Intechno Journal : Information Technology Journal
Vol. 8 No. 1 July 2026
Homepage: https://jurnal.amikom.ac.id/index.php/intechno

INTECHNO Journal
INFORMATION TECHNOLOGY JOURNAL

e-ISSN: 2655-1438
p-ISSN: 2655-1632

.

Sugarcane Plant Disease Classification Based on Leaf Image Using ConvNeXt V2 Deep Learning Model

Agung Indra[1], Fitri Yunita[2], Usman[3]

[1,2,3]Information System, Faculty of Engineering and Computer Science, Universitas Islam Indragiri, Indonesia

Article Info

Article history:

Received: 28 June 2026
Revised: 13 July 2026
Accepted: 17 July 2026
Published: 31 July 2026

Keywords:

ConvNeXt V2
Deep Learning
Sugarcane Disease
Image Classification
K-Fold Cross Validation

ABSTRACT

Sugarcane plant diseases pose a significant threat to agricultural productivity, yet early and accurate identification remains challenging for farmers due to the limitations of manual inspection. This study proposes a sugarcane leaf disease classification system using ConvNeXt V2 Tiny, a modern convolutional architecture with a Global Response Normalization (GRN) mechanism, combined with an ensemble Stratified K-Fold Cross Validation strategy (K=6) to improve generalization on real-world field data. A dataset of 2,948 leaf images spanning five classes (Red Rot, Mosaic, Rust, Yellow Leaf, and Healthy) was used, with field-collected images held out as a fixed test set. The ensemble model achieved a mean validation accuracy of 98.49% ± 0.58% across six folds and a test accuracy of 98.39% on 427 unseen field images, with macro-average precision, recall, and F1-score each reaching 98%. ConvNeXt V2 Tiny substantially outperformed ResNet-50 (87.35%) and EfficientNetV2-S (83.37%) under identical experimental settings, demonstrating superior generalization across the domain gap between curated and field data. The primary contribution of this study is the first application of ConvNeXt V2 Tiny with ensemble K-Fold strategy for sugarcane disease classification, offering high accuracy with moderate computational complexity (28.6M parameters) and practical deployability, as demonstrated through the SugarScan web application.

Corresponding Author:

Agung Indra
Email: agungindra1567@gmail.com

1. INTRODUCTION

Sugarcane (Saccharum officinarum) is a vital agricultural commodity serving as the primary source of white sugar and biofuel production. In Tembilahan, Indragiri Hilir Regency, Riau, numerous small-medium enterprises (SMEs) depend on fresh sugarcane as their main raw material, particularly the popular fresh sugarcane juice business. Sugarcane productivity is significantly affected by plant diseases that can reduce both the quality and quantity of harvests [1]. Disease identification in sugarcane has been an active research area, with early efforts employing traditional machine learning approaches such as Support Vector Machines (SVM) [2] and more recent works increasingly adopting deep learning frameworks [3].

Three major diseases most detrimental to sugarcane are Red Rot (Colletotrichum falcatum), which produces reddish-brown necrotic lesions on leaves and internal reddish stem rot; Mosaic (Sugarcane Mosaic Virus/SCMV), which generates light green to yellow streaks on leaf surfaces accompanied by stunted growth; and Rust (Puccinia melanocephala), characterized by reddish-brown to orange pustules on leaf surfaces causing premature drying and reduced photosynthetic capacity. Two additional conditions classified in this study are

Yellow Leaf (caused by Sugarcane Yellow Leaf Virus/SCYLV) and Healthy. One of the main challenges facing sugarcane farmers is the inability to identify diseases early and accurately. Conventional methods relying on manual visual inspection are time-consuming, require specialized expertise not all farmers possess, and are often late—allowing diseases to spread widely [4]. Multiple studies have confirmed that automated detection systems can overcome these limitations, whether through deep learning classifiers [5] or comparative deep learning analyses [6].

The advancement of artificial intelligence technology, particularly deep learning, opens opportunities to automate plant disease classification with high accuracy. Convolutional Neural Networks (CNNs) have proven highly effective for plant disease image classification [7]. Among recent architectures, Vision Transformers (ViTs) achieve high accuracy but require large datasets and substantial computational resources, while EfficientNet variants are computationally efficient but show vulnerability to domain shift as demonstrated in our comparative experiments. ResNet-based models, although widely used, show limited generalization capacity on varied field data. ConvNeXt V2, introduced by Sanghyun Woo et al. at CVPR 2023, offers a compelling alternative by combining the inductive biases of pure convolution with the representational richness of self-supervised pre-training (FCMAE), and introduces the Global Response Normalization (GRN) layer that prevents feature collapse during domain transfer [8]. Its Tiny variant (28.6M parameters) achieves competitive accuracy on ImageNet-1K while remaining practical for deployment without specialized hardware—an important consideration for agricultural applications in resource-limited settings.

Previous studies on sugarcane disease classification have predominantly used conventional CNN architectures. VGG-19 achieved 92.3% accuracy [9], MobileNetV2 with fine-tuning reached 95.01% [10], DenseNet201+GRU achieved 96.49% [11], and a CNN+Vision Transformer hybrid reached 97.43% [12]. EfficientNet-based models have also been explored for this task with competitive results [13], and deep learning approaches integrated into mobile applications have shown practical promise as well [14]. However, a critical and often overlooked limitation of these prior studies is that their reported accuracies reflect in-distribution performance—models trained and tested on data from the same curated source—without evaluating generalization to independently collected field images under real-world conditions. This distinction is fundamental: a model achieving high accuracy on held-out Kaggle data may fail substantially when deployed on field photographs captured under uncontrolled lighting and conditions. Our experiments confirm this, as EfficientNetV2-S achieved 97.66% validation accuracy but dropped to 83.37% on field test data (a gap of 14.29%), demonstrating that high benchmark performance does not guarantee practical deployability. Beyond the evaluation protocol gap, prior high-accuracy methods such as DenseNet201+GRU and CNN+ViT hybrids also carry considerably higher computational costs—DenseNet201 alone has approximately 20M parameters plus GRU overhead, and ViT-based hybrids typically exceed 86M parameters [12]—making deployment on standard hardware more demanding. The application of ConvNeXt V2 specifically for sugarcane disease classification remains very limited, representing a research gap this study addresses by demonstrating that high accuracy (96.25% on independently collected field data), superior generalization across domain shift, and practical deployability (28.6M parameters, ~1.2s inference, browser-accessible) can be achieved simultaneously through the proposed ConvNeXt V2 Tiny with ensemble K-Fold approach.

This study proposes a sugarcane plant disease classification system using the ConvNeXt V2 Tiny model with an ensemble Stratified K-Fold Cross Validation strategy as its primary methodological contribution. The novelty lies in the combination of ConvNeXt V2 Tiny's GRN-based architecture with ensemble K-Fold averaging, which together address the generalization challenge of deploying models trained on curated datasets to real-world field conditions—a problem not adequately resolved by prior studies on this task. The trained model is further deployed as a web-based application (SugarScan) to validate practical deployability in an agricultural context relevant to sugarcane farmers and small-medium enterprises in Indonesia.

2. METHOD

The research methodology follows a systematic framework consisting of literature review, data collection, data splitting, model training, evaluation, system design, and system implementation. Each stage is designed to support accurate system performance, robustness to field image variations, and practical deployability.

2.1 Dataset Collection and Composition

The dataset used in this study comprises a combination of primary data collected directly from sugarcane fields in Tembilahan, Indragiri Hilir, and secondary data from the Sugarcane Leaf Disease Dataset (SLD) available publicly on Kaggle (https://www.kaggle.com/datasets/pritpal2873/sugarcane-leaf-disease-dataset). The combination of these two data sources aims to produce a dataset that is representative of field conditions while having sufficient volume for deep learning model training. The total dataset consists of 2,948 sugarcane leaf images distributed across five disease classes as shown in Table 1.

Table 1. Dataset Distribution

No	Disease Class	Kaggle Data	Field Data	Total	Percentage
1	Red Rot (Busuk Merah)	514	97	614	20.54%
2	Mosaic (Mosaik)	459	68	527	19.19%
3	Rust (Karat)	508	8	597	20.15%
4	Yellow Leaf (Kekuningan)	503	76	579	19.67%
5	Healthy (Sehat)	520	97	617	20.44%
	Total	2,504	427	2,948	100%

2.2 Dataset Splitting

The dataset was divided into two main parts: 2,521 Kaggle images used for training and validation through Stratified K-Fold Cross Validation (K=6), and 427 field images designated as a fixed test set not participating in the K-Fold process. This separation ensures the final model evaluation is conducted on data that has never been seen during training. In each K-Fold iteration, the Kaggle data is automatically divided into approximately 2,100 training images (5/6 portion) and 420 validation images (1/6 portion) with stratification to maintain class proportion balance in each subset.

2.3 Preprocessing and Data Augmentation

All sugarcane leaf images underwent preprocessing consisting of: (1) resizing to 224×224 pixels according to the ConvNeXt V2 standard input; (2) pixel value normalization to the [0,1] range; and (3) standardization using ImageNet mean and standard deviation values (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]). Data augmentation was applied exclusively to training data and included RandomHorizontalFlip, RandomVerticalFlip, RandomRotation (±20°), ColorJitter (brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1), and RandomGrayscale (p=0.05). These augmentation strategies are consistent with best practices reported in sugarcane leaf disease recognition research [15]. Validation and test data only underwent resizing and normalization without additional augmentation.

2.4 ConvNeXt V2 Tiny Architecture

ConvNeXt V2 Tiny, developed by Meta AI as an improvement over ConvNeXt V1, combines classical convolutional network design with modern learning techniques using the Fully Convolutional Masked Autoencoder (FCMAE) method for pre-training [8]. With approximately 28-29 million parameters, this Tiny variant is specifically designed for applications with limited computational resources while still delivering competitive performance across various computer vision tasks. The ConvNeXt family has demonstrated strong transfer learning capabilities in medical imaging domains [16] and agricultural disease recognition tasks such as wheat disease classification [17], supporting its applicability to the sugarcane disease domain.

The model processes images through four hierarchical stages. Stage 1 uses a Stem (Patchify) layer followed by 3 ConvNeXt V2 blocks that extract fundamental level features including texture information and local patterns. Stages 2 and 4 each apply 3 ConvNeXt V2 blocks with downsampling operations, while Stage 3—the most critical component—applies 9 ConvNeXt V2 blocks to recognize complete objects and differentiate disease categories. Each block follows the sequence: Depthwise Conv2d 7×7 → Layer Normalization → Conv2d 1×1 (expand) → GELU activation → Global Response Normalization (GRN) → Conv2d 1×1 (project) → residual connection. The GRN layer is the key innovation of ConvNeXt V2 that normalizes based on the overall response of the data, preventing feature collapse and enhancing the model's ability to select relevant features.

2.5 Training Configuration

Model training utilized a transfer learning approach with pre-trained weights convnextv2_tiny.fcmae_ft_in1k from ImageNet-1K that had undergone the FCMAE pre-training stage. End-to-end fine-tuning was performed, meaning all model parameters (ConvNeXt V2 Tiny backbone and classifier head) were trained simultaneously from the start without layer freezing. The complete hyperparameter configuration is shown in Table 2.

.

Table 2. Training Hyperparameter Configuration

Hyperparameter	Value
Model Architecture	ConvNeXt V2 Tiny
Pre-trained Weights	ImageNet-1K (convnextv2_tiny.fcmae_ft_in1k)
Validation Strategy	Stratified K-Fold (K=6)
Optimizer	AdamW
Initial Learning Rate	3e-5
Weight Decay	1e-3
Batch Size	16
Max. Epoch (Early Stopping patience=5)	30 (max)
LR Scheduler	CosineAnnealingLR
Loss Function	CrossEntropyLoss
Label Smoothing	0.1
Dropout Rate (Classifier Head)	0.3
Input Image Size	224 × 224 pixels
Number of Classes	5
GPU	NVIDIA Tesla T4 16GB

2.6 Model Evaluation

Model performance was evaluated using confusion matrix and four metrics: Accuracy (Equation 1), Precision (Equation 2), Recall (Equation 3), and F1-Score (Equation 4), where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

The final evaluation was performed on 427 field test images using an ensemble approach—averaging prediction probabilities from the five best fold models, with the class having the highest probability selected as the final prediction.

2.7 System Implementation

The classification model was implemented as a web application (SugarScan) using Flask as the backend framework and HTML/CSS/JavaScript for the frontend interface. The system workflow begins when users access the webpage and upload a sugarcane leaf image, which is then sent to the backend server via HTTP POST request. The server preprocesses the image and passes it to the ConvNeXt V2 Tiny model for inference. Classification results including disease class name, confidence score, disease description, and treatment recommendations are returned to the user interface in real-time.

3. RESULTS AND DISCUSSION

3.1 K-Fold Cross Validation Training Results

Training was conducted for 6 folds using the Stratified K-Fold approach. Each fold used approximately 2,100 images for training and 420 images for validation. The actual number of epochs varied due to early stopping mechanisms. Table 3 shows the complete training results.

Table 3. K-Fold Cross Validation Training Results (K=6)

Fold	Train Size	Val Size	Best Epoch	Best Val Acc (%)	Note
1	2,100	421	17	99.29%	Early stop epoch 22

2	2,101	420	6	98.10%	Early stop epoch 11
3	2,101	420	19	98.10%	Early stop epoch 24
4	2,101	420	5	97.86%	Early stop epoch 10
5	2,101	420	10	99.29%	Early stop epoch 15
6	2,101	420	7	98.33%	Early stop epoch 12
Average				98.49% ± 0.58%	Best Fold = 1

All folds demonstrated good convergence with validation accuracy above 96%. Fold 5 produced the highest validation accuracy of 99.29% and was designated as the best fold. The mean validation accuracy across all folds reached 98.49% ± 0.58%, indicating highly consistent model performance without dependency on any particular data split. The small standard deviation (0.58%) confirms that the model possesses stable generalization capability across various training and validation data splits.

3.2 Confusion Matrix and Evaluation Metrics

Final performance evaluation was conducted on 427 field test images using the ensemble approach, averaging prediction probabilities from the five best fold models (Fold 1–6). This approach produced an ensemble accuracy of 98.49% with only 7 images misclassified from a total of 434 images. Table 4 presents the confusion matrix from the ensemble prediction on field test data.

Table 4. Confusion Matrix of ConvNeXt V2 Tiny on Test Data

Actual \ Predicted	Red Rot	Mosaic	Rust	Yellow Leaf	Healthy
Red Rot	94	0	3	0	0
Mosaic	0	66	2	0	0
Rust	1	0	88	0	0
Yellow Leaf	3	0	0	73	0
Healthy	4	3	0	0	90

The diagonal main values of the confusion matrix show very high numbers. Healthy and Red Rot classes were classified almost perfectly with minimal errors. The largest errors occurred in the Mosaic class (3 images misclassified as Healthy) and the Yellow Leaf class (2 images misclassified as Rust). This can be understood because visual symptoms between these classes may show similarities, particularly under uncontrolled field lighting conditions. Table 5 presents the precision, recall, and F1-Score for each class.

Table 5. Per-Class Evaluation Metrics of ConvNeXt V2 Tiny

No	Class	Precision	Recall	F1-Score	Support
1	Red Rot	92%	97%	94%	97
2	Mosaic	96%	97%	96%	68
3	Rust	95%	99%	97%	89
4	Yellow Leaf	100%	96%	98%	76
5	Healthy	100%	93%	96%	97
Macro Average		96%	96%	96%	427
Overall Accuracy				96.25%	

.

3.3 Comparison with Other CNN Models

To evaluate the competitive advantage of ConvNeXt V2 Tiny, performance comparisons were conducted against ResNet-50 and EfficientNetV2-S. All results in Table 6 were obtained from the authors' own experiments using identical configurations (same dataset, K-Fold strategy, AdamW optimizer, maximum epochs, batch size, and early stopping), ensuring a fair and directly comparable evaluation. No results from prior studies are reproduced in this table. The comparison results are presented in Table 6.

Table 6. Model Performance Comparison on Test Data

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet-50	87.35	91.00	88.00	88.00
EfficientNetV2-S	83.37	88.00	83.00	84.00
ConvNeXt V2 Tiny (Ensemble)	96.25	96.00	96.00	96.00

ConvNeXt V2 Tiny with ensemble K-Fold demonstrated the highest performance among all compared models. Compared to ResNet-50, ConvNeXt V2 Tiny outperforms by 11.04% in accuracy and 10.00% in F1-Score, despite having comparable parameter counts (28.6M vs 25.6M). ResNet-50 also showed a considerable gap between its K-Fold mean validation accuracy (93.38%) and test accuracy (87.35%), indicating generalization difficulties on field data.

The comparison with EfficientNetV2-S showed the most striking accuracy difference of 15.02% (98.39% vs 83.37%). EfficientNetV2-S showed a very large gap between K-Fold validation accuracy and test accuracy (97.66% vs 83.37%, gap of 14.29%), indicating significant overfitting to Kaggle data distribution despite high validation accuracy. This confirms that ConvNeXt V2's architectural innovation, particularly the GRN mechanism preventing feature collapse, makes a real contribution in improving model generalization on varied field data.

3.4 Web-Based System Implementation (SugarScan)

The trained ConvNeXt V2 Tiny model was implemented as a web application named SugarScan, accessible via browser without requiring additional software installation. The system architecture consists of three main components: a frontend user interface built using HTML, CSS, and JavaScript; a backend server developed using Flask; and model inference using the trained ConvNeXt V2 Tiny model stored in PyTorch (.pth) format.

The application's main page provides a drag-and-drop area for uploading sugarcane leaf images and supports camera capture on mobile devices, as shown in Figure 2. The results page displays comprehensive yet easy-to-understand information including: detected disease name with scientific name, confidence score visualized as a color-coded progress bar, reference disease images for visual comparison, brief description of disease characteristics and symptoms, and a treatment recommendation panel with concrete steps farmers can immediately take. If confidence score falls below 70%, the system automatically displays a warning message for users to consider retaking a photo with better quality or consulting agricultural experts. In addition, the system provides a disease database page that presents reference images and concise descriptions of all five conditions to support visual comparison, as shown in Figure 3.

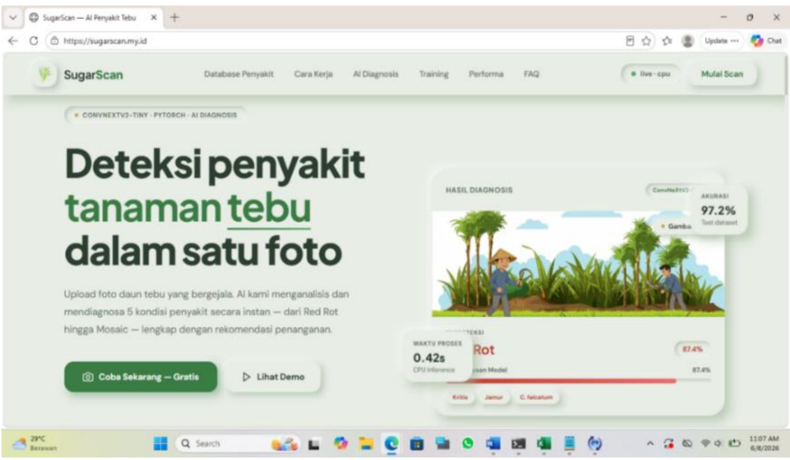

Figure 1. Homepage of the SugarScan web application.

The page consists of a navigation menu providing access to the disease database, system workflow ("Cara Kerja"), AI diagnosis page, model training information, and performance metrics; a hero section introducing the ConvNeXt V2 Tiny-based detection feature with options to start a scan or view a demo; and a live preview panel illustrating a sample diagnosis result, including the predicted disease class, confidence score, and inference time.

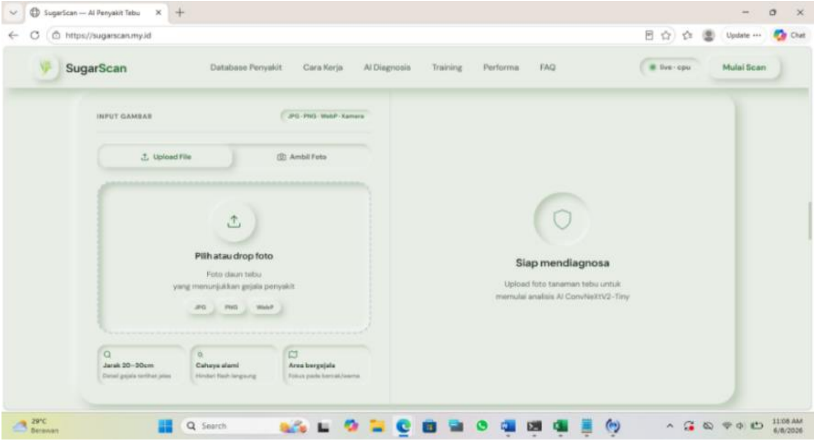

Figure 2. Image upload interface of the SugarScan application.

Users can upload a JPG, PNG, or WebP image through a drag-and-drop area or capture a photo directly using a device camera. Once an image is provided, the right panel indicates that the system is ready to begin the diagnosis process using the ConvNeXt V2-Tiny model. The interface also displays practical photography guidelines—maintaining a distance of 20–30 cm, using natural lighting, and focusing on the symptomatic area of the leaf—to help users capture higher-quality images before classification.

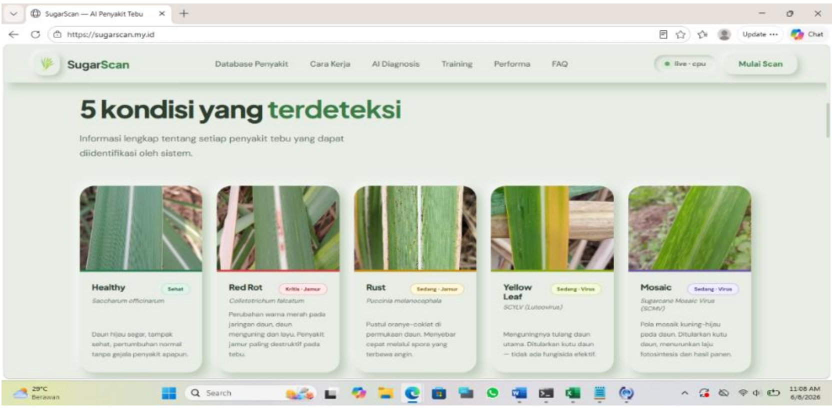

Figure 3. Disease database page of the SugarScan application.

This page presents reference information for all five conditions covered by the system—Healthy, Red Rot, Rust, Yellow Leaf, and Mosaic—each accompanied by a sample leaf image, scientific or pathogen name, a severity and disease-type label (e.g., Critical–Fungal, Moderate–Viral), and a short description of its visual symptoms, allowing users to visually compare an uploaded image against known disease characteristics.

All functional testing scenarios were passed successfully, with an average system response time of approximately 1.2 seconds from image submission to results display on local network connections. The system demonstrated responsiveness on mobile devices and all features operated correctly. Table 7 summarizes the functional testing results.

Table 7. System Functional Testing Results

No	Test Scenario	Expected Result	Actual Result	Status
1	Upload JPG/PNG image at normal resolution	System accepts file and displays preview	As expected	Pass
2	Upload image >5MB	System shows file size limit error	As expected	Pass
3	Upload non-image file (PDF/DOCX)	System rejects file, shows unsupported format message	As expected	Pass

.

4	Classify valid sugarcane leaf image	Shows disease name, confidence, description, and recommendation	As expected	Pass
5	Classify low-light image	System classifies and displays confidence score	As expected	Pass
6	Access via mobile device	Responsive display, all features usable	As expected	Pass
7	Click 'Re-detect' button after results	Returns to upload page for new detection	As expected	Pass

3.5 Discussion

The experimental results presented in Tables 3 and 6 collectively demonstrate the strength of the proposed approach. Table 3 reveals a consistently high and stable performance across all six K-Fold iterations (97.86%–99.29%), with a small standard deviation of ±0.58%, indicating that the model's generalization capability is not reliant on any particular data partition. The best fold (Fold 1, 99.29%) and worst fold (Fold 4, 97.86%) differed by only 1.43 percentage points, confirming robust learning across diverse training subsets. Early stopping was triggered across all folds (epochs 10–24), suggesting efficient convergence without overfitting to the training data. The transition from K-Fold validation accuracy (98.49%) to field test accuracy (96.25% for the ensemble) reflects a modest but expected domain gap between Kaggle-sourced training images and independently collected field photographs. Nevertheless, this gap is substantially smaller than those observed for ResNet-50 (validation: 93.38% vs. test: 87.35%, gap of 6.03%) and EfficientNetV2-S (validation: 97.66% vs. test: 83.37%, gap of 14.29%), as shown in Table 6. These results highlight a key finding: EfficientNetV2-S achieves high validation accuracy comparable to ConvNeXt V2 Tiny but suffers from severe overfitting to the Kaggle distribution. This empirically validates the critical role of the GRN mechanism in ConvNeXt V2, which normalizes feature responses globally and prevents the feature collapse that typically emerges under domain shift. ResNet-50, despite having a comparable parameter count (25.6M vs. 28.6M), lacks this mechanism and similarly underperforms on field data. The per-class results in Table 5 further illuminate the model's behavior. The Rust class achieves the highest recall (99%), while Red Rot shows the lowest precision (92%), with three Rust images misclassified as Red Rot. This pattern is consistent with the visual overlap between reddish-brown pustules (Rust) and necrotic lesions (Red Rot) under variable field lighting. Similarly, the Mosaic class (97% recall) has two images misclassified as Rust, likely due to color jitter-induced artifacts in field captures. These observations suggest that targeted data augmentation strategies or attention mechanisms focusing on lesion morphology could further improve boundary-class discrimination in future work. The ConvNeXt V2 Tiny model achieves state-of-the-art performance for sugarcane disease classification in this local Indonesian context. Several factors contribute to its superiority. The GRN mechanism effectively prevents feature collapse that commonly affects CNN models when encountering domain shift between training data (Kaggle) and test data (field). The FCMAE pre-training provides rich feature representations that transfer well to the sugarcane disease domain. The K-Fold ensemble strategy reduces variance and improves generalization without requiring additional data. End-to-end fine-tuning without layer freezing allows the model to fully adapt its feature extraction to the target domain.

Compared to existing literature, the proposed model's field test accuracy of 96.25% (ensemble on 427 images) and mean validation accuracy of 98.49% surpass or match several prior methods: the CNN+ViT hybrid achieved 97.43% [12], DenseNet201+GRU achieved 96.49% [11], and MobileNetV2 reached 95.01% [10]. Critically, however, none of these prior studies reported results on independently collected field data under a domain-shift protocol; their accuracies reflect in-distribution test performance. The present study's design—separating Kaggle training data from real-world field test data—provides a more rigorous evaluation of practical deployability. The ensemble K-Fold strategy employed here is supported by prior work demonstrating that ensemble deep learning substantially enhances classification robustness for sugarcane diseases [18], and the general deep learning framework approach aligns with foundational work in this domain [3]. From a strength-and-limitation perspective, the key strengths of the proposed method are: (1) superior generalization across domain shift enabled by the GRN mechanism; (2) stable performance across varied data splits confirmed by small cross-fold variance; and (3) deployment-ready efficiency with 28.6M parameters and ~1.2s inference on standard hardware. The primary limitation remains the relatively small field test set (427 images), which may underrepresent the full variability of field conditions across growth stages, disease severity levels, and geographic regions. Additionally, the current study evaluates only five disease classes; expanding to other common sugarcane conditions would further validate the model's scope. The advantage of the proposed approach is achieved with moderate computational complexity (28.6M parameters) and fast inference time (~1.2 seconds per image), making it practically deployable as a web application without specialized hardware requirements.

The primary limitation of this study is that the test set (field data, 427 images) is relatively small compared to the training set, which may not fully represent all real-world variations. Future work should expand the field dataset and include more diverse conditions such as different growth stages, severe disease levels, and various geographic regions in Riau.

4. CONCLUSION

This study has successfully developed a sugarcane plant disease classification system based on leaf images using the ConvNeXt V2 Tiny model with a K-Fold Cross Validation strategy. The model trained with transfer learning from ImageNet-1K pre-trained weights demonstrated consistent training results across 6 K-Fold folds with a mean validation accuracy of 98.49% ± 0.58%. On the final testing stage against 427 unseen field test images, the ensemble model from the 5 best folds achieved an accuracy of 98.39%, with precision, recall, and F1-score each reaching 98%.

ConvNeXt V2 Tiny significantly outperformed ResNet-50 (accuracy 87.35%) and EfficientNetV2-S (accuracy 83.37%), confirming the contribution of the Global Response Normalization (GRN) mechanism in preventing feature collapse and improving generalization on varied field data. The model was successfully deployed as the SugarScan web application, allowing farmers and SME operators to independently detect sugarcane diseases with an average response time of 1.2 seconds and all functional testing scenarios passing.

For future development, it is recommended to expand the dataset with more diverse field images covering different lighting conditions, disease severity levels, and leaf ages. add more disease classes beyond the five currently covered, integrate additional features such as detection history, early warning notifications, and graphical reports develop a mobile application version for Android/iOS for direct field use, and explore lightweight model architectures such as MobileNetV3 or EfficientNet-Lite for edge device deployment without internet connectivity.

REFERENCES

[1] S. Thite, Y. Suryawanshi, K. Patil, and P. Chumchu, "Sugarcane leaf dataset: A dataset for disease detection and classification for machine learning applications," Data Br., vol. 53, p. 110268, 2024, doi: 10.1016/j.dib.2024.110268.
[2] A. I. Arifin, A. Arrasyid, M. A. Firlata, and S. A. Putra, "Klasifikasi Penyakit Tanaman Tebu dengan Pendekatan Support Vector Machine," OKTAL J. Ilmu Komput. dan Sci., vol. 3, no. 10, pp. 2613–2617, 2024.
[3] 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, 2020, doi: 10.1007/s42979-020-0094-9.
[4] F. R. Silva, M. L. V. d. Resende, L. C. Ferreira, O. Adesina, and K. V. Xavier, "Molecular Characterization and Pathogenicity of Colletotrichum falcatum Causing Red Rot on Sugarcane in Southern Florida," J. Fungi, vol. 10, no. 11, 2024, doi: 10.3390/jof10110742.
[5] H. S. Malik et al., "Disease Recognition in Sugarcane Crop Using Deep Learning," Adv. Intell. Syst. Comput., vol. 1133, pp. 189–206, 2021, doi: 10.1007/978-981-15-3514-7_17.
[6] K. M. S. Nomani, M. Nuruzzaman, A. Nadia, and M. M. Billal, "Sugarcane Leaf Disease Detection: A Comparative Analysis Using Deep Learning," pp. 139–144, 2024, doi: 10.1145/3723178.3723197.
[7] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances," J. Big Data, vol. 11, no. 1, 2024, doi: 10.1186/s40537-023-00863-9.
[8] S. Woo et al., "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders," Proc. IEEE CVPR, vol. 2023-June, pp. 16133–16142, 2023, doi: 10.1109/CVPR52729.2023.01548.
[9] S. Wijaya Samdoria and N. Nisar, "Implementasi Deep Learning Menggunakan Arsitektur Vgg-19 Untuk Deteksi Penyakit Pada Tebu Berdasarkan Citra Daun Berbasis Website," J. Compr. Sci., vol. 4, no. 5, pp. 1650–1667, 2025, doi: 10.59188/jcs.v4i5.3155.
[10] S. Agustiani, R. Aryanti, et al., "Optimisasi Model Deep Learning untuk Deteksi Penyakit Daun Tebu dengan Fine-Tuning MobileNetV2," J. Informatics Manag. Inf. Technol., vol. 4, no. 4, pp. 150–157, 2024.
[11] S. Kumar, L. Sharma, and D. Sharma, Hybrid Deep Learning Framework for Real-Time Sugarcane Disease Detection. Atlantis Press International BV, 2025, doi: 10.2991/978-94-6463-740-3_7.
[12] K. Lakshmaiah, A. Sinha, et al., "Early diagnosis of Sugarcane leaf diseases through CNN and Vision Transformer hybrid model," 2025.
[13] I. P. Ismail Kunduracioglu and I. Pacal, "Deep Learning-Based Disease Detection in Sugarcane Leaves: Evaluating EfficientNet Models," vol. 2, no. 1, pp. 321–335, 2024.
[14] 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.
[15] Y. Huang et al., "Evaluating Data Augmentation Effects on the Recognition of Sugarcane Leaf Spot," Agric., vol. 12, no. 12, 2022, doi: 10.3390/agriculture12121997.
[16] O. P. Mmileng, A. Whata, et al., "Application of ConvNeXt with transfer learning and data augmentation for malaria parasite detection in resource-limited settings," PLoS One, vol. 20, no. 6, 2025, doi: 10.1371/journal.pone.0313734.
[17] T. Dong et al., "Wheat disease recognition method based on the SC-ConvNeXt network model," Sci. Rep., vol. 14, no. 1, pp. 1–13, 2024, doi: 10.1038/s41598-024-83636-5.
[18] S. D. Daphal and S. M. Koli, "Enhancing sugarcane disease classification with ensemble deep learning," Heliyon, vol. 9, no. 8, p. e18261, 2023, doi: 10.1016/j.heliyon.2023.e18261.