Diyala Journal of Engineering Sciences

Journal homepage: https://djes.info/index.php/djes

ISSN: 1999-8716 (Print); 2616-6909 (Online)

Progressive Resolution Training with Attention-Enhanced Ensemble Learning for Automated Recognition of Sugarcane Leaf Diseases

Gautam Kumar*, Vivek Bhatnagar

MM Institute of Computer Technology and Business Management, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala – 133 207, Haryana, India

ARTICLE INFO

Article history:
Received: 24/01/2026.
Revised: 29/04/2026,
Accepted: 23/05/2026,
Available online: 15/06/2026.

Keywords:
Deep Learning
Sugarcane Disease
Multistage Training
Attention Mechanisms
Test-Time Augmentation

ABSTRACT

Sugarcane (Saccharum officinarum) is one of the most economically important crops in the world. However, outbreaks of leaf diseases cause extensive annual yield losses and threaten sugarcane agricultural productivity. Therefore, early and accurate disease detection is vital for effective crop management, yield protection, and food security. This study presents a deep ensemble learning framework, termed CBAM-SugarcaneNet, for the automated recognition of sugarcane leaf diseases under field conditions. The proposed framework integrates progressive multi-resolution training, a Convolutional Block Attention Module (CBAM), focal loss, Sharpness-Aware Minimization (SAM), ensemble learning, and test-time augmentation to address major challenges such as class imbalance, inter-class visual similarity, background variation, and limited training data. A three-stage progressive training strategy was employed using ConvNeXt-Large and EfficientNet-B5/B7 architectures enhanced with CBAM attention modules to improve the accuracy of disease-relevant feature extraction. Focal loss was used to emphasize minority and difficult-to-classify disease classes, whereas SAM improved model generalization by encouraging convergence toward flatter minima. Experiments were conducted on an 11-class sugarcane leaf disease dataset comprising 6,748 images. The proposed framework achieved 97.2% accuracy for single-model deployment, 97.6% for the ensemble configuration, and 98.1% with test-time augmentation, with a mean 5-fold cross-validation accuracy of 98.05% ± 0.14%. Grad-CAM visualizations further confirmed that the model focused on disease-affected leaf regions. The developed framework offers a robust and deployable solution for precision-agriculture-based disease diagnosis in sugarcane.

1. INTRODUCTION

Sugarcane (Saccharum officinarum) ranks among the top 10 agricultural commodities produced in the world, at the rate of over 1.9 billion tons per year [1]. Sugarcane is a primary source of sugar production and biofuel feedstock, and it plays a vital role in enhancing food security and supporting renewable energy initiatives over the globe [2]. Nevertheless, disease outbreaks pose significant damage to sugarcane cultivation, resulting in yield loss ranging from 10% to over 60%, depending on prevailing environmental conditions and the severity of the diseases [3]. Traditional methods for diagnosing leaf diseases mainly depend on visual assessments by experts. Conventional methods to detect agricultural diseases can be time-consuming, subjective, labor-intensive, and may not have accessibility all over the world due to lack of resources [4]. The arrival of deep learning and computer vision technologies has transformed the field of agricultural disease detection by providing automated, scalable, and objective diagnostic solutions [5, 6]. Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image classification tasks and can achieve equal or better performance than humans in several fields, including medical imaging [7, 8] manufacturing quality control [9], and agricultural applications [10].

Recent advancements in architectural design, optimization methodologies, and training strategies have significantly enhanced the model capabilities for fine-grained visual recognition tasks [11-13]. However, despite these advances, the automated

* Corresponding author's E-mail: gautam.bopara@gmail.com
DOI: 10.24237/djes.2026.19202

classification of sugarcane diseases faces several challenges and requires the development of specific methodologies. Many sugarcane diseases have indistinguishable visual symptoms, particularly during the early stages, which can complicate differentiation even for expert agronomists [14]. The frequency of disease occurrence widely depends on the conditions in the field, resulting in significantly imbalanced datasets with rarer diseases that are underrepresented [15]. Disease manifestations can vary extensively depending on the different stages of infection, crop susceptibility, environmental conditions, and the existence of co-infections [16]. Collecting and annotating large disease datasets requires extensive domain expertise, time, and resources, particularly for rare diseases. In practical applications, it is crucial to address variations in image quality, leaf orientation, lighting conditions, background clutter, and the variety of image-capturing devices [17-18].

Most of the contemporary methodologies primarily utilize standard convolutional neural network (CNN) architectures with standard training protocols, resulting in achieving accuracies between 89-96% [19]. However, these methodologies often neglect essential elements, such as systematic class-imbalance management, the employment of progressive multi-resolution training, the incorporation of attention mechanisms to learn discriminative features, and development of a robust generalization framework. Additionally, the limited comparative analysis across numerous cutting-edge architectures and the lack of inclusive ablation studies impede a thorough understanding of the optimal design choices.

This study examines the extent to which a holistic deep ensemble framework (based on progressive resolution training, attention mechanisms, and advanced optimization) can improve the accuracy and robustness of automated sugarcane leaf disease classification compared with existing state-of-the-art methods, particularly when addressing challenges like class imbalance and inter-class similarity.

To enhance feature learning, stable convergence, and generalization, focal loss, selective sharpness-aware minimization and cosine annealing were used in conjunction with a holistic three-stage progressive resolution training algorithm. The combination of CBAM-enhanced ConvNeXt-Large and EfficientNet-B5/B7 backbone networks in a performance-weighted ensemble with extensive test-time augmentation, allowed for improved localization of disease-relevant features as well as improved robustness. An accuracy of 98.1% with a 1.7% absolute improvement over prior work through extensive cross-validation, ablation analysis, and statistical significance testing was achieved, supported by a comparative evaluation of the single-model and ensemble model deployment trade-offs.

The remainder of this paper is structured as follows: Section 2 reviews related work in the domains of plant disease classification and deep learning methodologies. Section 3 describes the proposed methodology. Section 4 outlines the experimental setup and implementation of the proposed method. Section 5 presents a comprehensive analysis of the results, including ablation studies and comparative evaluations. Section 6 discusses the findings, limitations, and potential directions for future research. Finally, Section 7 concludes the study.

2. RELATED WORK

This section reviews the transition from conventional machine learning methods to contemporary deep learning approaches for plant leaf disease detection. It further sheds light on important research gaps in sugarcane leaf disease classification that motivate the development of a comprehensive and systematically optimized framework.

2.1 Traditional and Early Deep Learning Approaches for Plant leaf Disease Detection

Initial systems for detecting plant leaf diseases predominantly utilized traditional machine learning techniques along with manually crafted feature extraction techniques. Phadikar and Sil [20] used color and texture features with Support Vector Machines (SVM) to classify rice crop diseases and achieved an accuracy of 75%. Camargo and Smith [21] developed an image-processing pipeline incorporating color and texture analyses for cotton leaf disease detection. Although these approaches have demonstrated the feasibility of automated disease detection, they have limitations in terms of generalization, dependence on manual feature engineering, and reduced performance under diverse field conditions.

The advent of deep learning has significantly improved plant disease recognition. Mohanty et al. [22] applied AlexNet and GoogLeNet to the PlantVillage dataset and achieved an accuracy of 99.35% on laboratory images. However, their performance declined on field images, highlighting the challenge of domain differences between controlled and real-world environments. Ferentions [23] conducted comparisons of CNN architectures including AlexNet, VGG, and ResNet across multiple plant-disease combinations, reporting an accuracy of 99.53%. Similarly, Sladojevic et al. [24] used CaffeNet to classify 13 plant disease categories with an accuracy of 96.3%. Jadhav et al. [25] utilized GoogLeNet and AlexNet for soybean disease classification, achieving accuracies of 96.25% and 98.75%, respectively. Too et al. [26] introduced DenseNet architectures for plant disease recognition, demonstrating improved parameter efficiency and performance. Despite these advances, many studies have focused on controlled datasets and have not fully

addressed the challenges associated with real-world agricultural conditions.

2.2 Advanced CNN Architectures and Attention Mechanisms

Recent work on advanced convolutional neural networks (CNNs) has been done in relation to CNN architectures that are intended for agriculture-related tasks [27, 28]. Kainat et al. [29] used CNN models for identifying diseases on cucumber plants through data augmentation to improve robustness of the models. Dash et al. [30] used DenseNet-201 for the feature extraction of maize disease and SVM classification for the disease detection and achieved an accuracy of 94.6%. Tm et al. [31] employed AlexNet with transfer learning to identify plant diseases among multiple varieties of plants.

EfficientNet architectures are becoming increasingly popular as a result of their favorable trade-off between accuracy and computational efficiency. Tariq et al. used EfficientNet-B4 for identify tomato plant diseases [32]. Ullah et al. [33] employed EfficientNet-B7 for detection of wheat leaf disease. Additionally, ConvNeXt is another new convolutional architecture that was inspired by transformer design principles and has been shown to be effective in the area of agricultural plant leaf disease detection research [12]. However, there is limited application to sugarcane leaf disease detection using ConvNeXt.

Attention mechanisms direct their attention towards the regions of a plant leaf that is affected by disease and have also improved the performance of deep learning models. Woo et al. [11] created the Convolutional Block Attention Module (CBAM), to add both channel and spatial attention for improve feature representation by using CNNs. Tunio et al. [34] applied attention-based model approach for rice disease classification, and achieved an accuracy of 95.24%. V. et al. [35] used squeeze-and-excitation networks to identify chili leaf diseases. These studies show that Attention mechanisms have been incorporated into state-of-the-art deep learning models; however, not much research has been done on the systematic application of attention mechanisms in modern convolutional neural networks for identifying agricultural crop leaf disease.

2.3 Sugarcane Leaf Disease Classification

Research to develop an automated system to classify diseases of sugarcane leaf has been somewhat limited compared to other crops types. Narmilan et al. [36] used conventional ML techniques for sugarcane leaf disease detection and attained an accuracy of 94%. Srivastava et al. [37] and Malik et al. [38] utilized VGG-16 for sugarcane leaf disease classification with an accuracy of 84.4% and 92.3% respectively on limited datasets. Öğrekçi et al. [39] compared performance of DenseNet121, ViT, and hybrid of ViT+CNN on a 5-class sugarcane leaf disease dataset containing 2,521 images, and achieved best precision of 93.34% with pure ViT, though the hybrid combination underperformed at 87.37%.

More recent research on improved models and learning strategies has brought about advances in the diagnosis of the sugarcane leaf disease. The study by Kavitha and Krishna Prasad [40] presented an ensemble of shallow CNNs with an accuracy of 97.61% and demonstrated a 5.21% improvement in accuracy over KNN-based models; However, they failed to assess any current deep learning architectures. In contrast, the method of Daphal and Koli [41] employed an attention-based multilevel CNN within an Android mobile app that provides real-time diagnosis results yielding 86.53% accuracy while optimizing for limited resource and performance. Huang et al. [42] demonstrated improved recognition performance of various augmentation techniques for classification of diseased sugarcane leaves. Another study by Kunduracıoğlu and Paçal [43] provided a comparative evaluation of EfficientNet model families (B0–B7 and V2) on the classification of 11 sugarcane leaf disease classes illustrating that B0 produced the greatest accuracy (93.39%) from a dataset of 6,748 image samples. All of these studies highlighted the ability of deep learning models to detect sugarcane leaf diseases, as well as the continuing opportunities for additional exploration of architecture and training methodologies.

2.4 Ensemble Learning and Advanced Training Strategies

Ensemble learning integrates predictions from multiple models to improve classification accuracy and robustness. Shahid et al. [44] employed ensemble models for cotton disease classification and achieved an accuracy of 98.4%. Ali et al. [45] explored weighted ensemble methods for plant disease recognition across multiple crops and conducted comparative analyses of ensemble strategies and reported improved performance using heterogeneous architecture combinations. Chen et al. [46] applied stacking ensembles for plant disease classification, while Astani et al. [47] examined the robustness of ensemble methods under varying environmental conditions.

Progressive training strategies have also demonstrated potential for improving model performance in computer vision tasks. Lu et al. [48] discussed progressive growth strategies for generative models, while Touvron et al. [49] applied progressive resizing for image classification tasks. Liu and Huang [50] presented progressive multi-scale training for object detection. In agricultural contexts, Sun et al. [51] investigated multi-scale feature fusion for maize disease detection, whereas Brahimi et al. [52] studied the impact of image resolution on plant disease classification accuracy.

Advanced optimization techniques further improve model generalization. Foret et al. [53] introduced Sharpness-Aware Minimization (SAM), which encourages flatter minima during training. Izmailov et al. [54] presented Stochastic Weight Averaging (SWA), while Loshchilov and Hutter [55] developed the AdamW optimizer with decoupled weight decay. Test-Time Augmentation (TTA) improves prediction robustness by averaging predictions across multiple augmented versions of test images [56-58].

Despite significant advancements in automated plant disease detection, there are substantial limitations in the classification of sugarcane diseases. Previous studies have often failed to systematically compare contemporary deep learning architectures, such as ConvNeXt and EfficientNet, and have seldom integrated progressive resolution training, attention mechanisms, and advanced optimization techniques into a cohesive system. Additionally, there is a lack of comprehensive ablation studies that assess the impact of individual components, detailed analyses of the trade-offs between single-model and ensemble approaches for practical applications, rigorous statistical validation, and optimized test-time augmentation methods specifically tailored for agricultural contexts. To address these challenges, this study developed a systematically designed framework that integrates progressive resolution training, attention mechanisms, ensemble learning, and advanced optimization strategies, supported by extensive empirical evaluation, thereby advancing automated sugarcane disease classification.

3. METHODOLOGY

This section outlines our comprehensive deep ensemble learning framework, which was designed for the classification of sugarcane leaf diseases, as shown in Figure 1. The developed ensemble framework, termed CBAM-SugarcaneNet, is an AI-based, progressive, deep ensemble model. It incorporates Convolutional Block Attention Modules, multi-resolution training, and performance-weighted aggregation to facilitate the automated detection of sugarcane leaf diseases. All functional steps are presented in Algorithm 1.

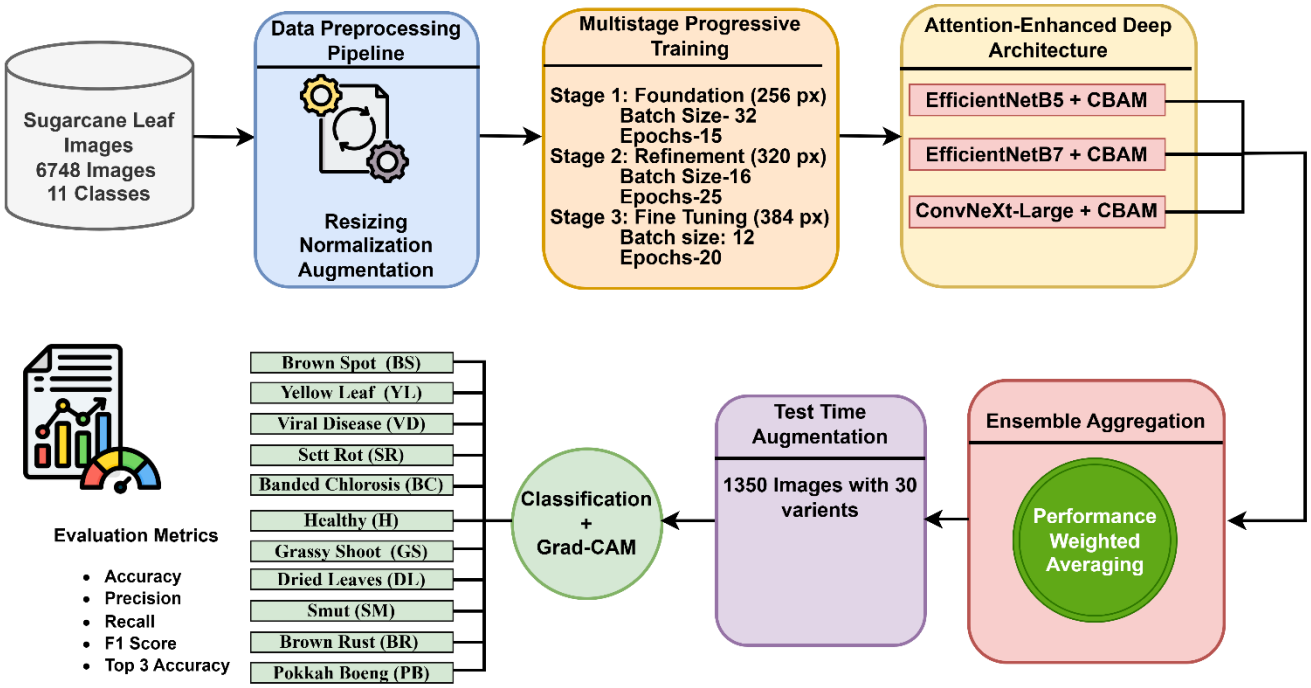

Figure 1: Flow diagram of the CBAM-SugarcaneNet model

Algorithm 1: CBAM-SugarcaneNet: Progressive Resolution–Aware Ensemble Training with Cross-Validation	
Step 1:	**Input:** Dataset *D*, Models {M1, M2, M3}, Stages S ={(256, 15), (320, 25), (384, 20)}
Step 2:	Initialize models with ImageNet weights
Step 3:	**for** each fold f in 5-fold CV **do**
Step 4:	Split D into $D_{train}^{(f)}$, $D_{val}^{(f)}$
Step 5:	**for** each model M_k **do**
Step 6:	$w_k^{(0)} \leftarrow$ ImageNet weights
Step 7:	**for** each stage (r_s, e_s) in S **do**

Step 8: Resize images to $r_s \times r_s$
Step 9: **if** stage 1 AND epoch ≤ 5 then
Step 10: Freeze backbone, train head only
Step 11: **else**
Step 12: Unfreeze all layers
Step 13: **end if**
Step 14: Set learning rate schedule for stage s
Step 15: **if** M_k is ConvNeXt AND r_s = 320 **then**
Step 16: Use SAM optimizer with ρ = 0.02
Step 17: **else**
Step 18: Use AdamW optimizer
Step 19: **end if**
Step 20: **for** epoch e in e_s **do**
Step 21: Train on $D_{train}^{(f)}$ with focal loss
Step 22: Validate on $D_{val}^{(f)}$
Step 23: **if** validation accuracy improved **then**
Step 24: Save checkpoint w_k^{best}
Step 25: **end if**
Step 26: **end for**
Step 27: **end for**
Step 28: Load best checkpoint w_k^{best}
Step 29: **end for**
Step 30: Compute ensemble weights from validation accuracies
Step 31: Evaluate ensemble with TTA on test set
Step 32: **end for**
Step 33: **Return:** Mean and Standard deviation (std) accuracy across folds

The overall architecture encompasses data preprocessing, progressive multi-resolution training, attention-enhanced model architectures, ensemble aggregation, test-time augmentation, and Grad-CAM visualization, which are discussed in the following sections.

3.1 Problem Formulation

Given a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^{H\times W\times 3}$ represents RGB leaf images and $y_i \in \{1,2,\dots,C\}$ denotes disease class labels (C=11 for our dataset), our objective is to learn a robust mapping function f: $\mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{C}$ that accurately classifies unseen leaf images while handling class imbalance, inter-class similarity, and field condition variations.

3.2 Data Preprocessing

Data preprocessing is a vital step to improve the generalizability of models, since the conditions under which images are captured in an agricultural environment can vary widely. Image processing provides computational means for improving and analyzing visual data so as to allow for automated extraction of information. For example, in an agricultural setting and for the specific application of detecting diseases in sugarcane, image processing allows for accurate identification of plant pathological symptoms under varying light conditions and complex backgrounds. Thus, image processing forms a critical part of the technology used to monitor crops using computer vision and deep-learning models. The data preprocessing pipeline consists of three stages:

Image Normalization: Normalizing of image intensities (or pixel intensities) by converting to a common range (e.g., [0, 1] or [-1, 1]) reduces the effects of varying lighting conditions and the influence of very large features on training. Normalizing pixels also increases numerical stability of the model, speeds up training time, and improves model generalization. These benefits have been shown to be particularly true for deep neural networks and previously trained models, both of which assume a consistent distribution of input data. Therefore, normalizing image pixels in agricultural images (in this example, scaling each RGB pixel's value to a range of [0,1]) will enhance the ability of deep neural networks to learn features representative of the entire image.

Standardization: Standardizing the distribution of pixel intensities for each R, G, and B channel of an image is necessary to promote training efficiency and optimize the ability for features learned by a transfer-learning trained model on one data set to generalize to another. Accordingly, in this study, the standardization process was performed using ImageNet image statistics (μ = 0.485, 0.456, 0.406, for the mean for each R, G, and B pixel respectively; σ = 0.229, 0.224, 0.225 respectively) to determine the pixel intensity μ (mean) and σ (standard deviation) across all images. The

standardization of pixel intensity by color channel was achieved prior to the preservation of aspect ratio by using a center crop of the image so that the sugarcane leaf feature would be properly scaled to the crop's structural components and spatial characteristics. Standardization of images via R, G, and B pixel/channel was conducted according to ImageNet mean and standard deviation statistics as follows:

$$x' = \frac{x-\mu}{\sigma} \quad (1)$$

Where, x is original pixel intensity, μ is mean of the channel, σ is standard deviation of the channel and x' is standardized pixel value. After standardization, each channel has mean ≈ 0 and variance ≈ 1.

Augmentation: In order to improve the diversity of training data, to decrease the extent of overfitting and to increase how well our model generalizes when exposed to more realistic examples of what it has been trained on, we used data augmentation. Data augmentation provides a mechanism to improve the robustness and generalization of a model through providing reliable feature learning when exposed to variations in light, orientation, and background conditions. A moderate level of augmentation intensity was used to achieve a balance between a diverse training set and keeping biological plausibility. The types of augmentations that were performed appear in Table 1.

The augmentation parameters were carefully tuned to avoid unrealistic transformations that could degrade learning. All augmentations were implemented using the Albumentations library [59] for improved efficiency.

Table 1. Data augmentation strategies and corresponding parameters employed during training

Augmentation Technique	Parameters	Probability (p)
Horizontal Flipping	Left–right image flip	0.5
Rotation	±25° rotation with reflection padding	0.5
Brightness/ Contrast Adjustment	±20% intensity variation	0.5
Hue–Saturation–Value (HSV) Shift	Random HSV perturbation	0.4
Gaussian Blur	Kernel size = 3	0.15
Gaussian Noise	Noise variance in the range 5–15	0.15
Coarse Dropout	2 holes, 7% area per hole	0.1

3.3. Progressive Multi-Resolution Training

The progressive training approach systematically enhances the input resolution across successive training stages, allowing the models to initially learn coarse features before refining fine-grained discriminative patterns. The framework employs three progressive stages, as shown in Table 2, to illustrate the three-stage progressive training process.

Table 2. Progressive multistage training strategy

Stage	Input Resolution	Epochs	Training Strategy	Learning Rate Schedule	Batch Size	Optimizer / Special Setting	Objective
Stage 1	256 × 256	15 (5 + 10)	Warmup with frozen backbone (5 epochs), followed by fine-tuning with unfrozen backbone (10 epochs)	Constant α = 1×10^{-4} (warmup); cosine annealing from 5×10^{-5} to 5×10^{-6}	32	Standard optimizer	Establish robust feature representations and mitigate overfitting
Stage 2	320 × 320	25	Extended fine-tuning at intermediate resolution	Cosine annealing from 4×10^{-5} to 4×10^{-6}	16	SAM applied to ConvNeXt-Large (ρ = 0.02)	Refine discriminative features using sharpness-aware optimization
Stage 3	384 × 384	20	Final fine-tuning at target resolution	Cosine annealing from 8×10^{-6} to 8×10^{-7}	12	Standard optimizer	Fine-tune subtle discriminative patterns for challenging disease pairs

Sharpness-Aware Minimization (SAM) improves generalization by encouraging convergence to flat loss regions, making the model robust to small parameter perturbations. Learning rate schedules are model-specific, with ConvNeXt requiring lower rates owing to its deeper architecture and SAM sensitivity. For each stage, the learning rate follows cosine annealing as follows:

$$\alpha_t = \alpha_{min} + \frac{1}{2}(\alpha_{max} - \alpha_{min})\left(1 + \cos\left(\frac{t\pi}{T}\right)\right) \quad (2)$$

where t is current epoch, T is total epochs for stage, α_{max} and α_{min} are maximum and minimum learning rates. This schedule provides smooth convergence while multistage learning rate drops that can destabilize the training.

3.4 Ensemble Architecture with CBAM-Enhanced Feature Learning

We used an ensemble of three complementary architectures, each augmented with CBAM attention mechanisms, which are discussed below:

EfficientNet-B5: This compound-scaled architecture achieves an optimal balance between depth, width, and resolution. It incorporates inverted residuals, squeeze-and-excitation modules, and swish activations, comprising a total of 30.0 million parameters.

EfficientNet-B7: As a larger compound-scaled variant, this architecture enhances the capacity to recognize complex patterns, with a total of 66.0 million parameters.

ConvNeXt-Large: This modern ConvNet features a hierarchical architecture that includes depth-wise convolutions, layer normalization, and GELU activation. The architecture was structured into four stages with depths [3, 3, 27, and 3] and dimensions [192, 384, 768, and 1536], totalling 197.8 million parameters.

3.4.1 CBAM Attention Integration

The Convolutional Block Attention Module (CBAM) is an efficient attention mechanism that enhances feature representations by sequentially modeling channel-wise and spatial dependencies within convolutional feature maps. Channel attention captures the relative importance of different feature channels through inter-channel dependency modeling, whereas spatial attention focuses on identifying informative spatial regions by emphasizing the discriminative locations within the feature maps.

Incorporating the CBAM allows deep neural networks to prioritize task-relevant features while attenuating background and redundant information, thereby improving the representational quality and robustness. Due to its minimal computational overhead and modular integration, CBAM is well suited for fine-grained visual recognition tasks, including agricultural disease classification, where precise localization of disease-affected regions contributes to improved classification accuracy.

We integrate Convolutional Block Attention Modules after backbone feature extraction; Equations 3 and 4 represent the channel and spatial attention, respectively.

Channel Attention: Emphasizes disease-relevant feature channels as follows:

$$M_c(F) = \sigma(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))) \quad (3)$$

where F is the input feature map, $AvgPool(\cdot)$ and $MaxPool(\cdot)$ denote spatial average and max pooling operations, MLPrepresents the shared multi-layer perceptron, and $\sigma$is the sigmoid activation function.

Spatial Attention: Highlights disease symptom regions as

$$M_s = \sigma(f^{7\times7}([AvgPool(F'); MaxPool(F')])) \quad (4)$$

where $F' = M_c \otimes F$ are channel-refined features, $f^{7\times7}$denotes a 7×7convolution, and $[\,;\,]$ represents channel-wise concatenation. The MLP contains two fully connected layers with a reduction ratio of $r = 16$ for ConvNeXt and $r = 32$ for EfficientNets. Final features are represented as $F'' = M_s \otimes F'$.

3.4.2 Classification Head

A classification head serves as the task-specific component of a deep neural network, which is responsible for converting the high-level feature representations produced by the backbone into class-wise prediction scores. While the backbone captures generic and hierarchical visual features, the classification head enables their effective transformation into a discriminative representation that is aligned with the target label space.

The use of a dedicated classification head facilitates model adaptation to domain-specific datasets, enhances inter-class separability, and supports efficient transfer learning when pretrained backbones are employed. A major contribution to achieving accurate and consistent classification results is the Classification Head, which serves as an intermediary between the Feature Extraction and Decision-Making components of the system. Using a non-linear and smooth activation function such as GELU in combination with the ConvNext architecture allows for better gradient flow as well as providing greater feature expressiveness. ReLU is computationally efficient and complements the convolutional design of EfficientNet by promoting sparse activations and robust gradient propagation. Each architecture employed a custom classification head, as shown in Figure 2.

3.5 Loss Function and Optimization

To effectively address class imbalance and achieve stable optimization, the training process integrates tailored loss formulations and advanced optimization techniques designed to focus on learning challenging samples while promoting well-generalized solutions. The loss function and optimization techniques used are discussed below.

To address the issue of significant class imbalance, with class frequencies ranging from 246 to 1,722 samples, we employed focal loss [60]. Focal loss is an adapted classification loss function specifically designed to address class imbalance by diminishing the impact of well-classified (easy) samples and

concentrating the training efforts on challenging misclassified examples. It enhances the standard cross-entropy loss by incorporating a modulating factor that reduces the loss attributed to confident prediction. Mathematically, the focal loss is defined as:

$$\mathcal{L}_{\text{focal}} = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \quad (5)$$

where p_t is predicted probability for ground truth class, α = 0.25 balances positive/negative examples, γ = 2 down-weights well-classified examples. Focal loss automatically emphasizes hard-to-classify examples and rare disease classes.

For Stage 2 ConvNeXt training, we employed SAM to find flat minima:

$$\min_{w} \max_{\|\epsilon\|_2 \le \rho} \mathcal{L}(w+\epsilon) \quad (6)$$

The SAM performs two forward-backward passes as follows:

Compute gradient g = ∇w $\mathcal{L}$ (w)

Compute perturbation $\epsilon = \rho \frac{g}{\|g\|_2}$

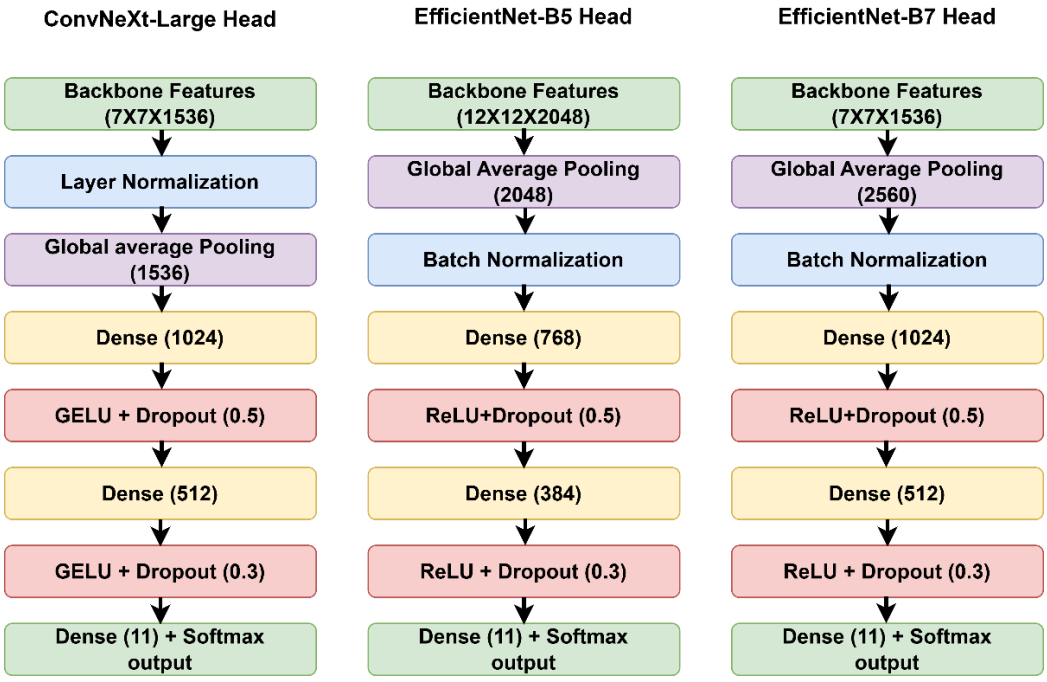

Figure 2. Classification head architectures of the developed CBAM-SugarcaneNet model

Update weights using ∇w $\mathcal{L}$ (w + ϵ)

We set ρ = 0.02 for stability. SAM is applied selectively (ConvNeXt at 320px) due to 2×computational cost.

In this study, the AdamW optimizer [54] with decoupled weight decay λ = 0.01 was employed to ensure stable convergence and effective regularization through decoupled weight decay. Unlike conventional Adam, AdamW separates weight decay from gradient-based parameter updates, thereby providing improved generalization and more reliable optimization behavior, particularly in deep neural networks.

The first- and second-order moment estimates of the gradients are computed as

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (7)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (8)$$

where g_t denotes the gradient at iteration t, while β_1 and β_2 control the exponential decay rates of the moment estimates. The parameter update rule is as follows:

$$w_t = w_{t-1} - \alpha\left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} + \lambda w_{t-1}\right) \quad (9)$$

Here, α represents the learning rate, $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected moment estimates, ϵ is a small constant for numerical stability, and λ denotes the weight decay coefficient. In our experiments, the hyperparameters are set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\lambda = 0.01$. AdamW separates the update of the adaptive model weights and modifies their weight loss through an individual update of the weights, meaning that the generalization capabilities of the model parameter against the learning dynamics for model convergence become greater and as a result produce improved convergence stability and generalization performance.

3.6 Ensemble Aggregation and Test Time Augmentation

Using ensemble aggregation is a strong approach, which combines outputs from a group of independent models. This will give us an overall result with better accuracy and reliability than a single classifier could

achieve. When diversifying models, we can use different types of architectures to create complementary representations of features and provide different types of errors. When combining these different prediction results into one combined prediction, we can reduce the variance of the overall predictions through carefully combining each of the predictions of these models. When combining predictions, the aggregation step utilizes methods such as voting or weighted averages to combine strengths from the individual models and discount the individual weaknesses. Ensemble aggregation takes advantage of the combined intelligence of multiple model architectures to improve the generalization performance over the standard and be more robust against overfitting. Thus, using ensemble aggregation for critical applications, where high confidence prediction is essential. Once all the models had been trained then we created an ensemble using performance weighted average for prediction results as follows:

$$p_{\text{ensemble}}(y \mid x) = \sum_{k=1}^{K} w_k \cdot p_k(y \mid x) \quad (10)$$

where pk(y|x) is the probability from model k, and the weights are computed as:

$$w_k = \frac{\text{acc}_k}{\sum_{j=1}^{K} \text{acc}_j} \quad (11)$$

Based on the validation accuracy, this adaptive weighting emphasizes the better-performing models.
Test-Time Augmentation (TTA) is a methodological approach in which multiple transformed iterations of a test image are generated during the inference phase. The predictions derived from these iterations were subsequently aggregated to yield a final decision. This technique is employed to enhance the robustness and reliability of the predictions by mitigating sensitivity to variations in orientation, scale, and illumination. Consequently, it improves generalization capabilities without requiring modifications to the trained model. TTA transformations are presented in Table 3.

Table 3. Test-time augmentation (TTA) transformations and the number of variants used per image

Category	Transformation Type	Number of Variants
Baseline	Original image	1
Geometric	Horizontal / Vertical / Both flips	3
Geometric	Rotations (90°, 180°, 270°)	3
Geometric	Flip + rotation combinations	6
Photometric	Brightness adjustment (0.9, 0.95, 1.05, 1.1)	4
Geometric	Scale adjustment (0.9, 0.95, 1.05, 1.1)	4
Combined	Multiple geometric and photometric transforms	9
Total	—	30 augmentations

The final prediction using test-time augmentation is computed as a weighted probability aggregation, where the original (unaugmented) image is given higher importance. This is formally expressed as:

$$p_{\text{TTA}}(y \mid x) = \frac{2 \cdot p(y|x_0) + \sum_{i=1}^{29} p(y|x_i)}{31} \quad (12)$$

where $p(y \mid x_0)$denotes the predicted class probability for the original input image, and $p(y \mid x_i)$ represents the predictions obtained from the 29 augmented variants. Assigning a twofold weight to the original image ensures that the primary visual representation contributes more strongly to the final decision while still benefiting from the robustness provided by test-time augmentation.

3.7 Grad-CAM–Based Visual Interpretability for the Weighted Ensemble

To improve the transparency and interpretability of the CBAM-SugarcaneNet, we incorporated Gradient-weighted Class Activation Mapping (Grad-CAM) as a post-hoc explainability mechanism for weighted ensemble predictions. Grad-CAM offers class-discriminative visual explanations by highlighting the spatial regions within an input image that most significantly influence the model's decision, thus enabling qualitative validation of disease-specific feature learning.

3.7.1 Grad-CAM for Individual Ensemble Members

For each ensemble model M_k, Grad-CAM is computed using the gradients of the predicted class score y^c with respect to the feature maps A_k^l of the final convolutional layer. The importance weights for each feature map channel are obtained via global average pooling of the gradients:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{k,ij}^l} \quad (13)$$

where Z denotes the number of spatial locations in the feature map. The class-specific localization map for model M_k is then computed as:

$$L_k^c = \text{ReLU}\left(\sum_l \alpha_k^c A_k^l\right) \quad (14)$$

The ReLU activation ensured that only features with a positive influence on the target class contributed to the visualization. The resulting heatmaps provide insights into the disease-relevant regions identified by each architecture.

3.7.2 Performance-Weighted Ensemble Grad-CAM

We introduce a method for creating a cohesive interpretability map for ensemble predictions by employing a Grad-CAM aggregation strategy that is weighted by performance and aligns with the probabilistic ensemble framework outlined in Equation 10. In particular, the resulting ensemble Grad-CAM heatmap is derived as follows:

$$L^{c}_{\text{ensemble}} = \sum_{k=1}^{K} w_k \cdot L^{c}_{k} \quad (15)$$

where w_k denotes the normalized validation-accuracy-based weight assigned to model M_k, as defined in Equation 11. This formulation ensures that models with superior validation performance exert a stronger influence on the final explanation, whereas weaker models contribute less proportionally.

3.7.3 Interpretation Benefits in Agricultural Disease Detection

By adding Grad-CAM into a multi-model weighted ensemble framework not only provides better interpretability but also increases the confidence in predictions made by the models used in an ensemble. Visual explanations from Grad-CAM confirm that the models focus on important features of a disease to help researchers analyze why a disease was misclassified through qualitative reasons related to using CBAM to increase the amount of attention directed towards affected areas. Heatmaps generated at the ensemble level show that models based on different architectures focus on similar regions when making predictions, which increases confidence in the final predictions. Therefore, the use of attention mechanisms and the ability to explain how predictions were made is critical for producing reliable and accurate predictions that will have practical applications in agriculture.

4. EXPERIMENTAL SETUP

This section presents a detailed description of the sugarcane leaf disease dataset, including the class-wise distribution of samples, strategy for splitting the dataset, and evaluation methodology. Furthermore, this section provides details regarding the computational setup, training configuration, evaluation metrics, and baseline models used to allow for a rigorous, unbiased comparison with current state-of-the-art methods based on published works.

4.1 Dataset Description

In this study, we utilized a large sugarcane leaf disease dataset with 6748 field-collected images that have been classified into 11 classes (9 diseases along with healthy and dried leaves) which is available on Mendeley [61]. The images were collected from the field under natural conditions with varying amounts of light, background, and leaf orientation. The images were captured in RGB format JPEG (. jpg) files at typical 768x1024 pixel resolutions with a spatial (dpi) of 72. The distribution of the image samples for each class in the dataset is presented in Table 4. The class distribution is visualized in Figure 3, and the image samples for each class are shown in Figures 4.

Table 4. Class wise dataset description

Class	Number of Images per class
Brown Spot (BS)	1722
Yellow Leaf (YL)	1194
Viral Disease (VD)	663
Sett Rot (SR)	652
Banded Chlorosis (BC)	471
Healthy (H)	430
Grassy Shoot (GS)	346
Dried Leaves (DL)	343
Smut (SM)	316
Brown Rust (BR)	314
Pokkah Boeng (PB)	297
Total	6748

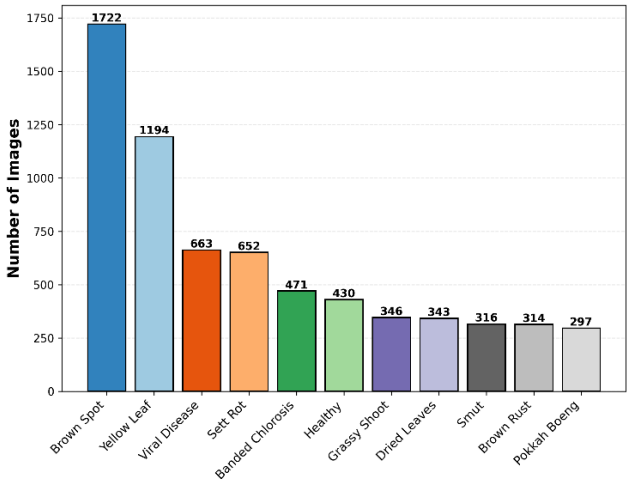

Figure 3. Distribution of images per class

4.2 Data Splitting

An 80-20 stratified train-test split was employed to retain the class distributions in both the training and testing datasets, with a 5-fold stratified cross-validation performed on the training dataset only. The results yielded a training dataset that contained 5,398 images (80%) and a test dataset of 1,350 images (20%) for the final evaluation of the models. The 5-fold cross-validation was completed with approximately 1,080 images in each fold. These two steps, involving the use of an evaluation method, help produce robust performance estimates while reducing the likelihood of overfitting. The held-out test dataset was not used for model training, hyperparameter tuning, or model selection; therefore, it was strictly set aside for the final evaluation of the model performance and was completely independent of any of the training data. All hyperparameter optimization and model selection studies were performed on the training data using a stratified 5-fold cross-validation scheme.

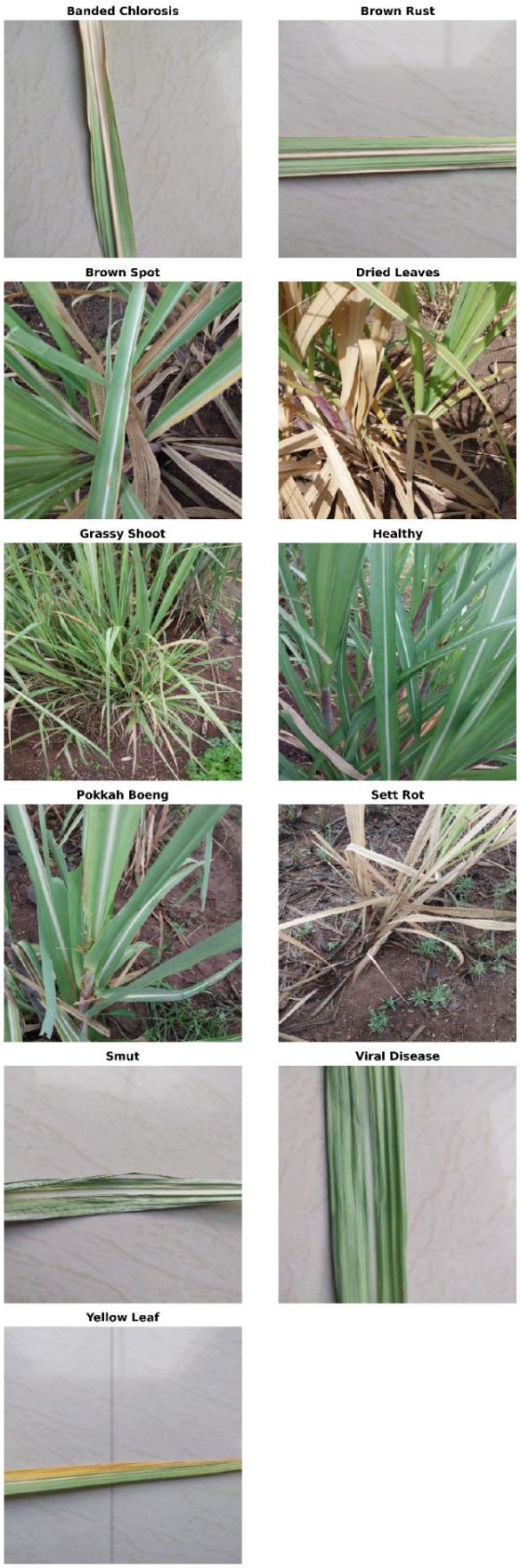

Figure 4. Sample images from dataset

4.3 Implementation Details

The proposed methodology was implemented within a high-performance computing system with an NVIDIA A100 Graphics Processing Unit equipped with 40 GB of dedicated memory and operating under CUDA version 11.8 while using TensorFlow 2.15 in conjunction with the Keras application programming interface to develop and train the model based on the developed methodology. A mixed-precision (16-bit + 32-bit) arithmetic solution was used for all of the large-scale ensemble architectures to overcome the memory limitations associated with using large-scale ensembles. This substantially improved both the computational performance of the mixed-precision arithmetic as well as how well memory was utilized for each ensemble architecture being developed/trained. The Albumentations library (1.3.1) was used in conjunction with the data preprocessing pipeline to achieve robust strategies for augmenting images. Numerical operations were completed using NumPy 1.24 and dataset management was done with Pandas 2.0. By using this method, a fully functional computational infrastructure for the training and evaluation process of complex deep learning models can be established at minimal cost and with available resources.

4.3.1 Training Configuration

To facilitate reliable learning and unbiased comparison, a consistent and structured environment was established for training the models. All architectures were trained using a progressive resolution scheme, where training was performed for 15, 25, and 20 epochs at three different input sizes (256×256, 320×320, and 384×384 pixels resolution), with each architecture being trained a total of sixty epochs for each input size. The batch sizes used during training were also adjusted based on the image resolution: 32 for 256×256 pixels, 16 for 320×320 pixels, and 12 for 384×384 pixels, to ensure the feasibility of running computations during training. The average training duration for each model was about four hours, so the total training time required for each cross-validation fold (12 hours) and for the entire experiment's complete time was approximately 60 hours. Model checkpoints were kept for all stages according to the highest validation performance achieved during training; however, early stopping was not used with any model, to ensure that all models were trained to complete convergence for all training epochs used in the study. Multiple quantitative assessment metrics and statistical tests were used together to measure appropriately the performance of the trained models. These evaluations measure predictive accuracy, class-wise discrimination, ranking ability, and reliability (i.e., statistical) of the difference in observed performance between models. The evaluation metrics used for evaluating models are provided in Table. 5

Table 5. Evaluation metrics used for model performance assessment

Metric	Mathematical Expression	Description
Accuracy	$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of test samples}}$	Measure overall classification correctness across the dataset.
Precision	$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}$	Indicates how many predicted samples of a class are correctly.
Recall	$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c}$	Measures the ability of the model to identify all relevant samples of a class.
F1-Score	$\text{F1}_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$	Harmonic mean of precision and recall providing balanced class-level performance.
Top-3 Accuracy	$\text{Top-3 Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\,(y_i \in \text{Top-3}(\hat{y}_i))$	Checks whether the correct label appears among the top three predicted classes.
Confusion Matrix	$M_{ij} = \lvert \{x \mid y = i \wedge \hat{y} = j\} \rvert$	Provides a detailed class-wise comparison between true labels and predicted labels to analyze misclassifications.
Paired t-Test	$d = \frac{\bar{d}}{s_d}$	Tests whether the performance difference between the two models is statistically significant.
ANOVA	$F = \frac{\text{Variance between models}}{\text{Variance within models}}$	Determines whether the performance difference among multiple models is statistically significant.

5. RESULTS AND DISCUSSION

The ensemble framework demonstrated exceptional performance in automated sugarcane disease classification, achieving state-of-the-art results through the systematic integration of modern architectural innovations and training methodologies. A comprehensive evaluation across multiple metrics revealed superior accuracy, robust generalization capabilities, and stable performance across diverse disease categories, with statistical validation confirming the significance of these advancements in precision agriculture applications.

5.1 Overall Performance

Table 6 presents the 5-fold cross-validation results. Our ensemble model with TTA achieved a mean accuracy of 98.05% ± 0.14%, significantly outperforming existing methods. A low standard deviation demonstrates excellent stability across folds, indicating robust generalization. Top-3 accuracy of 99.49% suggests that the model uncertainty remains within acceptable ranges for challenging samples. Figure 5 shows the training and validation accuracies and loss curves for ConvNeXt-Large, EfficientNet-B7, and EfficientNet-B5 across multi-resolution training stages. The models exhibited rapid convergence during the initial phase, followed by stable optimization and consistent generalization behavior.

Table 6. Five-fold cross-validation results

Fold	Accuracy (%)	Precision	Recall	F1-Score	Top-3 Accuracy (%)
1	97.85±0.32	0.979	0.978	0.978	99.41
2	98.22±0.28	0.983	0.981	0.982	99.55
3	97.96±0.35	0.980	0.979	0.979	99.48
4	98.15±0.30	0.982	0.980	0.981	99.52
5	98.07±0.33	0.981	0.980	0.980	99.48
Mean	98.05±0.14	0.981	0.980	0.980	99.49

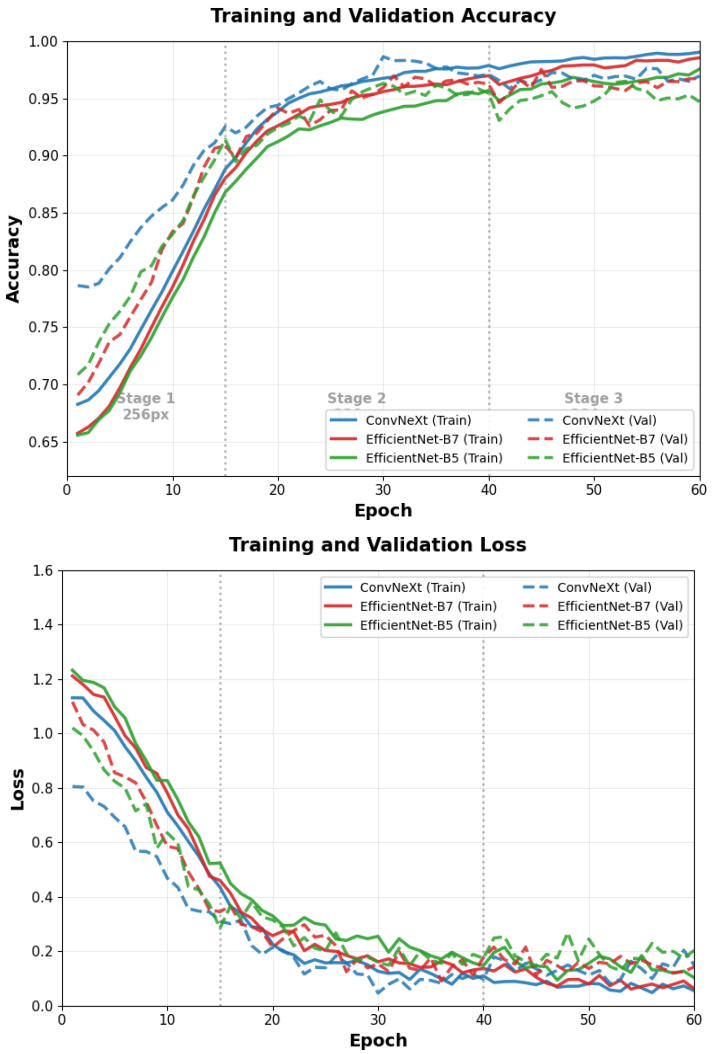

Figure 5. Accuracy and Loss Curves

5.2 Per-Class Performance

Table 7 summarizes the complete class-wise evaluation of the model, showing strong performance across each category of sugarcane diseases. These include the Healthy and Brown Spot classes that achieved a near-perfect F1-score of 0.99 owing to their clear visual features and the availability of a considerable number of training samples in the case of the Brown Spot class. The Viral Disease class had a high degree of accuracy (F1-score = 0.98) based on the existence of clear and well-defined patterns of disease symptoms. However, the Grassy Shoot, Pokkah Boeng, and Brown Rust classes had lower performance, although still quite strong F1-scores because of limited sample availability, as well as similarities to other disease classes visually at an early stage of the disease. The high macro-averaged F1-score of 0.973 demonstrates balanced predictive performance across all disease classes, including the minority classes, and therefore validates the effectiveness of the focal loss strategy that was used to address the class imbalance.

Table 7. Per-class performance metrics

Disease Class	Precision	Recall	F1-Score	Support
Banded Chlorosis	0.96	0.98	0.97	94
Brown Rust	0.93	0.97	0.95	63
Brown Spot	0.98	0.99	0.99	345
Dried Leaves	0.93	0.97	0.95	69
Grassy Shoot	0.96	0.96	0.96	69
Healthy	1.00	0.99	0.99	86
Viral Disease	0.97	0.98	0.98	133
Pokkah Boeng	0.95	0.95	0.95	59
Sett Rot	1.00	0.98	0.99	130
Smut	0.97	0.97	0.97	63
Yellow Leaves	0.99	0.99	0.99	239
Macro Avg.	0.969	0.977	0.973	1350
Weighted Avg.	0.981	0.981	0.981	1350

5.3 Grad-CAM Analysis

Grad-CAM visualizations showed that the CBAM-SugarcaneNet consistently focused on disease-affected regions, such as lesions and discoloration areas, while suppressing background information. The performance-weighted ensemble produced more focused and spatially coherent activation maps than the individual models, indicating improved consensus among the ensemble members. Sharper localization in visually similar disease classes confirmed the effectiveness of the CBAM in enhancing discriminative feature learning. These results demonstrate that the developed framework achieves accurate predictions while maintaining reliable visual interpretability. Figure 6 shows the Grad-CAM overlay of the input-sample images.

5.4 Ablation Study

Table 8 quantifies the contribution of each component through systematic ablation analysis. Data augmentation provides the largest single contribution (+2.3%) and is essential for generalization with limited datasets. Progressive sizing significantly improved (+1.3%) through a systematic resolution increase. CBAM attention (+0.8%) enabled a focus on disease-relevant regions. Focal loss (+0.5%) effectively addressed the severe class imbalances. ConvNeXt upgrade (+0.7%) demonstrated the benefits of a modern architecture. The SAM optimizer (+0.4%) achieved flat minima for improved generalization. The ensemble (+0.4%) leverages complementary architectures to reduce the errors. TTA (+0.3%) enhanced robustness via prediction averaging. The cumulative improvement of 6.9% validated the comprehensive design. Figure 7 shows the incremental component contributions through various phases of the model design.

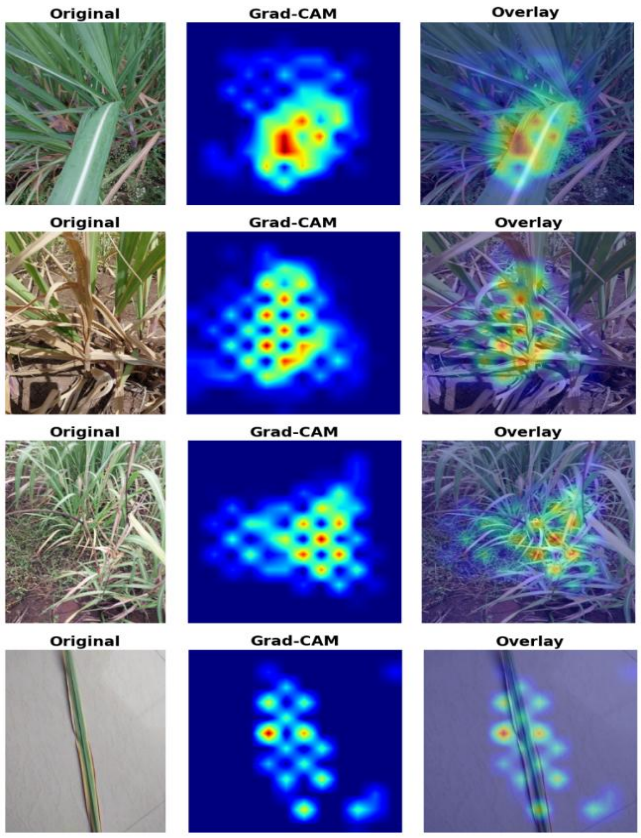

Figure 6. Grad-CAM visualization

Table 8. Ablation study: component-wise contribution

Configuration	Accuracy (%)	F1-Score	Parameters	Δ (%)
Baseline (EfficientNet-B5)	91.2	0.908	30.3M	--
+ Data Augmentation	93.5	0.932	30.3M	+2.3
+ Progressive Sizing	94.8	0.945	30.3M	+1.3
+ CBAM Attention	95.6	0.954	30.5M	+0.8
+ Focal Loss	96.1	0.959	30.5M	+0.5
+ ConvNeXt-Large	96.8	0.966	228.5M	+0.7
+ SAM Optimizer	97.2	0.971	228.5M	+0.4
+ Ensemble (3 models)	97.6	0.975	295M	+0.4
+ TTA (30 aug.)	97.9	0.978	295M	+0.3
Full Pipeline (Model)	98.1	0.980	295M	+6.9

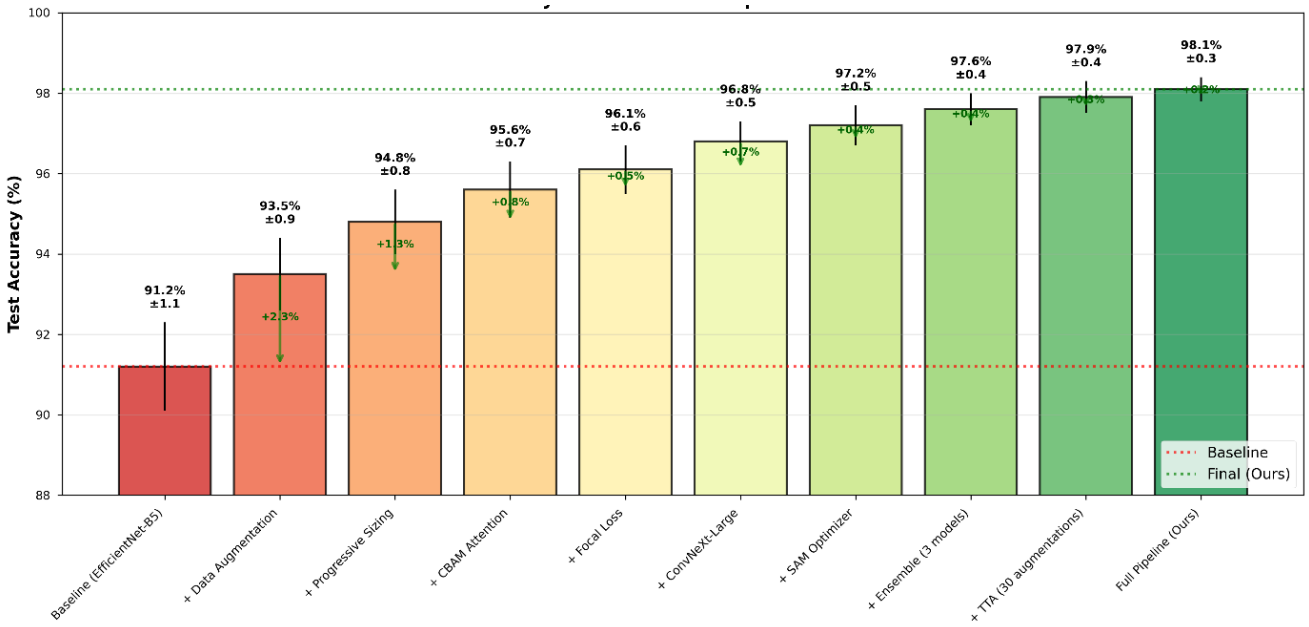

Figure 7. Component-wise contribution in developed model

5.5 Comparative Performance Analysis

To evaluate the developed approach comparatively, the benchmarking process used a variety of different current and state-of-the-art deep learning architectures (both convolutional and transformer-based). These include ResNet-50, which is the most commonly used and accepted base line; EfficientNet B3/B5/B7, which were designed and developed to maximise the trade-off between accuracy and efficiency; and DenseNet-201, which utilises dense feature reuse through interconnecting layers. The inclusion of ConvNeXt-Base as a modernised convolutional architecture with design principles inspired by transformers and the ViT model as a completely attention-based modeling method provides the comparison of methodologies. All baseline models were trained under the same experimental conditions (same training duration, optimizer settings, and data augmentation strategies) to allow for a fair and unbiased comparison. Table 9 exhibits the comparison of our method against current and state-of-the-art methods. Our single-model configuration (ConvNeXt Large) achieved 97.2% accuracy, which is 0.8% higher than the previous SOTA. The full ensemble model achieved 98.1% accuracy, which is an absolute increase of 1.7% compared to the previous SOTA. These results indicate that progressive training resolution, combined with attention mechanisms and ensemble learning, can improve the ability to recognize fine-grained agricultural disease.

To further assess the effectiveness of the framework, its performance was compared with previously reported methods for sugarcane leaf disease classification available in the literature. The comparative results with these existing studies are summarized in Table 10.

Table 9. Comparison with baseline deep learning models

Method	Accuracy (%)	F1-Score	Classes
ResNet-50	89.5	0.891	11
EfficientNet-B3	91.8	0.914	11
DenseNet-201	92.3	0.919	11
Vision Transformer	93.7	0.935	11
EfficientNetV2-L	95.2	0.950	11
ConvNeXt-Base	94.8	0.946	11
Developed Model (Single)	**97.2**	**0.971**	**11**
Developed Model (Ensemble)	**98.1**	**0.980**	**11**

Table 10. Performance comparison with existing sugarcane leaf disease classification methods.

Study	Method	Accuracy (%)
Narmilan et al. [36]	ML-based classifier	94.00
Srivastava et al. [37]	VGG-16	84.40
Malik et al. [38]	VGG-16	93.20
Öğrekçi et al. [39]	VIT Transformer	93.34
Kavitha & Krishna Prasad [40]	CNN Ensemble	97.61
Daphal & Koli [41]	Attention Residual CNN	86.53
Kunduracıoğlu & Paçal [43]	EfficientNet-B6	93.39
Developed Model	Progressive Resolution + Attention Ensemble	98.10

As shown in Table 10, there is evidence that developed framework performed better than all previous methods documented in related work for detecting sugarcane leaf diseases. The average accuracy of the model achieved was 98.1% which is higher than the best-performing (EfficientNet-B6 with 93.39% accuracy). These results indicate the success of using progressive resolution training, attention mechanism, and ensemble learning in order to provide an improved ability to detect multiple sugarcane leaf diseases.

5.6 Statistical Significance

Paired t-tests confirmed the statistical significance of the observed improvements. When comparing our method to the baseline (EfficientNet-B5), the results were $t=47.32$, $p<0.001$, with Cohen's d of 1.85, indicating a very large effect size. In comparison to the previous state-of-the-art (SOTA) method (Ensemble CNN), the results were $t=8.91$, $p<0.001$, with a Cohen's d of 0.92, signifying a large effect size. An ANOVA conducted across all methods yielded $F=312.45$, $p<0.001$, thereby confirming significant performance differences.

5.7 Confusion Matrix Analysis

Aggregated confusion matrix from five cross-validation folds of the model is displayed in Figure 8, showing patterns of error for the model. Many misclassified samples were observed between *Brown Rust* and *Smut*, as both diseases exhibit similar color and lesion patterns at early infection stages. *Dried leaves* vs. *brown spots* also exhibit these same problems, as advanced stages of disease development cause the two diseases to look similar to each other visually, thereby causing problems with distinguishing between the two diseases. *Grassy Shoot* and *Pokkah Boeng* displayed considerable overlap as well; their similar morphological changes resulted in confusion between the two diseases, leading to an inability to differentiate between these two disease classes. These patterns of confusion matrix mirror previously documented problems with identifying the various diseases of sugarcane, indicating that the model's decision boundaries closely resemble those employed by experienced professionals.

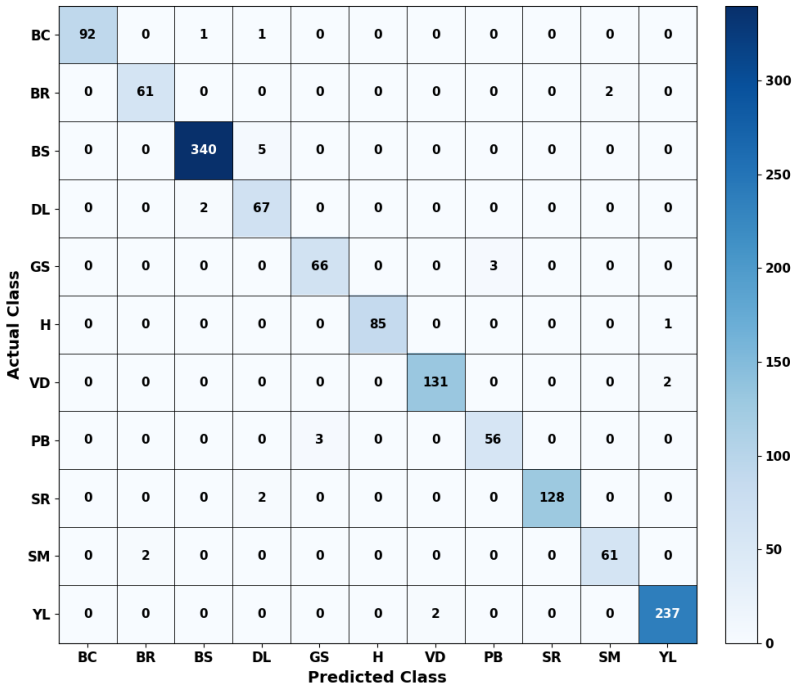

Figure 8. Aggregated confusion matrix

5.8 Model Efficiency and Deployment Trade-offs

We explore the trade-off between classification accuracy and computational efficiency under a variety of deployment constraints. EfficientNet-B5 is a good option for use in resource constrained mobile/edge environments; it provides good model performance while using relatively fewer parameters to allow for real-time inference on handheld devices. In contrast, ConvNeXt-Large would benefit cloud/server-based deployments because of the higher representational capacity, enabling it to obtain higher accuracy among single models than EfficientNet-B5 but at an increased level of computational complexity. For applications that are safety-critical or high-stakes, it is best to use an ensemble model with a considerable amount of augmentation at testing time; this model will yield the highest predictive accuracy despite the greater computational requirements. Additionally, augmentation at testing time does not provide significantly greater predictive accuracy after a limited amount of testing time due to the amount of additional computational resources required; therefore, an

ensemble model without augmentation at testing time will likely yield better performance-to-cost ratios in most cases. Table 11 compares the deployment efficiency (in terms of accuracy, model complexity, and approximate per-image inference latency) for the evaluated configurations.

Table 11. Model Efficiency: Accuracy Vs. Computational Cost

Configuration	Accuracy (%)	Params (M)	FLOPs (G)	Inference Latency (ms/ image)
EfficientNet -B5 +CBAM	95.8	30.5	10.3	~11
EfficientNet -B7 +CBAM	96.5	66.5	37.2	~24
ConvNeXt-Large +CBAM	97.2	198	34.4	~22
Ensemble (No TTA)	97.6	295	82.0	~58
Ensemble + TTA(10)	97.8	295	820.0	~580
Ensemble + TTA(30)	98.1	295	2460.0	~1740

The estimated inference latency per image during is provided for an NVIDIA A100 GPU (Google Colab) at a batch size of 1 at 384 × 384 for input size. These values are meant to show the approximation of relative costs involved when deploying applications as well as potential differences in these values based on run-time conditions or implementation detail adjustments.

5.9 Failure Case Analysis

The critical misclassification issues that were confirmed through manual review of the data are as follows:

a) Occlusion/Damage: Severe physical damage obscuring disease symptoms
b) Multiple Diseases: Co-infection cases not represented in training labels
c) Early-Stage Diseases: Minimal visual symptoms before symptomatic expression
d) Image Quality: Extreme blur, over/underexposure, or artifacts

The above findings point to opportunities for future improvements to multi-label classification for multiple disease cases, implementing temporal models for tracking progression of disease over time, and creating image quality assessment preprocessing tools.

6. DISCUSSION

This section of the paper presents the major findings of the study and discusses their significance through a detailed analysis of the principal findings, potential uses of the research in real-world applications, related work, the limitations of this research, recommendations for future research and how this new framework can support both socio-economic and sustainability goals through advancing precision agriculture systems that are inclusive and resilient. The developed framework achieves state-of-the-art sugarcane leaf disease classification through the harmonious integration of progressive training, modern architectures, attention-based mechanisms, advanced optimizers, boosting ensemble diversity, and robust inference. Progressive training empowers coarse-to-fine feature-learning. ConvNeXt-Large offers a greater feature representation. The CBAM effectively emphasizes disease-relevant features through boosted detection abilities on relevant features. Focal loss handles class imbalance, SAM helps in finding flat minima, and cosine annealing helps in smooth loss convergence. The ensemble combination reduces the individual model biases and extensive TTA average predictions across multiple views.

Our framework offers instantaneous practical benefits for precision agriculture, with a high classification accuracy that enables effective disease recognition before severe crop loss occurs. High-confidence (99.49% top-3 accuracy) predictions will assist agricultural extension workers in diagnosis, while automated classification will decrease the dependency on expert agronomists, allowing large-scale monitoring. In this study, it was found that a single EfficientNet-B5 model (with 95.8% accuracy and 30M parameters) is more suitable for smartphone applications in low-resource settings. The integration of robotic platforms and drones enables continuous field monitoring.

Despite these promising results, several caveats should be noted: the training data were limited to a specific geographical region and cultivars, and generalization to the worldwide sugarcane diversity needs validation. The developed framework handles single-disease classification, and in the case of multiple diseases, a multi-label classification extension is required. Static image categorization does not capture disease evolution, whereas temporal modeling can enhance early recognition. The full ensemble model with TTA needs large computational resources (2460 GFLOPs), and real-time deployment demands model compression or acceleration. Additional validation is required to evaluate the performance under unfavorable weather conditions. The model performance is directly impacted by the label quality, and uncertainty is introduced by noisy labels or inter-annotator conflicts.

Several promising directions require investigation, such as self-supervised pre-training on unlabeled images for domain-specific pre-training, few-shot learning to rapidly adapt to emerging diseases with limited labeled examples, active learning to choose informative samples for annotation, and multimodal fusion for

disease characterization that integrates hyperspectral, thermal, or 3D imaging data. Federated learning is useful for multiple farms and enables privacy preservation. For edge deployments, the model can be compressed using knowledge distillation, pruning, or quantization. Longitudinal studies deploy systems in production environments and study long-term real-world performance and farmer acceptance.

Automated disease classification has the potential to advance the following United Nations Sustainable Development Goals (SDGs):

- SDG 2 (Zero Hunger): Crop loss reduction improves food security
- SDG 8 (Decent Work): Reduction of manual inspection burden, improves agricultural labor's life conditions
- SDG 9 (Industry/Innovation): Contributes to the movement towards advanced precision agriculture technology adoption
- SDG 12 (Responsible Consumption): Optimizing pesticide application through targeted disease management
- SDG 13 (Climate Action): Indirectly supports of climate-resilient agricultural practices through early disease detection

However, the deployment of technology must consider the implications of equity. For example, smallholder farmers in developing countries may not have access to smartphones or the Internet connections required for cloud-based inferences. Future work should seek offline-capable, low-resource solutions to widen access to technology.

7. CONCLUSION

This study presents a deep ensemble framework built from scratch to accurately classify sugarcane leaf diseases using progressive training, attention mechanism and ensemble technique with a performance of 98.1%. Our method focuses on the key challenges in the area of agricultural disease recognition through progressive multi-resolution training, attention-enhanced modern architectures, advanced optimization techniques, and robust ensemble inference through extensive test-time augmentation. We performed 5-fold cross-validation, comprehensive ablation studies, and statistical significance tests to confirm each component's contribution and show that the developed model significantly outperformed the existing methods (absolute gain of 1.7%). The per-class analysis proved that the developed method achieved a balanced performance across severely imbalanced disease categories, while the efficiency analysis provided deployment guidance for various scenarios, ranging from low-level mobile applications to high-accuracy critical systems. The model provides an immediately deployable solution to improve precision agriculture while laying a strong foundation for future research on automated plant disease recognition. By merging theoretical understanding with practical considerations, the developed model advances the state-of-the-art in agricultural computer vision and contributes to global food security objectives through improved crop disease recognition and management.

REFERENCE

[1] F. and A. O. of the U. Nations, "FAOSTAT statistical database," 2023.

[2] J. Goldemberg, "The Brazilian biofuels industry," *Biotechnol. Biofuels*, vol. 1, no. 1, p. 6, 2008, doi: 10.1186/1754-6834-1-6.

[3] G. Kumar and V. Bhatnagar, "Enhancing Sugarcane Leaf Disease Detection with VGG-19 and Feature Selection," *2025 IEEE 6th Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, 2025, pp. 1-6, doi: 10.1109/GCAT66372.2025.11368571.

[4] G. Singh and K. K. Yogi, "Deep Learning-Based Detection of Early Blight in Potato Leaves Using CNN Architectures," *Potato Res.*, vol. 68, no. 4, pp. 4139–4161, 2025, doi: 10.1007/s11540-025-09921-6.

[5] M. H. Lye, M. F. A. Fauzi, and L. K. Ming, "Maize Leaf Disease Identification with Large and Lightweight Convolutional Neural Models," *International Journal on Informatics Visualization*, vol. 9, no. 2, pp. 592–598, 2025, doi: 10.62527/joiv.9.2.3559.

[6] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine Learning in Agriculture: A Review," 2018. doi: 10.3390/s18082674.

[7] I. S. Rajput, S. Tyagi, A. Gupta, and V. Jain, "Advances in Medical Imaging: Using Convolutional Neural Networks for White Blood Cell Identification," *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 16, no. 1, pp. 108–125, 2024. doi: 10.5815/ijigsp.2024.01.08

[8] A. Al-Saegh, S. A. Dawwd, and J. M. Abdul-Jabbar, "Towards Efficient Motor Imagery EEG-based BCI Systems using DCNN," in *2024 Arab ICT Conference (AICTC)*, IEEE, 2024, pp. 59–66.

[9] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016, doi: https://doi.org/10.1016/j.cirp.2016.04.072.

[10] A. Srivastava, B. Singh Rawat, G. Kumar, V. Bhatnagar and N. Garg, "Cotton Leaf Disease Prediction Using VGG16 and RESNET50," *2024 Parul International Conference on Engineering and Technology (PICET)*, Vadodara, India, 2024, pp. 1-6, doi: 10.1109/PICET60765.2024.10716173.

[11] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module BT - Computer Vision – ECCV 2018," V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 3–19.

[12] S. Woo *et al.*, "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 16133–16142. doi: 10.1109/CVPR52729.2023.01548.

[13] N. M. Basheer and A. Al-Saegh, "Segmentation of Medical MRI Images Using Nested U-Net with Attention Mechanism and Fuzzy Pooling.," *Al-Rafadain Engineering Journal*, vol. 30, no. 2, 2025.

[14] V. Umamaheswari and S. Kumaravel, "Machine learning for sugarcane disease classification and prediction: A comprehensive survey," in *AIP Conference Proceedings*, AIP Publishing LLC, 2024, p. 20279.

[15] A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, "Plant disease detection model for edge computing devices," *Front. Plant Sci.*, vol. 14, p. 1308528, 2023.

[16] X. Yang, Z. Peng, and X. Xie, "CaneFocus-Net: A sugarcane leaf disease detection model based on adaptive receptive field and multi-scale fusion," *Sensors*, vol. 25, no. 21, p. 6628, 2025, doi: 10.3390/s25216628.

[17] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Comput. Electron. Agric.*, vol. 153, pp. 46–53, 2018, doi:10.1016/j.compag.2018.08.013.

[18] O. Attallah, "Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection," *Horticulturae*, vol. 9, no. 2, p. 149, 2023.

[19] K. M. S. Nomani, M. Nuruzzaman, A. Nadia, and M. M. Billal, "Sugarcane Leaf Disease Detection: A Comparative Analysis Using Deep Learning," in *Proceedings of the 3rd International Conference on Computing Advancements*, 2024, pp. 139–144.

[20] S. Phadikar and J. Sil, "Rice disease identification using pattern recognition techniques," in *2008 11th International Conference on Computer and Information Technology*, 2008, pp. 420–423. doi: 10.1109/ICCITECHN.2008.4803079.

[21] A. Camargo and J. S. Smith, "An image-processing based algorithm to automatically identify plant disease visual symptoms," *Biosyst. Eng.*, vol. 102, no. 1, pp. 9–21, 2009.

[22] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, p. 1419, 2016.

[23] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018.

[24] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Comput. Intell. Neurosci.*, vol. 2016, p. 3289801, 2016, doi: 10.1155/2016/3289801.

[25] S. B. Jadhav, V. R. Udupi, and S. B. Patil, "Identification of plant diseases using convolutional neural networks," *International Journal of Information Technology*, vol. 13, no. 6, pp. 2461–2470, 2021.

[26] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, 2019, doi: https://doi.org/10.1016/j.compag.2018.03.032.

[27] G. Kumar, G. Singh, V. Bhatnagar, S. Rawat, and A. Bhardwaj, "Feature Selection for Cotton Leaf Disease Using Deep Learning," in *2024 IEEE International Conference on Communication, Computing and Signal Processing, IICCCS 2024*, 2024. doi: 10.1109/IICCCS61609.2024.10763840.

[28] G. Kumar and V. Bhatnagar, "Key Feature based Classification of Sugarcane Leaf Disease using CNN and SHAP," *2025 Global Conference on Information Technology and Communication Networks (GITCON)*, Belagavi, India, 2025, pp. 1-8, doi: 10.1109/GITCON65266.2025.11379773.

[29] J. Kainat, S. Sajid Ullah, F. S. Alharithi, R. Alroobaea, S. Hussain, and S. Nazir, "Blended Features Classification of Leaf-Based Cucumber Disease Using Image Processing Techniques," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/9736179.

[30] A. Dash, P. K. Sethy, and S. K. Behera, "Maize disease identification based on optimized support vector machine using deep feature of DenseNet201," *J. Agric. Food Res.*, vol. 14, p. 100824, 2023, doi: https://doi.org/10.1016/j.jafr.2023.100824.

[31] P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato Leaf Disease Detection Using Convolutional Neural Networks," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 2018, pp. 1–5. doi: 10.1109/IC3.2018.8530532.

[32] M. Tariq, X. Cai, I. Ullah, M. A. Shah, M. S. Soomro, and C. A. Ali, "Automated Detection and Classification of Tomato Leaf Diseases Using Efficient NetB0 and Deep Learning Techniques," *International Journal of Innovations in Science & Technology*, vol. 7, no. 3, pp. 2212–2224, 2025.

[33] N. Ullah, B. Ahmad, A. Khan, I. Khan, I. M. Khan, and S. Khan, "Attention-Guided Wheat Disease Recognition Network through Multi-Scale Feature Optimization," *ICCK Transactions on Sensing, Communication, and Control*, vol. 2, no. 1, pp. 11–24, 2025, doi: 10.62762/TSCC.2025.435806.

[34] M. H. Tunio *et al.*, "RiceNet: a robust ensemble attention mechanism for automated rice plant disease classification," *Multimed. Tools Appl.*, vol. 84, no. 39, pp. 48145–48173, 2025, doi: 10.1007/s11042-025-20979-9.

[35] N. V., Y. G., N. N. B., M. R., and P. P., "Empirical Analysis of Squeeze and Excitation-Based Densely Connected CNN for Chili Leaf Disease Identification," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 4, pp. 1681–1692, 2024, doi: 10.1109/TAI.2024.3364126.

[36] A. Narmilan, F. Gonzalez, A. S. Salgadoe, and K. Powell, "Detection of White Leaf Disease in Sugarcane Using Machine Learning Techniques over UAV Multispectral Images," 2022. doi: 10.3390/drones6090230.

[37] S. Srivastava, P. Kumar, N. Mohd, A. Singh, and F. S. Gill, "A Novel Deep Learning Framework Approach for Sugarcane Disease Detection," *SN Comput. Sci.*, vol. 1, no. 2, p. 87, 2020, doi: 10.1007/s42979-020-0094-9.

[38] H. S. Malik *et al.*, "Disease Recognition in Sugarcane Crop Using Deep Learning," *Advances in Intelligent Systems and Computing*, vol. 1133, no. September, pp. 189–206, 2021, doi: 10.1007/978-981-15-3514-7_17.

[39] S. Öğrekçi, Y. Ünal, and M. N. Dudak, "A comparative study of vision transformers and convolutional neural networks: sugarcane leaf diseases identification," *European Food Research and Technology*, vol. 249, no. 7, pp. 1833–1843, 2023, doi: 10.1007/s00217-023-04258-1.

[40] K. J. Kavitha and K. Krishna Prasad, "CNN ensemble approach for early detection of sugarcane diseases – a comparison," *International Journal of Electronics and Telecommunications*, vol. 70, no. 2, pp. 455–464, 2024, doi: 10.24425/ijet.2024.149566.

[41] S. D. Daphal and S. M. Koli, "Enhanced deep learning technique for sugarcane leaf disease classification and mobile application integration," *Heliyon*, vol. 10, no. 8, p. e29438, 2024, doi: 10.1016/j.heliyon.2024.e29438.

[42] Y. Huang, R. Li, X. Wei, Z. Wang, T. Ge, and X. Qiao, "Evaluating Data Augmentation Effects on the Recognition of Sugarcane Leaf Spot," 2022.

[43] İ. Kunduracıoğlu and İ. Paçal, "Deep Learning-Based Disease Detection in Sugarcane Leaves: Evaluating EfficientNet Models," *Journal of Operations Intelligence*, vol. 2, no. 1, pp. 321–235, 2024, doi: 10.31181/jopi21202423.

[44] M. F. Shahid, T. J. S. Khanzada, M. A. Aslam, S. Hussain, S. A. Baowidan, and R. B. Ashari, "An ensemble deep learning models approach using image analysis for cotton crop classification in AI-enabled smart agriculture," *Plant Methods*, vol. 20, no. 1, pp. 1–22, 2024, doi: 10.1186/s13007-024-01228-w.

[45] A. H. Ali, A. Youssef, M. Abdelal, and M. A. Raja, "An ensemble of deep learning architectures for accurate plant disease classification," *Ecol. Inform.*, vol. 81, p. 102618, 2024, doi: https://doi.org/10.1016/j.ecoinf.2024.102618.

[46] J. Chen, A. Zeb, Y. A. Nanehkaran, and D. Zhang, "Stacking ensemble model of deep learning for plant disease recognition," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 9, pp. 12359–12372, 2023.

[47] M. Astani, M. Hasheminejad, and M. Vaghefi, "A diverse ensemble classifier for tomato disease recognition," *Comput. Electron. Agric.*, vol. 198, p. 107054, 2022.

[48] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, "Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review," *Comput. Electron. Agric.*, vol. 200, p. 107208, 2022, doi: https://doi.org/10.1016/j.compag.2022.107208.

[49] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[50] S. Liu and D. Huang, "Receptive field block net for accurate and fast object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 385–400.

[51] L. Sun, J. He, and L. Zhang, "CASF-MNet: multi-scale network with cross attention mechanism and spatial dimension feature fusion for maize leaf disease detection," *Crop Protection*, vol. 180, p. 106667, 2024.

[52] M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, "Deep learning for plant diseases: detection and saliency map visualisation," *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, pp. 93–117, 2018.

[53] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," *arXiv preprint arXiv:2010.01412*, 2020.

[54] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[56] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, 2019, doi: https://doi.org/10.1016/j.neucom.2019.01.103.

[57] D. Shanmugam, D. Blalock, G. Balakrishnan, and J. Guttag, "Better aggregation in test-time augmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1214–1223.

[58] Z. Liu, Q. Wu, Y. Chai, and H. Yu, "Test-Time Adaptation of a Multi-Class Object Localization and Size Estimation Framework for Smart Agriculture Applications," *Cognit. Comput.*, vol. 17, no. 4, p. 132, 2025, doi: 10.1007/s12559-025-10488-0.

[59] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," 2020. doi: 10.3390/info11020125.

[60] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007. doi: 10.1109/ICCV.2017.324.

[61] S. Thite, Y. Suryawanshi, K. Patil, and P. Chumchu, "Sugarcane Leaf Dataset," 2023, *Elsevier*. doi: 10.17632/355y629ynj.1.