from the images.

We set the initial learning value of 0.001 to train the model. Later, the learning rate decreases by a factor of 0.1 if validation loss stagnates. We select a batch size of 32 to optimize training efficiency and manage the increased memory demand due to dense connections. The network is trained for 80 epochs with the RMSprop optimizer. The design expedites the capture of underlying visual patterns in the data. This offers improved prediction accuracy in the ensemble framework with consistency and stability.

*The EfficientNetB0 model*

EfficientNet-B0 systematically increases model dimensions using a compound coefficient. It incorporates the mobile inverted bottleneck (MBConv) block into its architecture[21]. The model begins with a stem convolution block that handles the initial data processing. This block consists of convolutional layers, normalization, and activation functions. After the stem, the architecture utilizes MBConv blocks, which are improved versions of Mobile Inverted Residual Bottleneck blocks[50]. The base network MBConv is represented using the following equations:

$$f_1(X_{in}) = \mathrm{swish}(\mathrm{BaN}\left(\mathrm{Conv}_{1\times1}(X_{in})\right)) \tag{12}$$

$$f_2(X_{in}) = \mathrm{swish}(\mathrm{BaN}\left(\mathrm{Conv}_{k\times k}^{\mathrm{deptwise}}(f_1(X_{in}))\right)) \tag{13}$$

$$f_3(X_{in}) = \mathrm{SE}(f_2(X_{in})) \tag{14}$$

$$f_4(X_{in}) = \mathrm{dropout}(\mathrm{swish}(\mathrm{BaN}\left(\mathrm{Conv}_{1\times1}(f_3(X_{in}))\right))) \tag{15}$$

$$f_{\mathrm{MBConv}}(X_{in}) = X_{in} + f_4(X_{in}) \tag{16}$$

In above Eqs. (12–16), '$X_{in}$' stands for input variable, while 'swish', 'BaN' and 'SE' stand for swish, batch normalization and squeeze and excitation function respectively. This representation simplifies the expression by introducing intermediate functions '$f_1$,' '$f_2$,' '$f_3$,' and '$f_4$,' each representing a specific transformation within the MBConv block. The final composite function '$f_{MBConv}$ (Xin)' is the sum of the original input '$X_{in}$' and the output of the transformation $f_4(X_{in})$. This residual connection ensures that the original information is preserved while integrating the learned transformations from the convolutional layers and the squeeze-and-excitation mechanism. This model structure permits modification of depth, width, and resolution scaling allowing more accurate prediction without unduly inflating model size or computational cost. The compound scaling is represented as:

$$Beta depth = \alpha^{\Phi}, width = \beta^{\Phi}, resolution = \Upsilon^{\Phi} st\alpha.\beta^2.\Upsilon^2 \approx 2 and \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \tag{17}$$

In Eq. (17), α, β, and γ are fixed scaling coefficients that is determined using a grid search mechanism, while 'Φ' is the predefined compound scaling coefficient that scales all the three variables (depth, width, resolution) of the network in proportional manner. The EfficientNetB0 includes the squeeze-and-excitation (SE) blocks in its connected layers and employs the channel-wise attention and recalibration to enhance feature utilization. This mechanism can be stated as:

$$V = \sigma\left(g_1\left(Wx_z\right)\right) \odot x_f \tag{18}$$

In Eq. (18), 'σ' is the sigmoid function, "$g_1$" is the ReLU function, "W" stands for weights, '$x_z$' represents the squeeze input obtained by GAP operation and ⊙ stands for multiplication by original feature map '$x_f$'. It helps to separate the emphasized features by dropping the less important ones. Another innovative aspect of EfficientNetB0 is the use of the Swish activation function, which is defined as:

$$\text{Swish}\left(x\right) = x * \sigma(x) \tag{19}$$

In Eq. (19), the factor 'x*σ(x)' enhances the rates of gradient flow, hence the enhancement on the learning abilities of the model. To fine-tune the EfficientNet-B0, the cosine annealing function sets a decay to the learning rate from its initial value (0.01) over time. This results in better convergence of the model. A batch size of 32 balances the computational efficiency and memory consumption.

*Ensemble strategy and weight aggregation*

The proposed ensemble learning structure, shown in Fig. 4, begins with data preprocessing and augmentation which yields an augmented dataset '$D_a$'. Each pre-trained model '$M_i$' is trained independently on $D_a$ to extract distinct feature representations. Each CNN model produces a probability distribution across various classes using the Softmax activation function. The Softmax function transforms the logit scores '$z_i$' into a categorical probability mass function (PMF). For a specific model 'i', the probability assigned to class 'j' is calculated as follows:

$$p_{i,j} = \frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}} \tag{20}$$

In Eq. (20), '$p_{i,j}$' indicates the probability to class 'j'. The '$z_j$' represents the pre-SoftMax output for class 'j', while 'n' means the total number of classes. This PMF is fundamental to Maximum Likelihood Estimation (MLE) in classification. The prediction confidence score '$Cf_i$' for a given sample is defined as the maximum element of the output PMF, representing the Maximum A Posteriori (MAP) probability from model 'i'. The confidence score for a model is obtained by selecting the maximum probability from the output of the SoftMax function for a given sample.

$$C_{fi} = \max(p_{i,1}, p_{i,2}, p_{i,3} \cdots p_{i,n}) \tag{21}$$

In the above Eq. (21), '$C_{fi}$' represents the confidence value for a model 'i', while the '$p_{i,n}$' denotes the probability of class 'j' from a model 'i'.

A Fuzzy Inference System (FIS) is constructed to calculate a dynamic, instance-specific weight '$W_i$' for each model '$M_i$'. The linguistic variables 'Ac' and '$C_f$' are mapped onto the fuzzy sets S ∈ {"Low"(L),"Medium"(M),"High"(H)} using triangular membership functions $\mu S(x) : [0,1] \rightarrow [0,1]$.

The fuzzy logic system outlined in this study employs triangular membership functions[49]. The membership values for high, medium, and low accuracy and confidence are estimated based on their individual performance. We achieved accuracies of 97.8%, 97%, and 96.5% using pre-trained architectures EfficientNet-B0, DenseNet121, and ResNet-50, respectively, on tomato datasets. As shown in Figs. 5 and 6, these outcomes provided a foundation for designing fuzzy membership functions within the framework.

We utilized the statistical properties of the Normal Distribution to determine the standard deviation parameters, $\sigma_{Cf} and \sigma_{Ac}$. The parameters were chosen so that $\mu_{H(x)}$ for the high set became nearly zero at the lower boundary. This fixed the base of the fuzzy set. The Gaussian membership functions were first adopted as a base to formulate the boundaries between fuzzy sets categorized as Low, Medium, and High depending on the accuracy (Ac) and confidence ($C_f$) metrics. Nevertheless, after running some tests, the final selection was the triangular membership functions because they were easier and faster to compute while retaining the same computation efficiency.

To transform accuracy ($A_c$) and confidence ($C_f$) into fuzzy sets of low, medium, and high, we use fuzzy membership functions based on Gaussians. The membership functions are defined as follows:

$$\mu_{h,C_f}\left(x\right) = e^{-\frac{(x-1)^2}{2\sigma_{C_f}^2}}, \mu_{h,A_c}\left(x\right) = e^{-\frac{(x-0.978)^2}{2\sigma_{A_c}^2}} \tag{22}$$

where in Eq. (22), the '$\sigma_{Cf}$' and '$\sigma_{Ac}$' control the spread of the Gaussian function. $\sigma_{Cf} = 0.075$ ensures that $\mu_{h,Cf}(x) \approx 0$ for $x \leq 0.85$. However, $\sigma_{Ac} = 0.004$ ensures that $\mu_{h,Ac}(x) \approx 0$ for $x \leq 0.97$.

Each membership function defines the desired transition by the values of $\sigma_{Cf}$ and $\sigma_{Ac}$. The term '$\sigma_{Cf}$' determines the rate of change of '$\mu_h$, $C_f(x)$' to 1, when x approaches 1 for confidence. The '$\sigma_{Ac}$' defines the rate of change of '$\mu_h$,$A_c(x)$' to 1, when x approaches 0.978 (EfficientNetB0) for accuracy. It is a known property of Gaussian functions that roughly 68% of the area of a Gaussian curve is within one standard deviation σ from

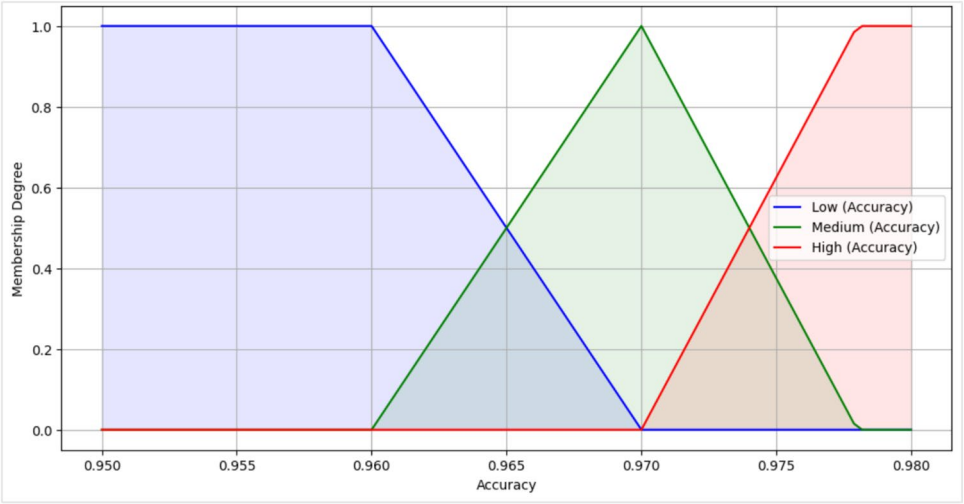



**Fig. 5**. Fuzzy Membership Functions for Model Accuracy: Low, Medium, and High Ranges.

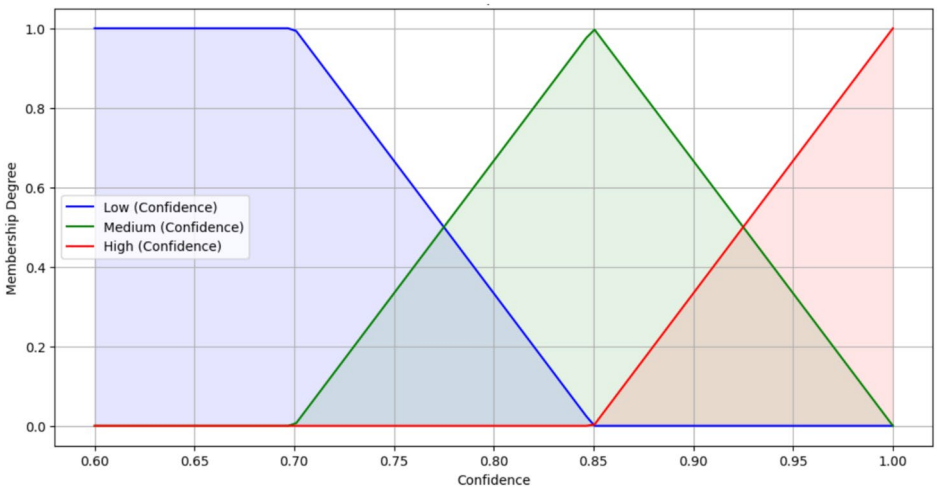



**Fig. 6**. Fuzzy Membership Functions for Model Confidence: Low, Medium, and High Ranges.

the mean. This fact can be utilized in determining '$\sigma_{Cf}$' and '$\sigma_{Ac}$' for the areas where we want the membership values to be close to 1.

For '$\sigma_{Cf}$', the membership function $\mu_h$,$C_f(x)$ is specified to take on values near zero for all $x \leq 0.85$, and to approach one as x increases towards one. In order to accomplish this gradual transition, the mean of the Gaussian function is set at $x = 1$. Furthermore, to ensure that $x = 0.85$ is roughly two standard deviations below the mean, the value of $\sigma_{Cf}$ is chosen accordingly. To make the function close to 0 around 0.85, we'll set '$\sigma_{Cf}$' so that $x = 0.85$ is approximately 2 standard deviations below the mean. The general form of confidence is defined as follows:

$$\mu_{h,C_f}(x) = e^{-\frac{(x-1)^2}{2\sigma^2_{C_f}}}, such\ that \sigma_{C_f} = \frac{1 - M^i(C_f)}{2} \tag{23}$$

For '$\sigma_{Ac}$', the mean of '$\mu_{h,Ac}(x)$' in Eq. (23) is set near 0.978, where the accuracy membership is intended to be high. We want the function to be close to 0 around $x = 0.97$, so we can set '$\sigma_{Ac}$' to make $x = 0.97$ roughly 2 standard deviations below the mean. Thus, the general form of the accuracy membership function is defined as follows:

$$\mu_{h,A_c}(x) = e^{-\frac{(x-0.978)^2}{2\sigma^2_{A_c}}} \text{ such that } \sigma_{A_c} = \frac{M^i(A_c^H) - M^i(A_c^L)}{2} \tag{24}$$

The estimated value for confidence $\sigma_{Cf} \approx 0.075$ and $\sigma_{Ac} \approx 0.004$ ensures that the Gaussian membership functions approximate the intended boundaries and transitions for high membership.

The same approximations were obtained from Gaussian transitions for middle and lower membership levels. The triangular function was empirically chosen to save on computational costs. Thresholds from Gaussian functions were retained as detailed in Eqs. (25–27).

Fuzzy set theory offers a framework for depicting and manipulating vague or uncertain information. In this context, we treat the accuracy (Ac) and prediction confidence ($C_f$) of each base model as linguistic variables. Rather than assigning these parameters fixed numerical values, decisions based on them are made using categorization like "Low", "Medium", and "High". Their definition is precisely described by the corresponding membership functions which mathematically articulate the fuzzy boundaries within which these terms apply. A vital attribute of the selected membership functions is their convexity.

A fuzzy set A is considered convex if, for any two elements $x_1, x_2$ within its domain and any convex combination $\lambda x_1 + (1—\lambda)x_2$ (where $\lambda \in [0, 1]$), the membership degree of the combined element is greater than or equal to the minimum of the individual membership degrees: $\mu_x(\lambda x_1 + \lambda)x_2) \geq \min(\mu_x(x_1), \mu_x\ (1—\lambda)x_2) \geq \min(\mu_x(x_1), \mu_x(x_2))$. This condition guarantees that fuzzy sets represented graphically as triangles or shoulder functions will not exhibit illogical discontinuities. It ensures that the concept encapsulated by a given linguistic label, such as Medium or Low, changes across its domain smoothly without abrupt shifts. This adds uniform interpretation by the fuzzy inference system[49].

For high membership, EfficientNet-B0 (97.8%) is assigned the highest weight due to its accuracy being close to 1. DenseNet121 (97%) also receives a significant weight, though it is slightly lower. ResNet-50 (96.5%) is given a lower weight as it falls below the 0.97 threshold for high membership. The higher membership is defined as follows:

$$\mu_h(x) = \begin{cases} 0 & \text{if } x < 0.85\,(\text{for } C_f) \text{ or } x < 0.97(\text{for } A_c\ ) \\ \frac{x-0.85}{0.15} & \text{if } 0.85 \leq x < 1(\text{for } C_f) \\ \frac{x-0.97}{0.008} & \text{if } 0.97 \leq x < 0.978(\text{for } A_c) \\ 1 & \text{if } x \geq 1\,(\text{for } C_f)\ \text{ or } x \geq 0.978(\ \text{for } A_c) \end{cases} \tag{25}$$

In above Eq. (25), the higher membership function is defined for both '$A_c$' and '$C_f$' that maps the input value to fuzzy sets (i.e. low, medium and high). For medium member ship, ResNet-50 (96.5%) is categorized in the medium membership range and thus receives a lower weight compared to the higher-accuracy models. DenseNet121 (97%) is classified with medium–high membership, while EfficientNet-B0 (97.8%) has a minimal medium contribution, primarily benefiting from its high membership weighting. The medium membership function is defined as follows:

$$\mu_m(x) = \begin{cases} 0 & \text{if } x < 0.70\,(\text{for } C_f)\ \text{ or } x < 0.96(\text{for } A_c) \\ \frac{x-0.70}{0.15} & \text{if } 0.70 \leq x < 0.85(\text{for } C_f) \\ \frac{x-0.96}{0.01} & \text{if } 0.96 \leq x < 0.97\ \ (for A_c) \\ \frac{0.978-x}{0.008} & \text{if } 0.97 \leq x < 0.978(\text{for } A_c) \\ \frac{1-x}{0.15} & \text{if } 0.85 \leq x < 1(\text{for } C_f) \\ 0 & \text{if } x \geq 1 \end{cases} \tag{26}$$

In above Eq. (26), the medium membership function maps accuracy and confidence input into fuzzy sets. For low member ship, DenseNet121 and EfficientNet-B0, both achieving accuracies over 0.97, are not penalized for low membership and will play a crucial role in the ensemble. In contrast, ResNet-50, which has an accuracy of 96.5%, is categorized in the medium–low membership range, leading to a diminished impact on the final prediction compared to the higher-accuracy models. The lower membership is defined as follows:

$$\mu_l(x) = \begin{cases} 1 & \text{if } x < 0.70\,(\text{for } C_f)\ \text{ or } x < 0.96\,(\text{for } A_c) \\ 1 - \frac{x-0.70}{0.15} & \text{if } 0.70 \leq x < 0.85(\text{for } C_f) \\ 1 - \frac{x-0.96}{0.01} & \text{if } 0.96 \leq x < 0.97(\text{for } A_c) \\ 0 & \text{if } x \geq 0.85\,(\text{for } C_f) \text{ or } x \geq 0.97\,(\text{for } A_c) \end{cases} \tag{27}$$

In above Eq. (27), the low membership function maps accuracy and confidence input into fuzzy sets.

The FIS uses a set of fuzzy If–Then rules to compute an initial weight '$W_{(i,k)}$' for model 'i' based on rule 'k'. The minimum operator is used for the conjunction of the input membership values, which is defined as follows:

$$W_{i,k} = \min(\mu_{set_A}(Ac_i), \mu_{set_C}(Cf_i)) \tag{28}$$

where in Eq. (28), $set_A$, $set_C \in \{L, M, H\}$ are the membership sets for accuracy and confidence in rule 'k'. It follows a specific set of rules as provided in Table 3. The rules ensure that the most reliable models significantly influence the final ensemble prediction. The rules depicts that if the accuracy is high and the confidence is high, the weight is high. However, if the accuracy is high but the confidence is low, the weight is medium. Conversely, when both accuracy and confidence are low, the weight is low, and if accuracy is high but confidence is low, the weight stays medium. Specifically, the four primary rules are defined as follows:

These rules ensure that the most reliable models significantly influence the final ensemble prediction. The accuracy (Ac) primarily determines the potential ceiling of the model's weight ($W_i$). It acts as a structural bias in the rule base, ensuring that the best-performing architecture (EfficientNetB0) receives a higher starting membership degree ($\mu_{(H,Ac)}$) for high weights. It is essential for the crop disease classification task, as it reflects the inherent architectural advantage of certain models over others in generalizing across diverse disease classes.

| Rule k | Condition | Weight $W_{(i,k)}$ |
|---|---|---|
| 1 | $Ac_i$ is H AND $Cf_i$ is H | $\min(\mu_H(Ac_i), \mu_H(Cf_i))$ |
| 2 | $Ac_i$ is M AND $Cf_i$ is H | $\min(\mu_M(Ac_i), \mu_H(Cf_i))$ |
| 3 | $Ac_i$ is H AND $Cf_i$ is M | $\min(\mu_H(Ac_i), \mu_M(Cf_i))$ |
| 4 | $Ac_i$ is M AND $Cf_i$ is M | $\min(\mu_M(Ac_i), \mu_M(Cf_i))$ |

**Table 3**. Fuzzy Rules Calculation in the Proposed Model.

The confidence score is a dynamic measure calculated for every single image at inference time. It represents the certainty of the model's prediction for that specific sample.

The score '$Cf_i$' acts as the instance-specific modulator of the weight ($W_i$). If a high-accuracy model (EfficientNetB0) encounters a confusing image (e.g., a sample with multiple symptoms) and produces a low '$Cf_i$, its weight is immediately suppressed via the fuzzy rule. This is critical in crop disease classification, where images often contain noise, artifacts, or ambiguous symptoms. '$Cf_i$' prevents a highly accurate model from dominating the final vote when it is clearly unsure about a particular difficult case, shifting the voting power to models that are more certain for that instance.

The integration of $A_c$ and $C_f$ via the T-Norm (min operator) enforces a conservative, risk-averse weighting strategy. The rules determine the degree of trust assigned to model *i* for a specific image. A higher weight is given only when the model has both strong past performance and high confidence in its current prediction. For every CNN model in the ensemble, the weight is dynamically computed during the training process using the fuzzy system. The final weight for model 'i 'is determined as follows:

$$W_i = \frac{\mu^i_{A_c} \times \mu^i_{C_f}}{\sum_{i=1}^{N} \mu^i_{A_c} \times \mu^i_{C_f}} \tag{29}$$

In the above Eq. (29), the terms '$\mu_{Ac}$', and '$\mu_{Cf}$' represent the membership value for model 'i'. 'N' indicates the total number of models. The final forecast is a weighted combination of each individual model predictions based on their calculated weights. Every rule is assessed through the minimum of membership degrees for both accuracy and confidence. The weight '$W_f$' for each model is calculated as:

$$W_f = w_{low} \times \mu_L\left(A^i_c, C^i_f\right) + w_{medium} \times \mu_M\left(A^i_c, C^i_f\right) + w_{high} \times \mu_H\left(A^i_c, C^i_f\right) \tag{30}$$

In the above Eq. (30), we used the minimum value for the low, medium membership degree and high membership degree were used to aggregate the fuzzy weight. In Eq. (33) '$w_{low}$', '$w_{medium}$', '$w_{high}$' are coefficients that have been assigned to the low, medium, and high membership functions, respectively. Initially, this value was kept equal to 0.33. The fuzzy weighting mechanism dynamically adapts throughout training: when a model's confidence and validation accuracy improve, its membership degrees shift toward the higher linguistic region, thereby increasing its weight without manual intervention. The aggregated prediction is calculated by summing the weighted predictions '$p_{i,j}$' from all L models:

$$\hat{p}_i = \sum_{j=1}^{L} W_j p_{i,j} \;\; such\ that \sum_{j=1}^{L} W_j = 1 \tag{31}$$

After training, we aggregate the predictions of the individual models using the previously calculated weights. The class with the maximum aggregated prediction probability $\widehat{p}^c_i$ is identified.

$$\widehat{y}_i = \text{argmax}\widehat{p}^c_i \tag{32}$$

The model is trained using the Adamax optimizer, selecting categorical cross-entropy as the loss function and defining accuracy as the performance metric. The loss function is outlined as follows:

$$L_{cce} = -\sum_{i=1}^{C} y_c \log P_{z,i} \tag{33}$$

In Eq. (33), 'C' is the number of classes, '$y_c$' is the true label for class 'i' for sample m, and '$P_{z,I}$' is the predicted probability by the model for class 'i'. Adamax is a variant of Adam used during training to achieve optimal parameters[45]. Unlike Adam, which uses the second moment, Adamax utilizes the infinity norm (maximum absolute value), providing greater stability in certain scenarios and being less sensitive to outliers. The infinity norm helps ensure that the maximum absolute value does not allow the largest gradient in a layer to dominate the adjustment of the learning rate. When using Adamax, the parameters are updated as follows:

$$\theta_J^{t+1} = \theta_J^t - \frac{\eta\prime}{\|r_t\|_\infty} m_t \tag{34}$$

In Eq. (34), '$\theta_j$' represents the model parameters, and 'η" indicates the learning rate. The terms '$m_t$' and '$r_t$' refer to the first moment and infinity norm, respectively. To the proposed ensemble model, dense layers with ReLU activation and Batch Normalization are incorporated to enhance learning efficiency, while Dropout prevents overfitting by randomly deactivating neurons during training. The final output layer employs a Softmax activation for multi-class classification.

The final layer of the output layer derived from the Softmax activation function, depicts multi-classification. The ensemble model was trained using the categorical cross entropic loss function with Adam optimizer and the adaptive learning rate with a decay of 0.0001. In summary, the fuzzy rule base is specifically designed to interpret the conjunction (min operator) of the membership degrees derived from these static and dynamic inputs. This results in a highly adaptive weight '$W_i$' that is not only influenced by the model's proven historical performance ($A_c$) but is also responsive to its certainty about the current input sample ($Cf_i$).

## Experimental results

The proposed fuzzy logic-based ensemble method was compared with several existing models that have been applied to the PlantVillage tomato leaf dataset. These include Inception V3 Module and Rainbow Concatenation[7], DenseNet(Densely Connected Convolutional Networks)[18], EfficientNet B0 (fine-tuned)[20], ResNet-50 + SeNet(Squeeze-and-Excitation Network)[33], DenseNet + C-GAN[34], CNN-GA(Convolutional Neural Network—Genetic Algorithm Weighted Average Ensemble)[39], VGG-16 + NASNet Ensemble (Neural Architecture Search Network Ensemble)[29], Deep Neuro-Fuzzy Neural Network[14], R-CNN (Region-based Convolutional Neural Network)[15], and Segmented-CNN[40], Vision Transformer[43], Swin Transformer[45].

### Experimental setup and dataset preparation

This section highlights the datasets used, preprocessing and splitting strategies, as well as the hardware, software, and hyperparameters configuration employed for training the proposed hybrid model.

*Dataset description and characteristics*

Table 4 summarizes four datasets. PlantVillage is large and clean but lab-based. Tomato Leaves includes field images but lacks geographic variety. PlantDoc has realistic conditions but few tomato samples. Tomato-Village adds regional diversity with real-world farm images but lacks size and environmental detail.

The distribution of data is shown in Table 5. Supplementing with field data increased the range of illumination, resolution, and occlusion by leaves which greatly improves the robustness of the model to real world conditions.

To ensure precision and dependability, the dataset was meticulously carefully curated and partitioned. High-resolution images were captured both within and outside of controlled environments. All datasets were obtained from multiple online repositories to ensure sufficient variety and real-world applicability. Some classes were underrepresented, so class imbalance mitigation measures were applied to improve model performance across all disease types. Before model training, images underwent augmentation and preliminary cleaning to maintain consistency, quality, and reliability.

*Composition and splitting of the dataset*

The selected tomato leaves datasets contain sufficient samples to ensure that it perseveres significant diversity. It is a combination of laboratory-captured and real-field images. To train the CGAN, a sample subset was chosen from the original data to prevent the leakage of data. To increase diversity and reproduce the variability of real world, 10 percent of the samples were distorted to imitate the on different illumination conditions by changing the intensity (I) element of the image. An additional 10% of the data was used to create image patches of varying sizes, representing partial and localized disease symptoms. The final training dataset was assembled by adding 10 percent CGAN-generated images, 10 percent samples altered for varying light, and another 10 percent drawn from patch-based acquisitions. The remaining 70 percent consists of original images taken from the base dataset, with mild augmentations applied to underrepresented classes.

| Dataset name | Web Source | Dataset Description |
|---|---|---|
| PlantVillage | https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset | A large and diverse dataset with over 54,000 images. It includes 38 plant species and 11,870 tomato samples. The tomato samples cover 10 different disease classes<br>Some disease types are underrepresented which introduce mild class imbalance<br>Well-labelled & organized<br>Images captured under controlled laboratory conditions with uniform backgrounds (white/grey) and consistent illumination |
| Tomato Leaves Dataset | https://www.kaggle.com/datasets/ashishmotwani/tomato? | Over 20,000 samples of tomato leaves with 10 diseases and 1 healthy class are available<br>Collected from both lab scenes and in-the-wild scenes. This helps the model generalize better for practical use<br>Field images suffering from variable lighting, poor focus are available. Need intensity correction and adjustment |
| PlantDoc dataset | https://www.kaggle.com/datasets/abdulhasibuddin/plant-doc-dataset | Comprises real in-field images of diseased and healthy plant leaves with natural backgrounds and illumination<br>Contains 577 tomato images across 8 classes with separate test samples available<br>Enhances robustness evaluation under non-laboratory conditions |
| Tomato-Village dataset | https://github.com/mamta-joshi-gehlot/Tomato-Village | Focuses on nutrient deficiency and viral disorders such as Magnesium Deficiency, Nitrogen Deficiency, Potassium Deficiency, and Spotted Wilt Virus |

**Table 4**. Summary and description of tomato leaf datasets used in the study.

| Disease/Class | Number of Images in PlantVillage Dataset | Number of Images in Tomato Leaves Dataset | Number of Images in PlantDoC Dataset | Number of Images in Tomato-Village |
|---|---|---|---|---|
| Bacterial Spot | 1,612 | 2,826 | 98 | |
| Early Blight | 1,000 | 2,455 | 74 | 347 |
| Healthy | 1,251 | 3051 | 54 | 151 |
| Late Blight | 1,209 | 3113 | 101 | 632 |
| Leaf Mold | 952 | 2754 | nil | nil |
| Septoria Leaf Spot | 1,379 | 2882 | 137 | nil |
| Spider Mites | 1,257 | 1747 | nil | nil |
| Target Spot | 988 | 1827 | nil | nil |
| Tomato Mosaic Virus | 373 | 2,153 | 44 | nil |
| Tomato Yellow Leaf Curl Virus | 1,849 | 2039 | 69 | nil |
| Powdery Mildew | nil | 1,004 | nil | nil |
| Magnesium Deficiency | nil | nil | nil | 655 |
| Nitrogen Deficiency | nil | nil | nil | 251 |
| Potassium Deficiency | nil | nil | nil | 50 |
| Spotted Wilt Virus | nil | nil | nil | 363 |
| Leaf Miner | nil | nil | nil | 716 |
| Total Training Samples | 11,870 | 27,851 | 577 | 3165 |

**Table 5**. Distribution of Tomato Leaf Disease Samples across Classes.

| S. No | Web Source | Datasets | Classes (Only Tomato leaf) | Training Samples After Augmentation | Validation Samples | Testing Samples |
|---|---|---|---|---|---|---|
| 1 | https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset | PlantVillage Dataset | 10 | 10,000 | 1000 | 1000 |
| 2 | https://www.kaggle.com/datasets/ashishmotwani/tomato? | Tomato Leaves Dataset | 11 | 12,000 | 1200 | 1200 |
| 3 | https://www.kaggle.com/datasets/abdulhasibuddin/plant-doc-dataset | PlantDoC Dataset | 8 | 577 | - | 108 |
| 4 | https://github.com/mamta-joshi-gehlot/Tomato-Village | Tomato-Village | 8 (Used 4) | 3552 | 444 | 444 |

**Table 6**. Datasets information and training samples distribution.

As shown in Table 6, the datasets were split into three sets in accordance with the common practices of deep learning: 80% for training, 10% for validation, and 10% for testing. The reserved training set was ensured at 80% so that an appropriate amount of data is available for the model to learn and be able to generalise.

*Hardware, software, and hyperparameters configuration*

We evaluated both the proposed and existing approaches on a system equipped with an Intel Core i7 processor, 32 GB of RAM, and Windows 11. For quicker processing in deep learning and better overall computational speed, particularly during CNN training, a Google Colab-GPU was used. Development, testing and evaluation of the proposed method were performed in Python using the TensorFlow, Keras, and PyTorch frameworks for model development and training, and NumPy and Pandas for data processing. This combination of tools offered sufficient hardware and software support for optimal model development and evaluation.

In the proposed methodology, hyperparameters optimisation is essential for enhancing the model's performance. The subsequent table delineates the principal hyperparameters for each element of the hybrid framework (CNN models, fuzzy logic ensemble, and C-GAN) along with their selected values. These values were chosen based on empirical testing and performance verification. Table 7 presents the hyperparameters details for proposed Hybrid model.

## Simulation results

This section covers the simulation results and discussion.

*Performance measuring parameters*

Accuracy, precision, recall, and F1-score are some of the key statistical measures that were employed to evaluate the performance of the classification model. All these measures are used to measure the ability of model to differentiate between healthy and diseased leaf categories with a high degree of reliability[3]. The common measure is accuracy (Ac), which is the percentage of correct samples that are categorized out of the total number of sampled. It is obtained as follows:

| Hyperparameters | Proposed Value | Description |
|---|---|---|
| Learning Rate | 0.001 | Determines the step size at each iteration while moving toward a minimum during training |
| Batch Size | 32 | Number of training samples used in one iteration to update model weights |
| Number of Epochs | 50 | Total number of complete passes through the training dataset |
| Optimizer | Adam | Optimizer used to minimize the loss function, providing efficient learning |
| Dropout Rate | 0.5 | Dropout is used to prevent overfitting by randomly dropping a percentage of neurons during training |
| ResNet-50 Learning Rate | 0.0001 | Learning rate for ResNet-50 to ensure stable and fine-tuned training |
| EfficientNet-B0 Learning Rate | 0.0001 | Learning rate for EfficientNet-B0 to balance optimization |
| DenseNet121 Learning Rate | 0.0001 | Learning rate for DenseNet121 to prevent overfitting while optimizing accuracy |
| Fuzzy Logic Rule Base | 9 rules | The number of rules used for the fuzzy logic-based weighting mechanism in the ensemble |
| C-GAN Latent Vector Size | 100 | Size of the latent vector used in the generator of the C-GAN |
| C-GAN Batch Size | 64 | Batch size used in the C-GAN model for training the generator and discriminator |
| C-GAN Epochs | 100 | Number of epochs used to train the C-GAN for generating synthetic samples |
| Synthetic Data Augmentation Ratio | 0.2 | Percentage of synthetic data generated by C-GAN relative to the original dataset |
| Ensemble Weighting Mechanism | Dynamic based on accuracy and confidence | The weighting mechanism of the ensemble is adjusted dynamically based on the individual models' performance |

**Table 7**. Hyperparameters Details for Proposed Hybrid Model.

$$A_c = TP_p + TN_n/(TP_p + TN_n + FP_p + FN_n) \tag{35}$$

In Eq. (35), the variables True Positives ($TP_p$) and True Negatives ($TN_N$) refer to the cases of the correct positive and negative predictions respectively. Any case that has been falsely classified as positive is referred to as a False Positive (FPp). False Negative is one such example that is wrongly classified as negative. The ROC curve indicates the effectiveness of the classifier in separating the various classes as the decision threshold varies. The need for precision (Pr) is especially relevant in the case of classifying the different diseases of tomatoes. Importantly, recall (Re) is a key measure for assessing the correctly diagnosed diseased cases, which is vital in controlling the disease's spread. The F1 score is a measure of the balance between precision and recall and focuses on the extremes of the two. The 'Pr', 'Re', and F1-score can be defined as follows:

$$P_r = TP_p/TP_p + FP_p \tag{36}$$

$$R_e = TP_p/TP_p + FN_n \tag{37}$$

$$F1 - Score = 2 \times Re \times Pr /(Pr + Re) \tag{38}$$

*Class wise performance results*

Figure 7 showcases confusion matrix of proposed architecture based on the PlantVillage dataset with 10 classes. The confusion matrix gives a clear view of how accurately the model classifies data across ten categories. Correctly classified samples are shown in diagonal elements and misclassified samples are shown in the off-diagonal elements. Most classes exhibit excellent accuracy, with the model consistently predicting the correct label (diagonal dominance). For instance, BS and LM show almost perfect discrimination. In matrix, it indicated a values of 108 on the diagonal and minimal off-diagonal errors. Very few misclassifications have occurred and the confusion appears to come from closely related diseases such as EB, LB and Healthy where some samples are erroneously classified as either similar diseases or healthy leaves. The classifier pertinent to the ten classes of tomato leaves was reliable and accurate as proven by the low remaining entries off-diagonal. These results show that the model was able to generalize well to the real situation.

Figure 8 illustrates the misclassification rate for each class in a tomato dataset comprising ten classes. This graph shows the misclassification rate of each class and provides a clue regarding the particular challenges of the model. Diseases which have similar visual symptoms, including early blight (EB) and late blight (LB), have slightly higher misclassification rates than those with completely different diseases. The proposed hybrid model is better than the baseline approaches and can significantly minimize these errors due to dynamic weighting strategy and synthetic data augmentation methods. The overall misclassification rate of 0.82 percentage is an indication of very low error rate and accurate predictions of the model over the test dataset.

As shown in Fig. 9, the model achieves a high training accuracy of 99.3% with a low training loss of 0.021. Furthermore, the recorded validation accuracy and loss metrics of 99.4% and 0.021 respectively indicate effective generalization with minimal overfitting, as well as demonstrating efficient use of computational resources. Also, the recorded test accuracy and test loss of 99.2% and 0.015 respectively indicate strong performance on latent domain data. It can be noted that over training, validation, and testing phases, the model demonstrates consistent predictive performance, which indicates dependable classification alongside low predictive variance. The model also maintained consistently low loss values indicating intelligent error reduction with high precision in predicted outcomes throughout all tested scenarios as well as confident alignment to the predictions made during previous assessments across all datasets. Together, these results validate the model's performance with respect to the precise and timely classification of real-time tomato diseases.

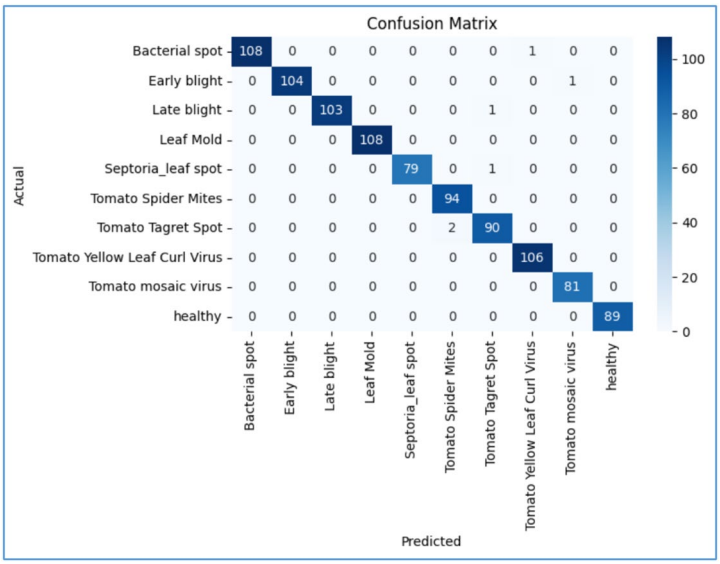



**Fig. 7**. Confusion matrix of proposed on PlantVillage dataset with 10 classes.

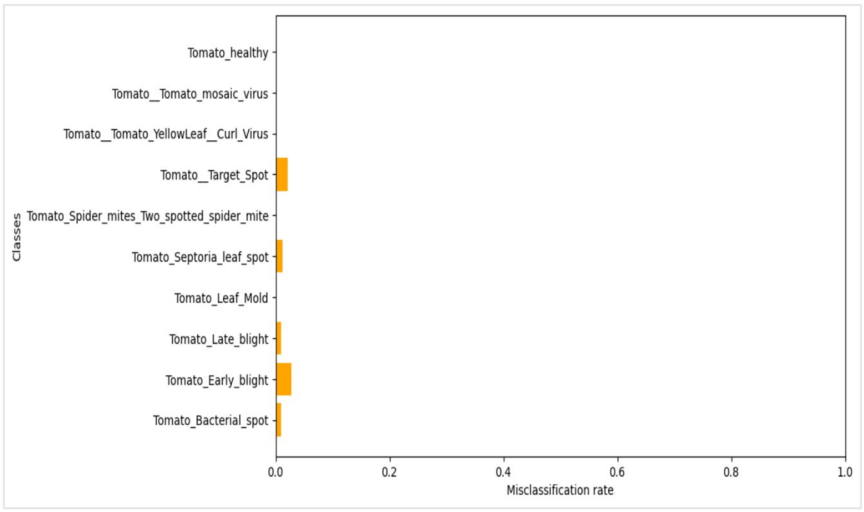



**Fig. 8**. Class-wise misclassification rate on tomato dataset with 10 classes.

The class-wise precision, recall and F1 metric of the proposed hybrid approach of tomato disease identification is presented in the Fig. 10. All classes in the figure achieve high precision, recall and F1-measure which are all above 98%. For instance, the "BS" (Bacterial Spot) and "Healthy" classes achieve almost perfect scores in all three metrics. Other classes such as "EB," "LB," and "LM" also achieve above 98% in all three metrics.

These results highlight several important aspects. The high precision values imply that the majority of leaves classified as diseased are diseased, which minimizes unnecessary interventions. Nevertheless, the fact that the recall was high in all classes indicates that the model is able to identify most diseased leaves reliably. This minimizes the cases of missed diseases. High F1-scores indicate the balanced power of the model in terms of precision and recall, which proves that the model is reliable, even when the data on a certain class is less rich. The consistently high values of the metric used in all types of diseases indicate the high quality and unbiased detection ability of the hybrid model, which is particularly applicable to the practical implementation of the hybrid model in crop health monitoring.

Table 8 compares the performance of the proposed hybrid approach with existing models based on precision, recall, and F1-score over 10 classes. The proposed hybrid approach scores nearly perfect in the majority of the classes, such as 100% precision, recall, and F1-score in Bacterial Spot and Healthy leaves. In hard categories such as Early Blight (EB) and Late Blight (LB), the strategy achieves 100% of precision and 97–99% of recall, which is more promising than previous models like Inception V3, DenseNet, and ResNet-50 + SeNet [T8.

Importantly, for rare or imbalanced classes like Mosaic Virus (MV) and Yellow Leaf Curl Virus (YLC), precision, recall, and F1-score remain above 98%. This reflects a superior generalization. These results are

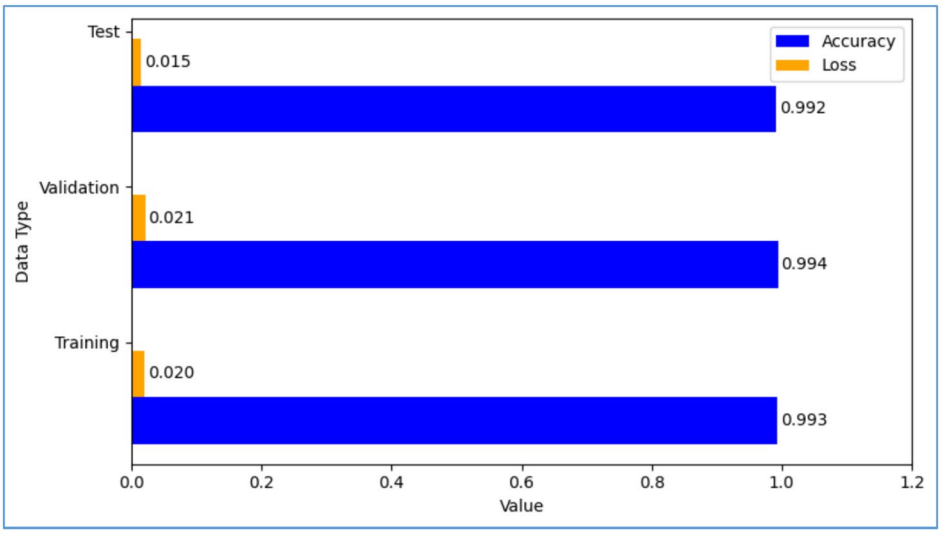



**Fig. 9**. Training, validation, testing accuracy and loss metric of proposed model.

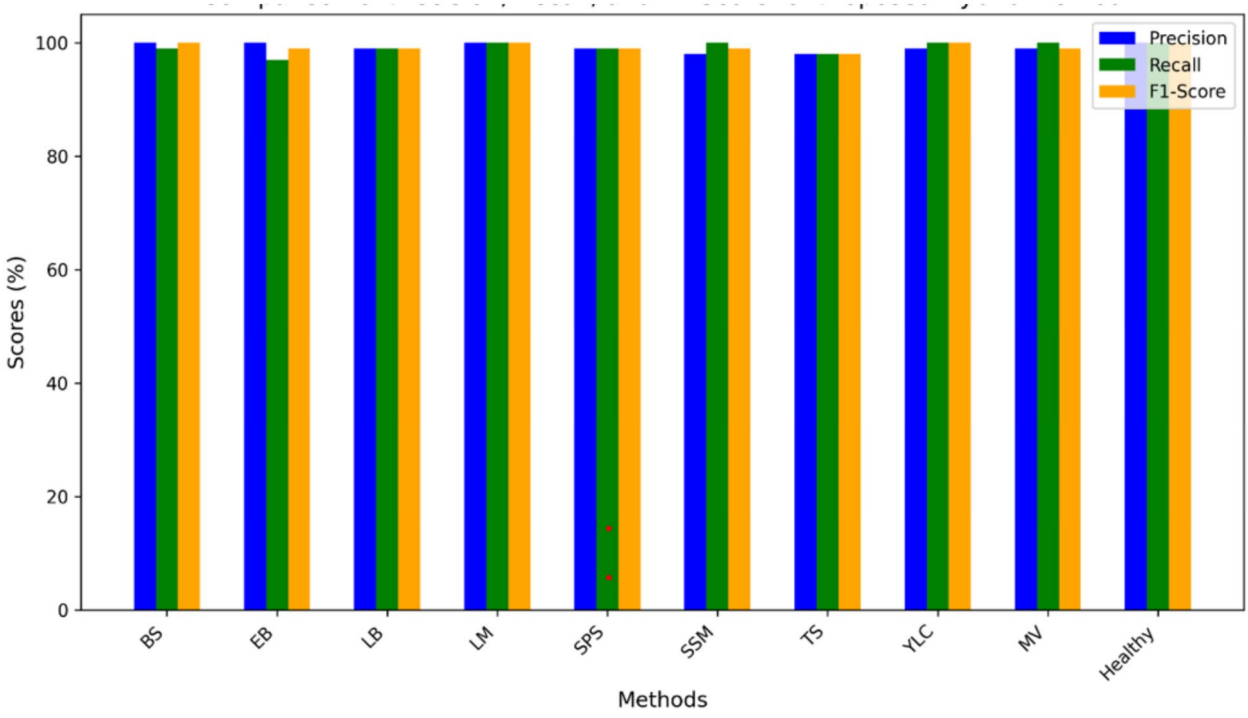



**Fig. 10**. Class-wise Precision, Recall, F1 score of proposed model on tomato dataset with 10 classes.

attributed to dynamic fuzzy weighting and synthetic data augmentation, which help prevent bias toward frequent classes. The mean F1-score achieved by the hybrid model was 99.3, a notable increase relative to the 94.8, 95.4, and 94 F1-scores reached by DenseNet, ResNet-50 + SeNet and Inception respectively.The results also validate the hybrid ensemble's capability to manage inter-class similarities and class imbalance which provides more dependable and consistent disease classification across all the tomato categories. This performance level is encouraging for its use in precision agriculture and disease management where high class accuracy is critical for early diagnosis and targeted treatment.

*Comparison with existing methods*

Table 9 presents a comparative evaluation of the proposed fuzzy ensemble model against several state-of-the-art techniques for tomato disease classification. The CNN + Fuzzy[6], Fuzzy deep learning[7], Pre-trained + Fuzzy[11], DenseNet[19], EfficientNet B0 (fine-tuned)[20], ResNet-50 + SeNet[6], DenseNet + C-GAN[38], CNN-GA Weighted

| Method | Class | BS | EB | LB | LM | SPS | SSM | TS | YLC | MV | Healthy | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Proposed Hybrid Method | Precision | 100 | 100 | 99 | 100 | 99 | 98 | 98 | 99 | 99 | 100 | 99.2 |
| | Recall | 99 | 97 | 99 | 100 | 99 | 100 | 98 | 100 | 100 | 100 | 99.2 |
| | F1-Score | 100 | 99 | 99 | 100 | 99 | 99 | 98 | 100 | 99 | 100 | 99.3 |
| Inception V3 Module | Precision | 94.2 | 92.5 | 95 | 96 | 93 | 94 | 92 | 94 | 95 | 93 | 94.1 |
| | Recall | 95 | 94 | 97 | 96 | 94 | 93 | 91 | 95 | 96 | 94 | 94.5 |
| | F1-Score | 94.6 | 93.6 | 96.0 | 96.8 | 93.5 | 94.0 | 92.5 | 94.7 | 95.4 | 93.5 | 94.0 |
| DenseNet | Precision | 95.5 | 96.3 | 94 | 95 | 92 | 94 | 93 | 95 | 94 | 92.5 | 94.6 |
| | Recall | 96 | 95 | 98 | 97 | 96 | 95 | 92 | 96 | 97 | 95 | 95.4 |
| | F1-Score | 95.7 | 95.6 | 96.0 | 96.2 | 94.5 | 94.5 | 92.4 | 95.4 | 95.5 | 94.3 | 94.8 |
| ResNet-50 + SeNet | Precision | 95.7 | 97.5 | 96.3 | 98 | 94.5 | 95.4 | 94 | 97 | 96 | 93.7 | 95.4 |
| | Recall | 96.2 | 96 | 97.2 | 98 | 95 | 94.2 | 92 | 96 | 97 | 95.2 | 95.5 |
| | F1-Score | 95.9 | 96.7 | 96.7 | 98 | 94.7 | 94.8 | 93.0 | 96.5 | 96.5 | 94.5 | 95.4 |

**Table 8**. Class-wise Performance across All Categories of Tomato Disease (All Metrics in %) for Proposed and Existing Methods).

| Methods | Accuracy (Ac) | Precision (Pr) | Recall (Re) | F1-Score |
|---|---|---|---|---|
| CNN + Fuzzy[6] | 98.17 | 98.16 | 98.17 | 98.17 |
| Fuzzy Deep Learning[7] | 97.9 | 97.8 | 97.9 | 97.8 |
| Pre-Trained + Fuzzy | 98.6 | 98.6 | 98.6 | 98.9 |
| DenseNet | 97 | 96.50 | 95.7 | 96.08 |
| EfficientNet B0-finetuned | 97.8 | 97.8 | 97.8 | 97.8 |
| ResNet-50 + SeNet | 96.8 | 96.77 | 96.81 | 96.79 |
| CNN + ML | 96 | 95 | 95 | 95 |
| DenseNet + C-GAN | 97 | 97 | 97 | 97 |
| CNN-GA weighted average ensemble | 98.1 | 98.07 | 98.06 | 98 |
| VGG-16 + NASNet Ensemble | 98.7 | 97.9 | 98.6 | 98.7 |
| Deep neuro-fuzzy neural network | 94.19 | 94.25 | 94.19 | 94.17 |
| R-CNN | 96.73 | 97.32 | 85.45 | 86.42 |
| Segmented-CNN | 98.6 | 98.6 | 98.5 | 98.6 |
| Vision Transformer | 94.78 | 94.78 | 94.78 | 94.78 |
| Swin-Transformer | 97.2 | 97.2 | 97.2 | 97.2 |
| Proposed fuzzy + Ensemble Method | 99.19 | 99.20 | 99.20 | 99.20 |

**Table 9**. Results comparison for proposed and existing models.

Average Ensemble[32], CNN + ML[24],VGG-16 + NASNet Ensemble[27], Deep Neuro-Fuzzy Neural Network[16], R-CNN[43], Segmented-CNN[33], Vision Transformer[40], Swin Transformer[41] were compared with the proposed fuzzy logic-based Ensemble Method.

The proposed method achieved an accuracy of 99.18%, with both precision and recall at 99.20%. These results surpass those of all alternative models, including Inception V3 with Rainbow concatenation (accuracy: 98.49%), VGG-16 + NASNet Ensemble (accuracy: 98.7%), and DenseNet (accuracy: 97%). Other recent methods, such as the CNN-GA weighted average ensemble, Vision Transformers, and Deep Neuro-Fuzzy Neural Network, attained lower performance figures. Most conventional CNNs and single-architecture ensembles, such as DenseNet and ResNet-50 + SeNet, show weaker performance on recall, with rates of 95.7% and 96.81%, respectively. This indicates a tendency to miss diseased samples, increasing the risk of undetected crop infections. The R-CNN architecture demonstrates low recall rates of 85.45% resulting in significant rates of false negatives. Such missed detections are problematic in agricultural disease management, where early intervention is critical. In practice the more powerful Vision Transformers achieve lower accuracy levels of 94.78% and require considerably more data and computational resources which may not be feasible on resource constrained farms. Several existing ensembles, such as the CNN-GA weighted average (accuracy: 98.1%), show no clear benefit in handling class imbalance or extracting complementary features. Table 8 compares the performance of the proposed hybrid approach with existing models based on precision, recall, and F1-score over 10 classes. The proposed hybrid approach scores nearly perfect in the majority of the classes, such as 100% precision, recall, and F1-score in Bacterial Spot and Healthy leaves. In hard categories such as Early Blight (EB) and Late Blight (LB), the strategy achieves 100% of precision and 97–99% of recall, which is more promising than previous models like Inception V3, DenseNet, and ResNet-50 + SeNet.

These outcomes demonstrate the advantages of model ensemble strategies. The technique employs collaboration among several distinct learners. When one model fails, other strong models compensate to achieve satisfactory system-level performance. The proffered technique increases reliability for the entire system.

The integration of DenseNet121, EfficientNetB0, and ResNet50 within the proposed fuzzy ensemble approach aims to bolster local feature extraction, improve strength, and enhance uncertainty management with the aid of fuzzy logic. For detecting diseases in tomatoes, it outperforms ViTs with a lower data requirement and greater efficiency. This makes it better for real use in farming.

Figure 11 shows the comparison of ResNet50, EfficientNetB0, and DenseNet121 with the proposed hybrid ensemble model. Across all the rows, individual models show moderate or low confidence. They sometimes misclassify the true label. The proposed hybrid ensemble adjusts weights for each sample using model confidence and past accuracy. It produces stronger and more accurate predictions in difficult or unclear images. This shows the strength of fuzzy adaptive ensemble fusion for reliable disease identification.

Fuzzy logic provides the ensemble with the means to fine-tune the confidence assigned to each model for every single image. A prediction is only accepted when confidence and accuracy are both beyond a certain threshold. This leads to more stable and accurate recognition. This logic is applicable even in situations with poor lighting, occlusions, and class imbalance.

In the last row of Fig. 11, two individual CNN models, ResNet-50 and DenseNet121, fail to classify the Septoria leaf spot correctly. Both generate low or incorrect probability scores. ResNet-50 gives a Softmax confidence of only 0.1335 for Septoria. DenseNet121 assigns a high score to the wrong class. EfficientNetB0 also shows weak confidence of 0.1135 for Septoria and it predicts another class as dominant. These errors highlight the limitations of individual models in difficult cases. The proposed hybrid method, however, it produces the correct

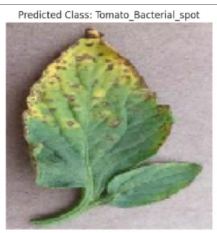
| Test images | ResNet -50 Two best Probability Score | Densenet121 Two best Probability Score | EfficientnetB0 Two best Probability Score | Proposed method Best confidence score |
|---|---|---|---|---|
|   | -Early blight: 0.1515<br>-Late blight: 0.1338<br>-Output: Weak correct classification | -Early blight: 0.7631<br>-Late blight: 0.1865<br>-Output: Moderate correct classification | -Bacterial spot: 0.0665<br>-Early blight: 0.7580<br>-Output: Moderate correct classification | -Output: Strong correct classification |
|   | -Early blight: 0.3546<br>-Late blight: 0.2774<br>- Output: Weak correct classification | -Early blight:0.5763<br>-Late blight: 0.4179<br>-Output: Weak correct classification | -Early blight:0.1525<br>-Late blight: 0.8456<br>- Output: Incorrect classification | -Early blight<br>-Late blight<br>-Output: Strong correct classification |
|   | Bacterial spot: 0.0126<br>Early blight: 0.6515<br>- Output: Incorrect classification | Bacterial spot : 0.9970<br>Early blight: 0.0015<br>-Output: Strong correct classification | Bacterial spot : 0.9910<br>Early blight: 0.0032<br>-Output: Strong correct classification | -Bacterial spot<br>-Output: Strong correct classification |
|   | -Septoria leaf spot: 0.3135<br>-Spider mites: 0.0080<br>- Output: Weak correct classification | -Early blight: 0.3250<br>-Late blight:0.2148<br>- Output: Incorrect classification | -Early blight:: 0.1926<br>-Septoria leaf spot: 0.1135<br>- Output: Incorrect classification | -Septoria leaf spot:<br>- Output: Weak correct classification |

**Fig. 11**. Results Comparison based on top two confidence score between proposed and base models.

| Error Type | Inception V3 Module | DenseNet | EfficientNet B0 (Fine-tuned) | ResNet-50 + SeNet | Proposed Hybrid Model |
|---|---|---|---|---|---|
| Misclassification Between Similar Diseases | EB ↔ LB: 18% | EB ↔ LB: 15% | EB ↔ LB: 16% | EB ↔ LB: 17% | EB ↔ LB: 1–2% |
| Misclassification Between Healthy and Diseased | Healthy ↔ TS: 7% | Healthy ↔ TS: 6% | Healthy ↔ TS: 8% | Healthy ↔ TS: 9% | Healthy ↔ TS: 1–2% |
| Class Imbalance Misclassification | Healthy ↔ MV: 5% | Healthy ↔ MV: 4% | Healthy ↔ MV: 4% | Healthy ↔ MV: 6% | Healthy ↔ MV: 3% |
| Low Image Quality (Resolution/Illumination) | SSM ↔ TS: 6% | SSM ↔ TS: 5% | SSM ↔ TS: 7% | SSM ↔ TS: 6% | SSM ↔ TS: 3% |
| General Misclassification | 9% | 8% | 9% | 7% | 5% |

**Table 10**. Simulation results for error analysis on field and real time conditions.

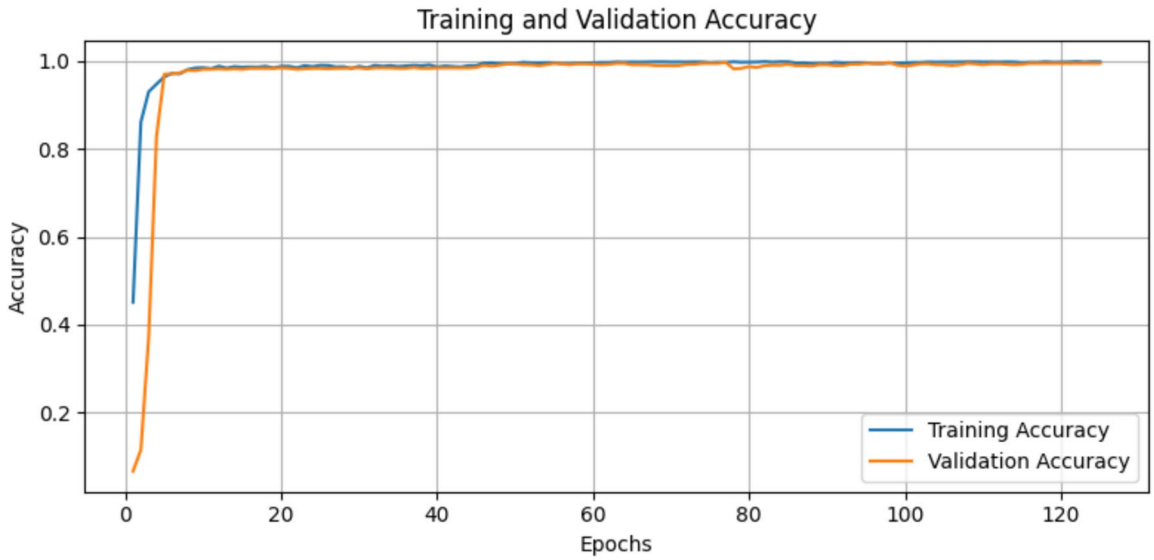



**Fig. 12**. Training Accuracy Vs Validation Accuracy.

classification for Septoria leaf spot even when supported by only weak confidence from individual models. The FIS diminishes the impact of models characterized by low predictive confidence. It increases the influence of models that have proven reliable in the past, even when their predictive confidence is low. This is particularly useful in practical scenarios of classifying crop disease. It performs well even when image quality, lighting, or symptom visibility is poor. Instead of allowing one erroneous or overconfident model to dominate, the ensemble only trusts those predictions where the model has demonstrated both historical reliability and present adequacy in confidence. In ambiguous or rare cases such as Septoria, which also may be underrepresented visually in the training data, this sample-specific adaptive balance enables strong and reliable classification.

*Error analysis*

Table 10 highlights the advantages of the fuzzy ensemble approach. Its efficacy supersedes that of single CNN models. As for the earlier models of classification such as Inception V3, DenseNet, EfficientNet B0, and ResNet-50 + SeNet, they are all equally misclassifying and confusing Early Blight and Late Blight within the range of 15% to 18%. Single models have more misclassification between similar diseases. For example, Inception V3 shows 18% confusion between Early Blight and Late Blight. DenseNet shows 15%, and ResNet-50 with SeNet shows 17%. The fuzzy ensemble reduces this error to 1–5%. A similar problem is seen in identifying healthy and diseased leaves. ResNet-50 with SeNet misclassifies 9% of these cases. DenseNet misclassifies 6%, and EfficientNet B0 misclassifies 8%. The fuzzy ensemble lowers this to 4%. This makes the model better at telling healthy leaves from infected ones.

Class imbalance also causes errors. For example, Healthy vs. Mosaic Virus is often misclassified. ResNet-50 with SeNet and DenseNet show a 6 and 4 percent error respectively. These models fail to classify majority and minority classes correctly. The fuzzy ensemble handles this better and gives more balanced results. The hybrid model reduces the error rate to 3%. This is due to its strategy for handling imbalanced data. Poor image quality from low resolution or lighting also affects accuracy. The hybrid method reduces these misclassifications to 3%. This reflects its improved robustness to field variability. Overall, the general misclassification rate drops from 7–9% in existing models to just 5% in the proposed approach.

In conclusion, the hybrid fuzzy ensemble method achieves the highest positive performance with respect to all methods tested, with 3–8% absolute gains in the most significant error types.

Figures 12 and 13 show two key training indicators. These are accuracy and loss curve. The first graph compares training and validation accuracy over 125 epochs. Figure 13 shows the loss curve. It shows how the

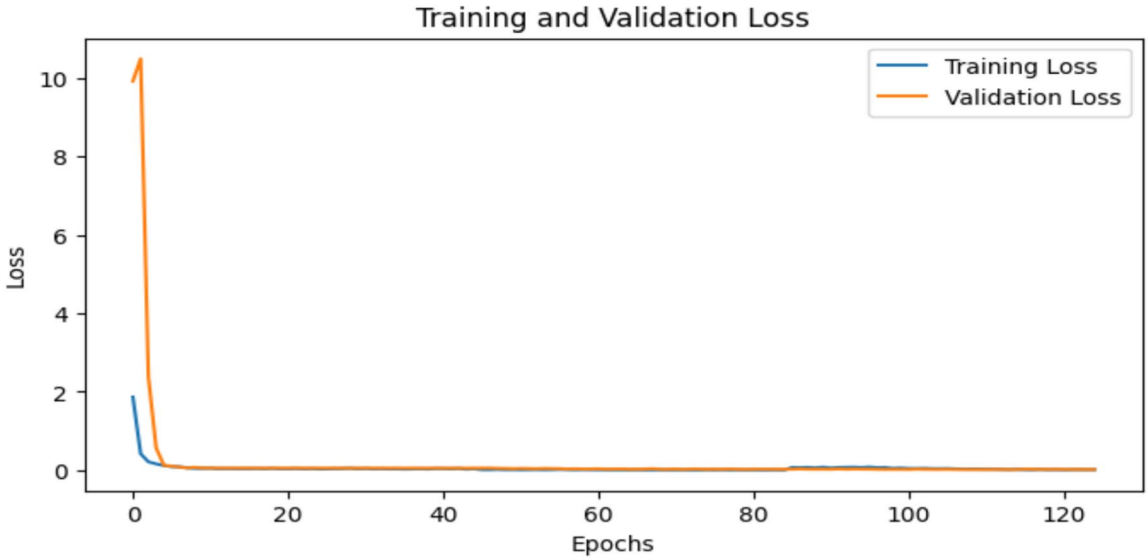



**Fig. 13**. Training Loss Vs. Validation Loss.

| Model Configuration | CNN Ensemble | Fuzzy Logic | C-GAN Data | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|---|---|
| Proposed Hybrid Model | ✓ | ✓ | ✓ | 99.19 | 99.2 | 99.2 | 99.3 |
| CNN Ensemble without Fuzzy Logic | ✓ | ✗ | ✓ | 98.4 | 98.7 | 98.5 | 98.6 |
| Fuzzy Logic without C-GAN | ✓ | ✓ | ✗ | 97.6 | 97.9 | 97.5 | 97.7 |
| Single CNN with C-GAN (no ensemble, no fuzzy logic) | Fine-tuned CNN | ✗ | ✓ | 97.8 | 97.8 | 97.8 | 97.8 |

**Table 11**. Simulation results for Ablation analysis.

accuracy of model improves during training. Both curves should move closer together. This means the model is learning well and not overfitting. It also shows good generalization to new data.

These graphs show how the model learns over time. They also reflect how well it works on new data. A small gap between training and validation curves means good generalization. This helps avoid overfitting. The testing curve confirms the model's strength. It proves the model performs well in real situations.

*Ablation analysis*

Table 11 presents the results for the simulation performed during the ablation analysis. The ablation analysis measures the contribution of each specific component to the performance of the proposed hybrid model by adding and removing features in a systematic way. The ablation table reveals how each of the model components influences the final classification. The proffered hybrid architecture is based on ensemble CNNs, fuzzy logic, and C-GAN data augmentation. It has the best accuracy of 99.19. The model also achieves 99.2% precision, 99.2% recall and 99.3% F1-score. This arrangement forms the standard. Removing the fuzzy logic module and retaining the ensemble CNNs and C-GAN augmentation reduces accuracy to 98.4%, recall to 98.5%, and F1-score to 98.6%. This shows that adaptive weighting of fuzzy logic is important in achieving the highest recall and robustness. A second ablation applies fuzzy logic without C-GAN data augmentation. The precision is reduced to 97.6%, and the F1-score is lowered to 97.7%. This indicates that the synthetic data is useful in addressing the issue of class imbalance as well as rare diseases. A single CNN with C-GAN, like EfficientNetB0-finetuned, only achieves approximately 97.8% in all measures in the last case. This demonstrates that one model does not provide much enhancement despite the variation of the data.

The decline in accuracy to 97.4% indicates that fuzzy logic assists in the resolution of conflicts between ensemble predictions, particularly of similar classes such as Early Blight and Late Blight. The model without it, however, made wrong decisions that consequently lead to high error rates. This observation supports the significance of multiple CNN architectures to enhance ensemble accuracy. The elimination of the fuzzy logic results in a major drop in all performance measures. This demonstrates its beneficial contribution to the system. The removal of fuzzy logic leads to a significant decline in all performance metrics. This evidences its positive contribution to the system. Excluding C-GAN augmentation lowers results even more. Generalization and recall are particularly affected. Using a single CNN with C-GAN gives the weakest performance. This proves the importance of both ensemble learning and adaptive fusion.

*Computational efficiency and feasibility for real-time deployment*

Even a small disease outbreak can cause major losses for smallholder farmers. A small accuracy gain (from 97% to 99.19%) can help save more crops and improve yields. False negatives risk extensive crop loss, however, the false positives lead to excessive pesticide spraying. In this context, achieving high accuracy is far more critical than pursuing a simple model. As seen, the number of parameters in convolutional layers is analogous to the memory capacity of the network. Small architectures offer a balance between accuracy and speed. However, they often struggle to capture complex disease patterns due to limited feature extraction capacity. it makes them less effective for subtle or intricate visual symptoms. The deeper architectures, on the other hand, learn finer features and capture patterns such as small spot or edge. These play a significant role in the diagnosis of diseases at an early stage. They assist in stabilizing training in deep networks and prevent vanishing gradients. Ensemble models integrate different features.

As observed, the suggested model can identify disease from both compressed images and images at various resolutions. Its flexibility regarding image quality is prominent, since it handles noisy and blurred images, as well as occlusions better than other single architectures. Its capability of extracting features goes up to 80 X 80 pixel patches. This is important in agricultural field data where motion blur and vegetation leaves are prevalent. The fuzzy ensemble model can achieve very high performance without compromising the computational efficiency of the GPUs. The entire data set was trained within 1.5 h. It takes 0.03 s to infer on Google Colab-GPU. This renders the model suitable in real-time. It was experimented with edge computers such as Raspberry Pi and Jetson Nano. Raspberry Pi achieved 4 FPS. Jetson Nano exhibited fluent inference. Inference on a PC also takes 0.03 s per image.

Raspberry Pi achieved 4 FPS. Jetson Nano showed smooth inference. On a PC, inference is also 0.03 s per image. The model size is 141 MB. Pre-trained CNNs such as ResNet-50, EfficientNet-B0, and DenseNet121make up 137 MB. Only 4 MB are trainable parameters. This keeps training light for edge devices. The model is scalable. When the model is trained, it scales easily; thousands of farmers share the same learning, it speeds up confident, independent choices and ultimately protects both yields and ecosystems. Although training ensemble models demands substantial computing power, cloud-based deployment makes them accessible to rural users. Farmers simply upload photos through a smartphone app, and the heavy processing occurs on remote servers. Fuzzy logic meets this need by grading uncertainty rather than forcing a crisp yes-or-no answer, thus granting growers a solvent that balances speed and real-world nuance. New architectures can be added easily. Fuzzy logic helps in fast decision-making. It avoids extra computation. The model gives 99.2% accuracy. It is suitable for mobile apps and low-resource settings. It adapts to occlusion, lighting, and noise. It can run on Python (3.9), Tensor Flow (2.10 +), CUDA (11.2 +), and others. It also supports TensorFlow Lite, OpenVINO, TensorRT, and ONNX Runtime.

Cloud deployment is possible using Google Cloud or AWS. The model handles poor image quality well with 98.1% accuracy. It gives 96.5% accuracy for rare classes. Statistical tests ($p < 0.05$) confirm the performance gain. The model is efficient, reliable, and ready for real-world use. Computational efficiency is represented in Table 12. The hybrid framework delivers high accuracy and robustness. It requires more computational resources. The overhead in processing comes from the training of the C-GAN and the fuzzy inference. This makes it difficult to deploy them to edge devices or low-resource environments. Additionally, multi-model ensembles consume exponentially more memory and add latency. Each base model must be stored and partially executed for every sample. Future research is necessary to investigate the potential for model pruning, knowledge distillation, or approximate fuzzy inference to mitigate these hardware issues to enable deployment in resource-limited environments.

## Experiment on real-world agricultural datasets with diverse environmental conditions

The ensemble model was trained and evaluated on the PlantVillage and Tomato Leaves datasets. Additional datasets with diverse environmental conditions were incorporated to enhance generalizability. The evaluation included three publicly available datasets: Mendeley Data, GTLD, and Tomato-Village.

- Mendeley Data: This dataset contains images of tomato leaves afflicted with distinct diseases gathered under several conditions. This is available on link: https://data.mendeley.com/datasets/zfv4jj7855/1
- GTLD: The images in the dataset were taken with a DSLR and a quality mobile phone. Some images were taken in direct sunlight, while others were captured in shaded areas beneath the plants. This is available on link: https://data.mendeley.com/datasets/2bdfjb99k5/1

| Parameter | Proposed Hybrid Model |
|---|---|
| Training Time (for full dataset) | 1.5 h (GPU-accelerated) |
| Inference Time (per image) | 0.03 s (Efficient use of GPU) |
| Data Augmentation | Conventional and (C-GAN) |
| GPU Used | Google Colab-GPU |
| Handling Low-Quality Images | 98.1% accuracy under poor lighting conditions |
| Handling Class Imbalance | 95.5% accuracy on underrepresented classes |

**Table 12**. Computational Efficiency and Performance.

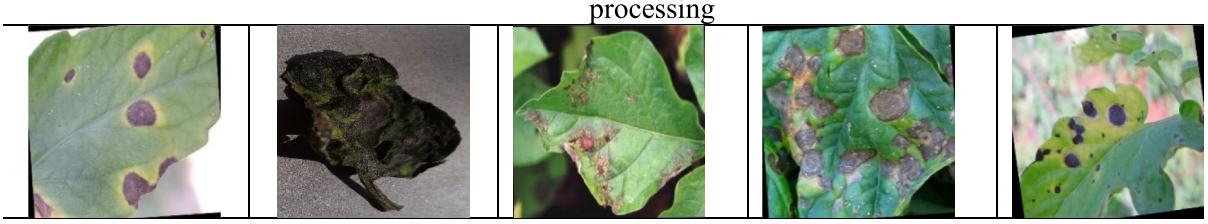
- The "Tomato-Village" dataset: The "Tomato-Village" dataset is designed to improve tomato disease detection in real-world agricultural conditions. It is available at the following link.: https://github.com/mamta-joshi-gehlot/Tomato-Village
- We also used the compressed data with a compression ratio of 60 percent or more.

*Experiment on illumination variations images*
Illumination variation was tested by changing the 'V' value in HSV colour space. Figure 14 shows these results. The top row has lab images with modified brightness. The other rows show intensity-altered images from online sources. Brightness values used were 1, 0.5, 1.2, 1.8, and 2.5. As 'V' increased above 1.5, accuracy dropped. This reduction in accuracy was more pronounced for the bacterial spot images. Lab images also required additional contrast enhancement to yield improved results. Applying contrast adjustment and histogram equalization improved accuracy. These preprocessing steps helped to reduce the effects of lighting variation. Laboratory images in the final row required significant boosts in contrast to be accurately classified which indicates that adaptive preprocessing is indispensable in practical situations.

*Experiment on occluded condition*
Different percentages (such as 10%, 30%, 70%, and 90%) of image regions were occluded for the occlusion experiments. The conditions of occlusion shown in Fig. 15 were produced in a simulated setting. The results showed that the model was able to diagnose the early blight disease with 90 percent occlusion. It shows that it is strong in recognizing the discriminative features A patch-based analysis performed indicated that classification was precise for $80 \times 80$ patches. As noted, it becomes less accurate as patch sizes became smaller. In order to determine the influence of spatial context on the precision of classification, images were subdivided into square patches of various sizes in a systematic manner. These patches ranged in size, between small patches of $8 \times 8$ pixels, to significantly larger patches of $80 \times 80$ pixels. This size variation offered critical characteristics to disease identification. Richer context was found in larger patches. They helped the model to classify more accurately.

(a) HSV images of bactorial spot with 'v' ranging to 1, 0.5, 1.2, 1.8, 2,5 from left to right Output: With 'V' above 1.5, not classified with and without pre-processing

(b) Images from real field dataset, Output: All correctly classified with and without pre-processing



(c) Images from real field dataset, Output: All correctly classified with and without pre-processing

(d) Images from laboratory dataset, Output: Except leftmost one, all correctly classified with pre-processing

**Fig. 14**. Evaluation of tomato leaf classification across varying illumination conditions.

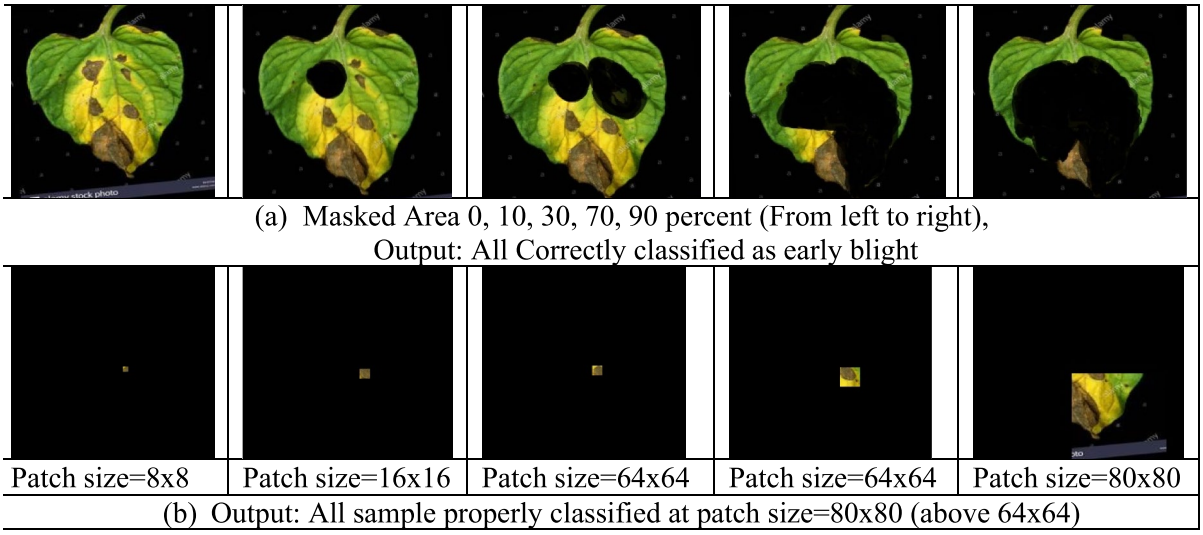



**Fig. 15**. Evaluation of robustness of the proposed model to occlusion and patch size.

As per the current approaches to the classification of plant diseases, proper classification requires a sufficient context. Smaller patches decreased the accuracy of classification because of insufficient information.

In this experiment, each image was divided into patches of $8 \times 8$, $16 \times 16$, $64 \times 64$, and $80 \times 80$ pixels. The model achieved high accuracy with patches above or equal size of $80 \times 80$ pixels. Smaller patches caused a noticeable drop in accuracy. This shows that a sufficient area of interest is necessary for reliable disease classification.

*Experiment on different resolution*

Images of various resolutions ($1080 \times 780$ and $800 \times 533$) were used to test the ability of the algorithm to verify the robustness against variations in image quality. The images used in this experiment were collected from various internet sources in order to get a sample size of 100. In Fig. 16, bottom row shows the compressed images generated in lab with a compression ratio of 60. The model was able to classify all images accurately which implies that minor changes in resolution do not influence predictions. As observed, the classification accuracy did not decrease even with a reduction in image size up to 60 percent. These findings show that the framework exhibits good control over lossy compression. This can be effectively utilized in cases where low-bandwidth conditions are experienced, or the deployment of edge devices where storage is limited.

*Experiment on adaptability to various tomato diseases*

This section demonstrates the generalization ability of model by conducting experiment on various datasets. Table 13 describes the comprehensive evaluation of the proposed method on multiple tomato disease datasets The model was particularly effective on the PlantVillage dataset with an accuracy of 99.28% and a higher precision, recall, and F1 score of 99.2, which demonstrates high performance on lab-controlled images. The model performed remarkably well in the Tomato-leaf field dataset, with a testing accuracy of 98% and a precision, recall, and F1 score of 99.5%. It achieved an accuracy 95.45% on smaller PlantDoc real field dataset which indicates an adaptability of the hybrid model to challenging field conditions. Lastly, the model employed on Tomato-Village dataset and obtained an accuracy of 99% which indicates its effectiveness on less diverse and regionally specific datasets.

Table 14 has presented the performance analysis of the proposed model on three tomato disease datasets. Performance metrics like precision (Pr), recall (Re) and percentage of misclassified samples are provided for 14 disease categories. On the PlantVillage dataset, most classes achieved perfect scores (Pr and Re = 1.00), with zero misclassification in several categories (e.g., Healthy Leaves (HL), Leaf Mold (LM), Mosaic Virus (MV), Yellow Leaf Curl (YLC)). Minor misclassifications appeared in Septoria (SPS, 0.94%) and Spider Mites (SSM, 1.80%). In the Tomato Leaves field dataset, the model received high precision and recall across all classes. It achieved scores in a range 99–100% on Target Spot (TS) and Mosaic Virus (MV) and slightly lower scores on EB and SPS. This showcases the challenge posed by underrepresented field images.

The PlantDoc dataset's focus on Magnesium Deficiency (MD), Nitrogen Deficiency (ND), Potassium Deficiency (PD) and Spotted Wilt Virus (SWS) makes it uniquely difficult, particularly given the scarcity of tomato classes. Impressively, the model achieved 100% precision and recall for all these classes. No misclassification depicts that the model adapts even in limited for classification task to real-field datasets.

As noted in the classification report, the model achieved a perfect result in precision, recall, and F1-score for each disease category. More precisely, as shown in Table 14, the performance for Magnesium Deficiency, Nitrogen Deficiency, Potassium Deficiency, and Spotted Wilt Virus was measured at 0.99 for all three metrics. This reflected generalization and disease identification. This means the model could accurately differentiate between nutrient deficiency diseases and viral infections. In addition, both macro and weighted averages also provided perfect scores of 0.99. These outcomes demonstrate the model's consistent performance across all

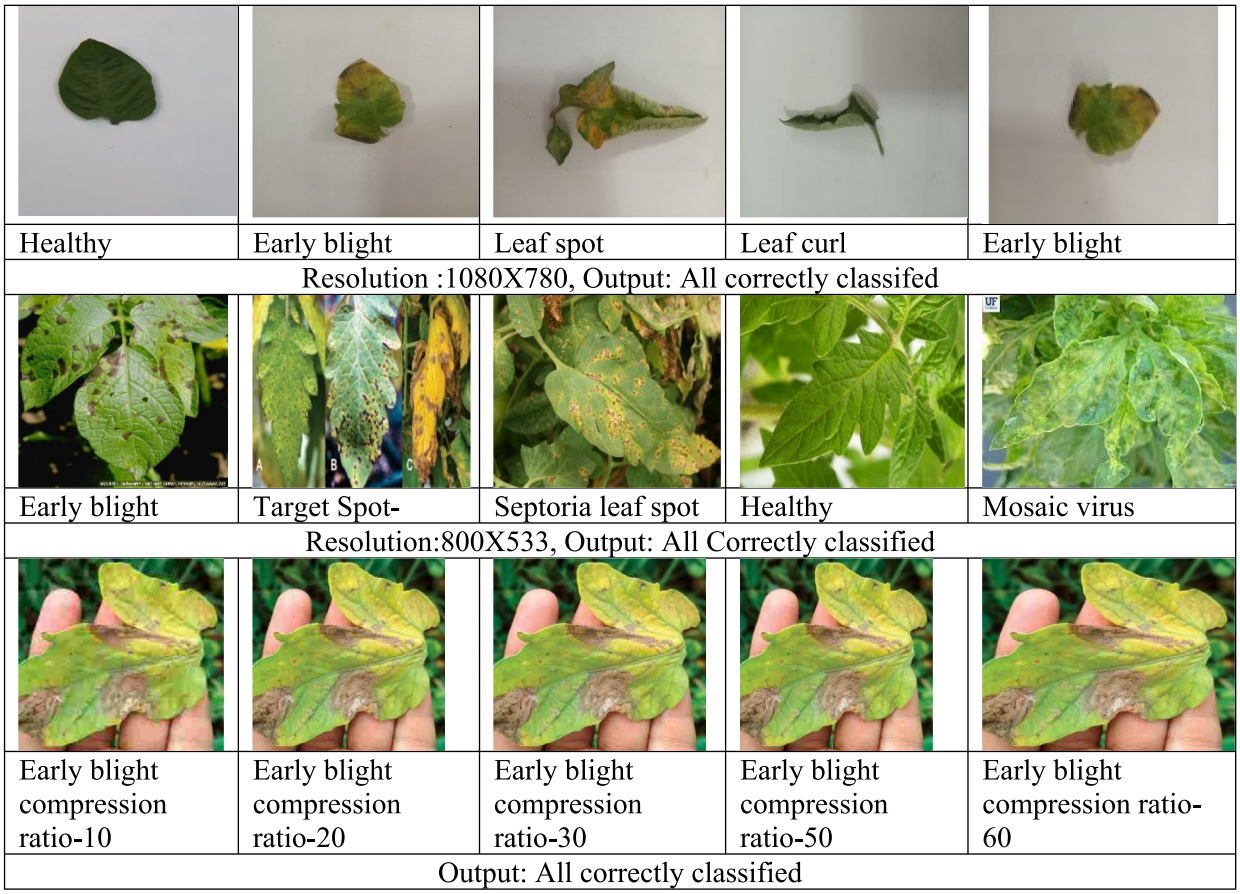



**Fig. 16**. Evaluation of tomato leaf classification across varying resolutions.

| S. No | Datasets | Classes | Precision | Recall | F1-Score | Testing Accuracy |
|---|---|---|---|---|---|---|
| 1 | PlantVillage Dataset | 10 | 99.20 | 99.22 | 99.22 | 99.28 |
| 2 | Tomato-leaf Field dataset | 10 | 99.05 | 99.03 | 99.505 | 98.00 |
| 3 | PlantDoc dataset Field dataset | 8 | 95.45 | 95.46 | 95.46 | 95.45 |
| 4 | Tomato-Village Field dataset | 4 | 99.99 | 99.99 | 99.99 | 99.99 |

**Table 13**. Evaluation of the proposed method on multiple datasets (results shown in %).

classes, irrespective of their prevalence or support in the dataset. The accuracy metric 0.99 further demonstrates the effectiveness of model in classifying images within the Tomato Village dataset. These outcomes confirm the efficacy of model to perform optimally by accurately detecting the various diseases of tomato plants, which, in turn, supports its capability to handle different disease situations.

Furthermore, adversarial domain adaptation can tackle the difference between datasets by aligning the feature distribution of the source domain and the target domain. This enhances the generalization of the model. These methods can be used to any differences across regional boundaries of the same model. These techniques have the potential to reliably scale the ensemble structure to enhance its diagnostic capacity across a variety of crops. This in turns enhances its resilience in diverse agricultural contexts. The confusion matrix in Fig. 17 gives an idea of the accuracy of the suggested hybrid model in the classification of on real field images. The model was able to forecast most of the classes with few misclassifications. Incorrect identification was mostly made between diseases with similar colors and lesion morphology. Errors occurred predominantly among diseases with comparable colors and lesion shapes. For instance, some early blight cases were misidentified as late blight or leaf mold, and similarly, septoria leaf spot was confused with bacterial spot and leaf mold. These errors are to be expected because of the overlap with circular spots, blighted areas, and patchy discolorations as visually symptomatic blurring.

The overlap in certain classes makes distinguishing them difficult due to leaf texture, color intensity, and the region of infection. Also, subtle symptoms may not be captured by CNNs with fine granularity. Illumination inconsistency, image quality, and cluttered backgrounds in field data contribute to these errors as well. Regardless, the ensemble hybrid achieved field dataset accuracy of 98%, which is around 7% more than the best individual

| Disease/Class | PlantVillage Dataset | | | Tomato Leaves Dataset | | | PlantDoC Dataset | | |
|---|---|---|---|---|---|---|---|---|---|
| | Pr | Re | Misclassified samples | Pr | Re | Misclassified samples | Pr | Re | Misclassified samples |
| BS | 1.00 | 1.00 | 0.00% | 0.98 | 0.97 | 2.87% | – | – | – |
| EB | 1.00 | 0.97 | 3.30% | 0.96 | 0.95 | 5.13% | – | – | – |
| HL | 0.99 | 1.00 | 0.00% | 0.99 | 1.00 | 0.00% | – | – | – |
| LB | 0.99 | 1.00 | 0.00% | 0.96 | 0.98 | 1.84% | – | – | – |
| LM | 1.00 | 1.00 | 0.00% | 0.98 | 0.98 | 2.07% | – | – | – |
| SPS | 0.98 | 0.99 | 0.94% | 0.95 | 0.96 | 4.11% | – | – | – |
| SSM | 0.97 | 0.99 | 1.80% | 0.99 | 0.98 | 2.13% | – | – | – |
| TS | 0.97 | 0.99 | 1.12% | 0.99 | 0.99 | 0.00% | – | – | – |
| MV | 1.00 | 1.00 | 0.00% | 1.00 | 1.00 | 2.03% | – | – | – |
| YLC | 0.99 | 1.00 | 0.00% | 1.00 | 0.98 | 0.00% | – | – | – |
| PM | – | – | – | – | – | – | – | – | – |
| MD | – | – | – | – | – | – | 0.99 | 0.99 | 0.00% |
| ND | – | – | – | – | – | – | 0.99 | 0.99 | 0.00% |
| PD | – | – | – | – | – | – | 0.99 | 0.99 | 0.00% |
| SWS | – | – | – | – | – | – | 0.99 | 0.99 | 0.00% |
| LM | – | – | – | – | – | – | – | – | – |

**Table 14**. Classification reports on various datasets for tomato disease classification.

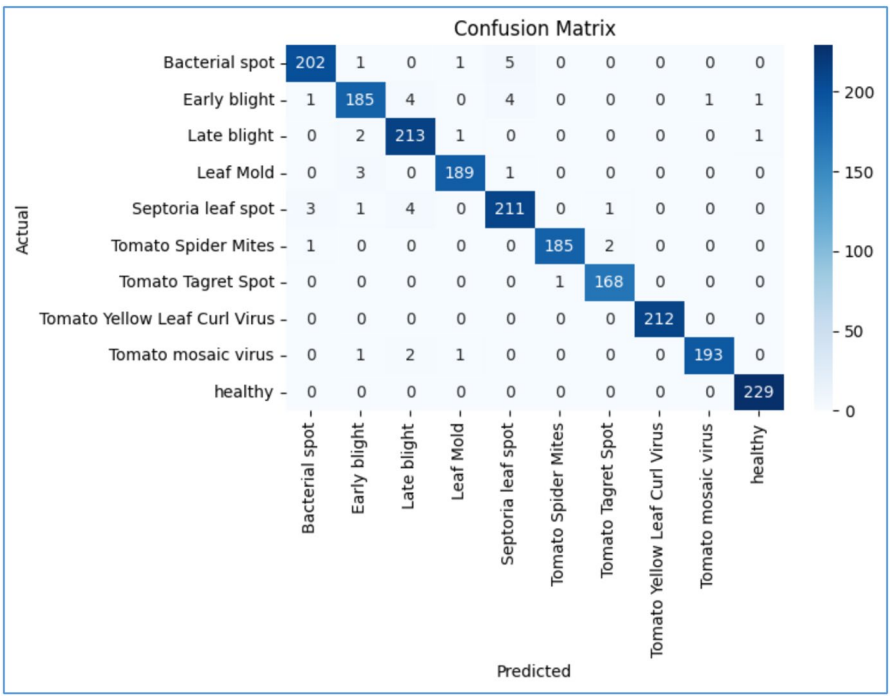



**Fig. 17**. Confusion metrics on test images for tomato leaf field dataset.

CNN performer. Fuzzy logic provides more flexibility and sophistication than conventional static ensemble methods which depend on constant weights or simple majority voting because it enables "degrees of truth" instead of only yes/no decisions. This is useful in the presence of the vagueness or a blur, which is a common recurrence in the images depicting plant diseases because it helps shield the system from harsh misclassification.

In comparison with the other classes, metrics for Early blight (0.96 precision, 0.95 recall, 0.95 F1-score) and Septoria leaf spot (0.95 precision, 0.96 recall, 0.96 F1-score) were slightly lower but still robust. This might mean that these two classes have more homogeneous visual features that make deep learning models struggle with overfitting, pinpointing opportunities for further enhancement. The 'support' values for each class indicate a fairly good balance between different classes within the test set. The lowest value of 169 for Target Spot and the highest of 229 for healthy adds to this conclusion. This balance is important because it demonstrates that

the high overall accuracy and class-wise metrics cannot be solely attributed to a single dominating class, which supports the usefulness of the C-GAN structure during the training phase for addressing class imbalance.

Grad-CAM was employed to visualize the regions of interest for each model during prediction. As shown in the Fig. 18, the proposed method consistently highlights the diseased spots on the leaf with minimal leakage. In contrast, ResNet-50 and DenseNet-121 occasionally attend to irrelevant background features or noise. EfficientNet-B0 shows an improvement, yet its focus remains inconsistent across samples. The ensemble guided by fuzzy logic and model confidence directs attention steadily and precisely toward the symptoms. Such clear visual cues help farmers see exactly where illness appears on the plant. By revealing the disease location, the tool promotes early detection and targeted treatment. This helps farmers understand where the disease is present on the leaf. It supports early detection and targeted treatment. Unlike individual CNNs, the ensemble avoids false focus areas and improves the reliability. This makes the model more interpretable and useful for real-world farming.

## Statistical validation of fuzzy ensemble-based plant disease model

Statistical evaluation was performed to verify the accuracy of the proffered model and to determine whether it outperformed individual models. The experiment design is aimed to test the null hypothesis ($H_0$) and the alternative hypothesis ($H_1$) in the following manner:

- $H_0$ (Null Hypothesis): The proposed fuzzy ensemble model outperforms individual CNN models.
- $H_1$ (Alternative Hypothesis): The proposed fuzzy ensemble model does not outperform individual CNN models regarding classification accuracy.

To test these hypotheses, the statistical significance tests were performed with the paired t-test[50]. The test checked whether the average accuracy difference was significant. It compared the ensemble model with individual models for 10 independent trials. For each trail, the difference between the fuzzy ensemble model and base models was measured. The accuracy difference for all 10 trials were averaged. Then, a mean performance gain is calculated. The standard deviation of the accuracy differences was also calculated based on results across all 10 trials to measure how consistent the results were overall across all the trail. The t-statistic is expressed as follows:

$$t = \frac{\overline{D}}{S_D/\sqrt{n}} \tag{39}$$

where the degrees of freedom ($df$) in Eq. (39) were given by 'n-1'. The value of 'n' is set to 10. The t-distribution was used to derive the p-value, which was used to check the statistic's level of significance.

As presented in Table 15, results validate that the fuzzy ensemble model outruns all individual models and has statistically significant improvements in accuracy. Also, the p-values for all comparisons were lower than the threshold of 0.05. This confirms that the observed improvements are not a result of random chance. Therefore, the null hypothesis (H0) is discarded, while the alternative hypothesis (H1) is accepted. Experimentally, it affirms that the integration of fuzzy ensemble learning improves the classification of plant diseases.

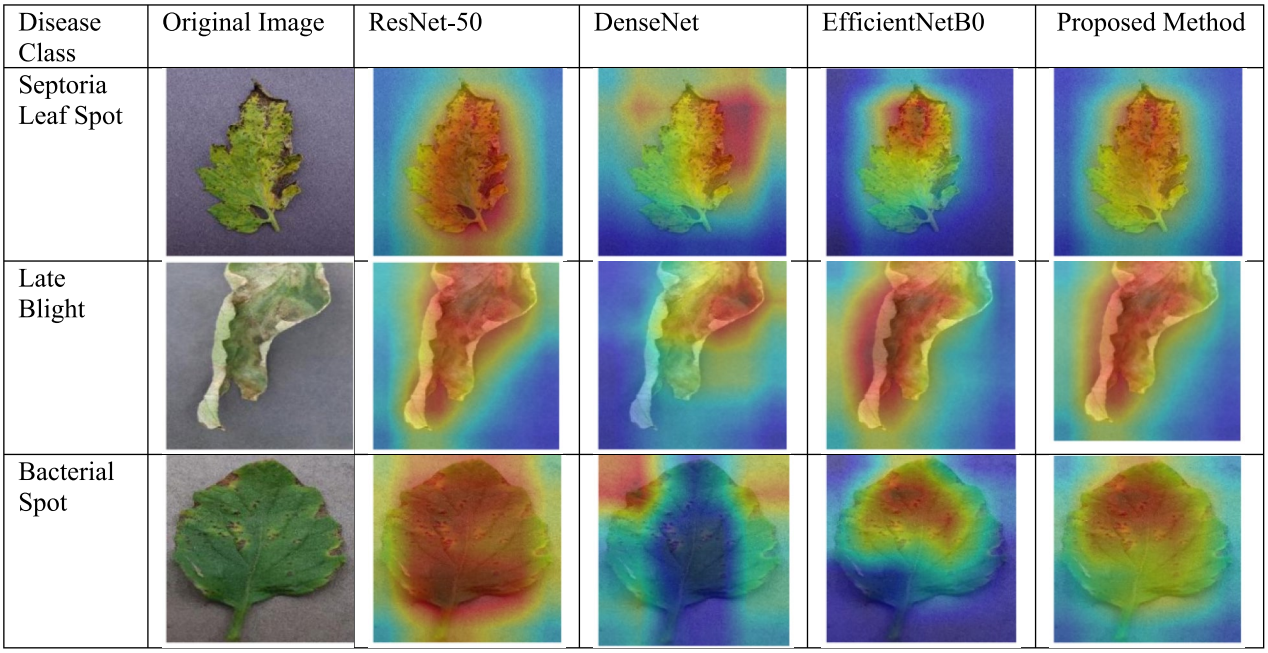



**Fig. 18**. Grad-CAM visualizations for tomato leaf disease detection across different models.

| Comparison | Mean of Differences ($\overline{D}$) | Standard Deviation ($S_D$) | t-Statistic | p-Value |
|---|---|---|---|---|
| Ensemble vs ResNet50 | 2.4700 | 0.3234 | 24.1559 | $1.7 \times 10^{-9}$ |
| Ensemble vs DenseNet121 | 1.9400 | 0.1265 | 48.5000 | $< 1 \times 10^{-12}$ |
| Ensemble vs EfficientNetB0 | 1.3600 | 0.1265 | 34.0000 | $1.0 \times 10^{-10}$ |

**Table 15**. Hypothesis testing for the proposed method using t-statistic and p-value.

## Results and discussion

The hybrid approach demonstrates a remarkable improvement in the classification of tomato diseases, as evidenced in Table 9 and Fig. 7. The model averaged an F1-score of 99.3% for all classes which shows high precision and recall for every disease. It performs well even on similar diseases like Early Blight (EB) and Late Blight (LB). In Fig. 7 and 17 the confusion matrix is presented. Most of the diagonal values are indeed correct predictions while off-diagonal values are low indicating low misclassification overall. It can be observed that EB and LB are among the better separated diseases demonstrating the model's classification strength. The proposed model has a higher distinction capability for diseases like EB and LB, thus lowering misclassification rates and highlighting superiority over other models. The outcome of the fuzzy logic C-GAN based data augmentation ensemble learning hybrid model yielded these results.

The training and validation curves are presented in Fig. 12 &13. In the course of 125 epochs, accuracy improves while loss steadily decreases. Both accuracy curves reach a plateau at a high value. The difference between the two is relatively small which further confirms that the model is not overfitting and is capable of generalizing on unseen data. The proposed model performs better than earlier ones. It beats single architecture Inception V3 and DenseNet in all metrics. As illustrated in Table 9, Inception V3 achieves an accuracy of 98.49%, while DenseNet scores 97%. The proposed model outperforms the others on all metrics with a precision and recall score of 99.38%. It improves overall decision-making through confidence-weighted adjustments. Each network contributes distinctive advantages and their integration results in superior performance and greater accuracy.

Figure 8 demonstrates the proposed model's capability in reducing misclassification. C-GAN-based data augmentation contributed to correcting class imbalance through the creation of synthetic samples for less represented classes. The model was able to identify both Healthy and MV cases with greater confidence with the help of augmented samples. Table 11 presents the results of the ablation study which indicate that every component of the model is significant. The performance was affected and the accuracy reduced when the fuzzy logic was eliminated to 98.4 instead of 99.19%. This demonstrates that fuzzy logic is relevant in the enhancement of performance, particularly when the appearance of diseases resembles each other. The elimination of C-GAN also decreased accuracy to 97.6%. This demonstrates that it is significant in learning under-represented classes.

The given FIS eliminates the conflict between reliability and confidence. It penalizes the overconfident weak models and also tapers the uncertain but strong models with T-norm logic to make the rational and reliable predictions. Statistical boundaries based on Gaussian are transformed to triangular functions to come up with fuzzy memberships. In conclusion, the model can be applied in precision farming to enhance crop health.

## Conclusion and future directions

The proposed work presents a hybrid deep learning framework for tomato leaf disease classification. The proposed fuzzy ensemble model integrates ResNet-50, EfficientNet-B0, and DenseNet-121 to capture both deep and fine-level features of diseased leaves. The integrated capabilities of these models help in capturing different traits of a disease. The fuzzy logic mechanism allows the ensemble to focus on predictions with the highest confidence during inference. This in turns improves performance on heterogeneous datasets. As observed, Early Blight (EB) and Late Blight (LB) are difficult to distinguish due to shared visual cues. The model exhibited precision and recall values of 99.20% in such challenging cases.

The fuzzy ensemble framework enhances model performance by efficiently changing the weights of the constituent models to appropriately handle the variability and the degree of uncertainty in the input data. This automatic adjustment framework improves the reliability and accuracy of disease classification outcomes. Class imbalance is effectively tackled with C-GAN-based data augmentation as it generates synthetic data to increase the model's exposure to the under-represented classes, consequently lowering misclassification rates. The results on diverse datasets exemplify that the proffered architecture outperforms earlier approaches in several metrics. It reduced the misclassification rates for complex categories as well as enhanced performance in difficult distinctions such as Healthy and Mosaic Virus (MV).The visual explanations using Grad-CAM confirm that the model can focus on the correct disease regions even under challenging conditions. The suggested approach offers an explainable framework to automated diagnosis of tomato diseases. It performs effectively in both controlled as well as real field environments. It successfully corrected misclassifications by reducing the influence of incorrect high-accuracy models and enhancing low-confidence correct outputs.

Future work will focus on extending the model to more crops to increase its usefulness in agriculture. Real-time use on edge devices like drones and IoT sensors can support monitoring of disease and smart farming. Adding factors such as temperature and humidity can improve crop health prediction. Advanced models like transformers may increase accuracy and robustness. Training on diverse datasets from different regions and crop stages will make the model stronger. The incorporation of self-supervised learning and domain adaptation techniques may enhance its capacity to manage variations in light, occlusion, and environmental changes. Pruning and quantizing the model will make it ready to roll on edge devices… These enhancements will contribute to the establishment of an efficient and reliable crop disease detection system.

## Data availability

The dataset utilized in this study is online available at https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset.

## Code availability

https://github.com/satrughankumar/TOMATO-CROP-IMAGES-GENERATED-USING-CGAN

Received: 11 April 2025; Accepted: 13 January 2026
Published online: 02 February 2026

## References

1. Karačić, V. et al. Bacillus species: Excellent biocontrol agents against tomato diseases. *Microorganisms* **12**(3), 457 (2024).
2. Gehlot, M., Saxena, R. K. & Gandhi, G. C. "Tomato-Village": a dataset for end-to-end tomato disease detection in a real-world environment. *Multimedia Syst.* **29**(6), 3305–3328 (2023).
3. Zhou, C., Zhou, S., Xing, J. & Song, J. Tomato leaf disease identification by restructured deep residual dense network. *IEEE Access* **9**, 28822–28831. https://doi.org/10.1109/ACCESS.2021.3058947 (2021).
4. Dang, M., Wang, H., Li, Y., Nguyen, T. H., Tightiz, L., Xuan-Mung, N., & Nguyen, T. N. (2024). Computer Vision for Plant Disease Recognition: A Comprehensive Review. The Botanical Review, 1–61.
5. Hang, Yan, et al. "Ensemble Technique of Deep Learning Model for Identifying Tomato Leaf Diseases Based on Choquet Fuzzy Integral." Cybernetics and Systems (2024): 1–24.
6. Mezenner, A. et al. Fuzzy Aggregation-Based Deep Learning Ensemble for Enhanced Tomato Leaf Disease Classification. *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.* **33**(07), 911–932 (2025).
7. Deng, S. et al. Tomato Leaf Disease Identification Framework FCMNet Based on Multimodal Fusion. *Plants* **14**(15), 2329 (2025).
8. Zhu, H., Wang, D., Wei, Y., Zhang, X. & Li, L. Combining Transfer Learning and Ensemble Algorithms for Improved Citrus Leaf Disease Classification. *Agriculture* **14**(9), 1549 (2024).
9. Lokesh, G. H. et al. Intelligent Plant Leaf Disease Detection Using Generative Adversarial Networks: a Case-study of Cassava Leaves. *The Open Agriculture Journal* https://doi.org/10.2174/0118743315288623240223072349 (2024).
10. Wu, Q., Chen, Y. & Meng, J. DCGAN-based data augmentation for tomato leaf disease identification. *IEEE access* **8**, 98716–98728. https://doi.org/10.1109/ACCESS.2020.2997001 (2020).
11. Ghous, Hamid, et al. "Multi-crops leaves diseases classification using fuzzy logic and pre-trained CNN methods." Journal of Computing & Biomedical Informatics (2024).
12. Ruby, et al. Advanced image processing techniques for automated detection of healthy and infected leaves in agricultural systems. *Mesopotamian J. Comput. Sci.* **2024**, 44–52 (2024).
13. Bachhal, P. et al. Maize leaf disease recognition using PRF-SVM integration: a breakthrough technique. *Sci. Rep.* **14**(1), 10219 (2024).
14. Kaur, B. et al. Precision diagnosis of citrus leaf diseases using image enhancement and nonlinear fuzzy ranking ensemble approach NLFuRBe. *Sci. Rep.* **15**(1), 32296 (2025).
15. Pandiyaraju, V. et al. Improved tomato leaf disease classification through adaptive ensemble models with exponential moving average fusion and enhanced weighted gradient optimization. *Front. Plant Sci.* https://doi.org/10.3389/fpls.2024.1382416 (2024).
16. Tian, X., Meng, X., Wu, Q., Chen, Y. & Pan, J. Identification of tomato leaf diseases based on a deep neuro-fuzzy network. *J. Inst. Eng. India Series A* **103**(2), 695–706. https://doi.org/10.1007/s40030-022-00642-4 (2022).
17. Khan, M., Gulan, F., Arshad, M., Zaman, A. & Riaz, A. Early and late blight disease identification in tomato plants using a neural network-based model to augmenting agricultural productivity. *Sci Prog.* **107**(3), 368504241275371. https://doi.org/10.1177/00368504241275371 (2024).
18. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems, 30..
19. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708), https://doi.org/10.48550/arXiv.1608.06993.
20. Kumar, R., Singh, D., Chug, A., & Singh, A. P. (2022, May). Evaluation of Deep learning based Resnet-50 for Plant Disease Classification with Stability Analysis. In 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1280–1287). IEEE.
21. Tan, M., & Le, Q. (2019, May). EfficientNet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105–6114). PMLR.
22. Saeed, A. et al. Smart detection of tomato leaf diseases using transfer learning-based convolutional neural networks. *Agriculture* **13**(1), 139 (2023).
23. Attallah, O. Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection. *Horticulturae* **9**(2), 149 (2023).
24. Wajid, A.H., Saher, N., Nawaz, S.A., Arshad, M., & Nasir, M. (2024). Tomato leaf disease detection and classification using convolutional neural network and machine learning. Journal of Computing & Biomedical Informatics.
25. Najim, M. H., Abdulateef, S. K. & Alasadi, A. H. Early detection of tomato leaf diseases based on deep learning techniques. *Int J Artif Intell* **13**(1), 509–515 (2024).
26. Santosh, K. C. & Bharathi, N. Image processing technique on identification of leaf types and detection of diseases portion on tomato leaves. *Int. Res. J. Adv. Eng. Manag. (IRJAEM)* **2**(04), 564–570 (2024).
27. Shanthi, D. L., Vinutha, K., Ashwini, N. & Vashistha, S. Tomato leaf disease detection using CNN. *Procedia Computer Science* **235**, 2975–2984 (2024).
28. Javidan, S. M., Banakar, A., Vakilian, K. A. & Ampatzidis, Y. Tomato leaf diseases classification using image processing and weighted ensemble learning. *Agron. J.* **116**(3), 1029–1049. https://doi.org/10.1002/agj2.21293 (2024).
29. Paul, S. G. et al. A real-time application-based convolutional neural network approach for tomato leaf disease classification. *Array* **19**, 100313. https://doi.org/10.1016/j.array.2023.100313 (2023).
30. Bandyopadhyay, S., Sarkar, A., paria, A., & Jana, B. (2022, February). Deep learning approaches for analysis and detection of tomato leaf diseases. In International Symposium on Artificial Intelligence (pp. 265–273). Cham: Springer Nature Switzerland, https://doi.org/10.1007/978-3-031-22485-0_24.
31. Mishra, U. et al. A hybrid approach for plant leaf detection using ResNet50-intuitionistic fuzzy RVFL (ResNet50-IFRVFLC) classifier. *Comput. Electr. Eng.* **123**, 110135 (2025).
32. Moussafir, M. et al. Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks. *Plant Soil* **479**(1), 251–266 (2022).
33. Sharma, P., Berwal, Y. P. S. & Ghai, W. Performance analysis of deep learning cnn models for disease detection in plants using image segmentation. *Information Processing in Agriculture* **7**(4), 566–574 (2020).

34. Şener, A., & Ergen, B. (2024). Advanced CNN Approach for Segmentation of Diseased Areas in Plant Images. Journal of Crop Health, 1–15.
35. Wang, Y., Zhang, P. & Tian, S. Tomato leaf disease detection based on attention mechanism and multi-scale feature fusion. *Front. Plant Sci.* **15**, 1382802 (2024).
36. Kalpana, P. & Anandan, R. A capsule attention network for plant disease classification. *Traitement du Signal* **40**(5), 2051–2062. https://doi.org/10.18280/ts.400523 (2023).
37. Muhammad, A., Salman, Z., Lee, K. & Han, D. Harnessing the power of diffusion models for plant disease image augmentation. *Front. Plant Sci.* **14**, 1280496 (2023).
38. Abbas, A., Jain, S., Gour, M. & Vankudothu, S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* **187**, 106279. https://doi.org/10.1016/j.compag.2021.106279 (2021).
39. Chen, Z., Wang, G., Lv, T. & Zhang, X. Using a hybrid convolutional neural network with a transformer model for tomato leaf disease detection. *Agronomy* **14**(4), 673 (2024).
40. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv 2020, arXiv:2010.11929.
41. Zhang, H., & Ren, G. (2024). Intelligent leaf disease diagnosis: image algorithms using Swin Transformer and federated learning. The Visual Computer, 1–24.
42. Kalpana, P. et al. Plant disease recognition using residual convolutional enlightened Swin transformer networks. *Sci Rep* **14**, 8660. https://doi.org/10.1038/s41598-024-56393-8 (2024).
43. Abdullah, A. et al. A deep-learning-based model for the detection of diseased tomato leaves. *Agronomy* **14**(7), 1593 (2024).
44. Megalingam, R. K., Manoharan, S. K. & Maruthababu, R. B. Integrated fuzzy and deep learning model for identification of coconut maturity without human intervention. *Neural Comput. Appl.* **36**(11), 6133–6145 (2024).
45. Jain, A. et al. Bayesian optimized CNN ensemble for efficient potato blight detection using fuzzy image enhancement. *Sci. Rep.* **15**(1), 31259 (2025).
46. Creswell, A. et al. Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* **35**(1), 53–65. https://doi.org/10.1109/MSP.2017.2765202 (2018).
47. Lilhore, U. K. et al. Enhanced convolutional neural network model for cassava leaf disease identification and classification. *Mathematics* **10**(4), 580 (2022).
48. A method for stochastic optimization He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778. 2016.
49. Ünver, M. (2025). Gaussian Aggregation Operators and Applications to Intuitionistic Fuzzy Classification. Journal of Classification, 1–28.
50. Rainio, O., Teuho, J. & Klén, R. Evaluation metrics and statistical tests for machine learning. *Sci. Rep.* **14**(1), 6086 (2024).


## Acknowledgements

This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R827), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

## Author contributions

Satrughan Kumar, Yogesh Kumar Sharma, and Munish Kumar contributed to the conceptualization and implementation of the deep learning framework. Umesh Kumar Lilhore optimized the model and fine-tuned the algorithm. Sultan Mesfer A Aldossary contributed to model validation and played a key role to review and refine the manuscript. Sarita Simaiya helped analyze the model's output and performance metrics. Md Monish Khan led the coding, architecture design, and experimental execution. Shimaa A. Hussien provided essential revisions to the manuscript, ensuring the clarity and precision of the methodology and results. Ehab Seif Ghith supervised the research, guiding model validation and contributing to the manuscript revisions.

## Funding

This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R827), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

## Declarations

## Competing interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Ethical approval

All authors have read, understood, and have complied as applicable with the statement on "Ethical responsibilities of Authors" as found in the Instructions for Authors.

## Additional information

**Correspondence** and requests for materials should be addressed to U.K.L. or M.M.K.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.