A Controlled Empirical Study of Loss Functions for Imbalanced Cassava Leaf Disease Classification

Thanh-Hai Tong-Le [1,2] and Thanh-Nghi Doan [3,4,*]

[1] Office of Infrastructure Management, Tien Giang University, Dong Thap, Vietnam
[2] College of Engineering and Technology, Tra Vinh University, Vinh Long, Vietnam
[3] Faculty of Information Technology, An Giang University, An Giang, Vietnam
[4] Vietnam National University, Ho Chi Minh City, Vietnam
Email: tonglethanhhai@tgu.edu.vn (T.-H.T.-L.); dtnghi@agu.edu.vn (T.-N.D.)
*Corresponding author

***Abstract*—Class imbalance remains a critical challenge in agricultural image classification, where standard Cross-Entropy (CE) training tends to favor frequent classes at the expense of rare but agronomically important disease categories. This study presents a controlled empirical comparison of four loss functions: Cross-Entropy (CE), Weighted Cross-Entropy (W-CE), Class-Balanced Loss (CB), and Focal Loss (FL) for five-class cassava leaf disease classification on the iCassava 2019 dataset (imbalance ratio = 8.41). Five Convolutional Neural Network (CNN) backbones (LeNet, ResNet-50, EfficientNet-B0, MobileNetV2, and ConvNeXt-Base) share a common Multilayer Perceptron (MLP) classifier head and follow matched data splits, augmentation policies, and optimization schedules to reduce confounding factors in the loss-function comparison. On the held-out test set (N = 1885), ConvNeXt-Base with focal loss and horizontal-flip Test-Time Augmentation (TTA) attains the highest observed Macro-F1, 89.39%, together with 91.67% accuracy, improving minority-class F1 for Cassava Bacterial Blight (CBB) (+2.57%) and Healthy (+2.38%) relative to the CE baseline while maintaining stable Cassava Mosaic Disease (CMD) performance. Class-balanced loss achieves a nearly identical result on ConvNeXt-Base (89.37% Macro-F1), whereas both W-CE and CB degrade performance on compact backbones (LeNet: −6.5 to −7.3% Macro-F1). These results suggest that explicit class weighting can make optimization less stable in low-capacity feature spaces. Within the scope of a single dataset and a single random seed, the findings indicate that the effectiveness of loss-function reweighting depends on backbone capacity: sample-level modulation through focal loss generally provides more consistent improvements than fixed class-level weighting, with the clearest gains on higher-capacity architectures.**

***Keywords*—cassava leaf disease, imbalanced learning, focal loss, class-balanced loss, plant disease detection, agricultural image classification**

I. Introduction

Convolutional Neural Networks (CNNs) are standard architectures for visual recognition tasks [1, 2]. In agriculture, image-based classification provides a scalable tool for monitoring plant health and reduces the need for manual field inspections [3, 4]. For staple crops such as cassava (Manihot esculenta), which is susceptible to various viral and bacterial infections [5], automated symptom recognition helps farmers manage diseases at an early stage.

A common issue in agricultural computer vision is class imbalance. In natural datasets, localized diseases (e.g., Cassava Bacterial Blight) occur less frequently than widespread infections (e.g., Cassava Mosaic Disease). When training with standard Cross-Entropy (CE) loss, majority classes dominate the gradient updates. This often yields high overall accuracy but poor performance on minority classes, resulting in low Macro-F1 scores [6, 7]. Modifying the loss function is a direct way to address class imbalance without altering the data distribution through explicit resampling [8–10]. However, existing studies on loss functions for cassava disease classification typically vary multiple experimental factors simultaneously—such as backbone architecture, data split, and augmentation policy—making it difficult to attribute performance differences solely to the loss function. Moreover, principled class-level weighting methods (e.g., class-balanced loss) have not been systematically compared with sample-level modulation approaches (e.g., focal loss) under strictly controlled conditions on this benchmark.

Accordingly, the novelty of this study lies not in proposing a new loss function or dataset, but in providing a controlled four-loss comparison under a matched experimental protocol that isolates the effect of loss-function choice from architectural and training

Manuscript received March 9, 2026; revised April 2, 2026; accepted May 6, 2026; published August 12, 2026.

confounders while yielding practical guidance for imbalanced cassava disease classification.

In this study, we evaluate the impact of four loss functions within a controlled experimental framework. Fig. 1 illustrates the training pipeline. We train five CNN architectures, ranging from LeNet to ConvNeXt-Base. These models share the same classification head and follow the same hyperparameter strategy. After establishing CE baselines, we compare Weighted Cross-Entropy (W-CE), Class-Balanced loss (CB) [8], and Focal Loss (FL) [11] using $\gamma = 2.5$. FL down-weights well-classified majority instances, thereby forcing the model to focus more strongly on hard and minority samples. We also evaluate Test-Time Augmentation (TTA) as a complementary inference-time strategy. The contributions of this study are fourfold:

- We establish a controlled experimental protocol in which five backbone families share the same classifier head, data splits, augmentation policy, and optimization schedule, thereby isolating the impact of the loss function from confounding architectural differences.
- We extend the comparison beyond cross-entropy and focal loss to include weighted cross-entropy and class-balanced loss [8], providing a broader perspective on loss-level imbalance mitigation strategies.
- We observe a notable interaction between backbone representational capacity and loss-function effectiveness: focal loss improves minority-class F1 more clearly on higher-capacity backbones (ConvNeXt-Base, ResNet-50), whereas explicit class-level weighting (W-CE, CB) tends to reduce performance on lower-capacity models in the current setup.
- We evaluate horizontal-flip test-time augmentation as a complementary inference-time technique and quantify the resulting accuracy–efficiency trade-off across architectures.

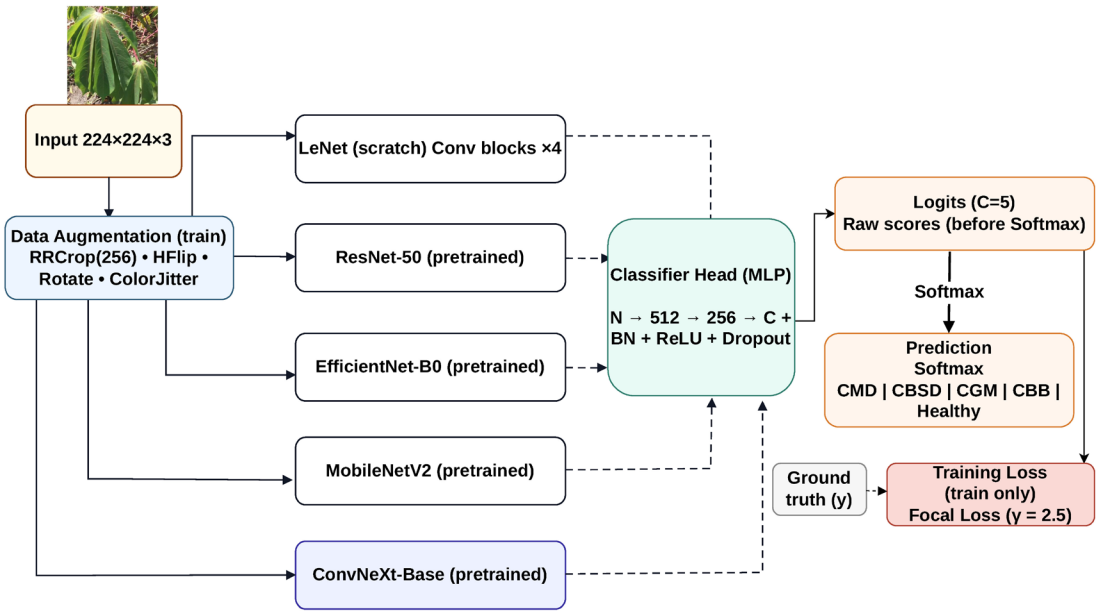


Fig. 1. Overview of the training pipeline.

The remainder of this paper is organized as follows: Section II reviews related work on class imbalance, loss functions, and cassava disease classification. Section III describes the methodology, including backbone architectures, the shared classifier head, and the four loss functions. Section IV details the experimental setup. Section V presents the results and discusses the key findings. Section VI concludes with a summary and directions for future work.

II. Related Work

A. Class Imbalance in Deep Learning

The imbalance ratio ($IR = N_{max}/N_{min}$) measures the severity of dataset skew. Buda *et al.* [6] found that CNN performance drops as IR increases, causing the decision boundaries to shift toward the majority classes. In these situations, models often become overconfident in predicting the majority class and fail to learn useful features for minority classes [12, 13]. Methods to handle imbalance fall into three main groups. Data-level methods modify the input distribution using over-sampling (Synthetic Minority Over-sampling Technique (SMOTE) [14], Adaptive Synthetic (ADASYN) [15]) or data augmentation [16]. These methods can improve balance but may increase training time. Algorithm-level methods modify the loss function. Examples include cost-sensitive cross-entropy, class-balanced loss [8], Label-Distribution-Aware Margin (LDAM) [9], focal loss [11], and recent variants such as Dynamic-Recall Focal Loss [17]. Cui *et al.* [8] formalized the concept of the effective number of samples and proposed CB weights defined as $wc = (1-\beta)/(1-\beta^{n_c})$, where n_c is the number of

training samples in class c and $\beta \in [0, 1)$ controls the degree of rebalancing. This approach provides a principled alternative to simple inverse-frequency weighting. Finally, decoupling methods separate feature learning from classifier tuning to reduce boundary distortion [18, 19]. A common gap in the current literature is that loss functions are rarely compared under strictly controlled conditions: studies often change the backbone, augmentation, or data split alongside the loss function, making it difficult to attribute performance differences solely to the loss. The present work addresses this gap by fixing all variables except the loss function.

B. Loss Functions

For a K-class classification problem, let $\mathbf{p}(x) = (p_1(x), \ldots, p_K(x))$ denote the predicted class probabilities. The standard cross-entropy loss for a sample is defined as

$$\mathcal{L}_{\mathrm{CE}}(x, y) = -\sum_{k=1}^{K} y_k \log p_k(x) \quad (1)$$

This reduces to $-\log p_y(x)$ when y is one-hot encoded. Under class imbalance, frequent classes contribute more heavily to the training objective, often yielding strong overall accuracy but weaker recall and F1 scores for under-represented classes. FL [11] reshapes the cross-entropy objective by multiplying it with a modulating factor that suppresses the contribution of well-classified examples:

$$\mathcal{L}_{\mathrm{FL}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \quad (2)$$

For a training sample (x, y), let pt denote the Softmax probability assigned to the ground-truth class y, where γ is the focusing parameter and αt is an optional class weight. In this study, we set $\gamma = 2.5$ and $\alpha_t = 1.0$ for all focal-loss experiments. From a theoretical perspective, focal loss is classification-calibrated, although its raw outputs are not strictly proper estimates of class-posterior probabilities. Therefore, in this study, focal loss is used as an optimization objective rather than as a calibration method [20].

W-CE assigns inverse-frequency weights to each class: $w_c = N/(K \cdot n_c)$, where N is the total number of samples and K is the number of classes. CB loss [8] replaces simple inverse-frequency weights with effective-number weights: $w_c = (1 - \beta)/(1 - \beta^{n_c})$, providing more principled rebalancing. In this study, we use $\beta = 0.9999$ following the original recommendation.

C. Disease Detection in Cassava

Table I summarizes representative results reported in the literature. A recent systematic review by Paçal *et al.* [21] highlights the growing importance of loss-function design in plant disease detection pipelines. Because the studies compared may differ in dataset version, split strategy, preprocessing, and evaluation protocol, these numbers are provided for context rather than for a strict head-to-head comparison. Zhong *et al.* [22] addressed cassava leaf disease classification on the 2021 Kaggle dataset using a Transformer-embedded ResNet (T-RNet) and an imbalance-aware FAMP-Softmax loss, reporting 91.1% weighted accuracy/F1. Other researchers used attention modules with EfficientNet [23] or model ensembles [24], achieving accuracies of 88.1%–89.9%. While ensembles achieve high accuracy, their computational cost makes them harder to deploy on edge devices. Instead of introducing a more complex architecture, the present study examines whether loss-function choice alone can improve class-balanced performance under a controlled protocol. This framing is intended as a controlled empirical comparison rather than a direct head-to-head replacement for prior methods. The surveyed literature reveals two gaps: (i) existing comparisons of loss functions for cassava classification typically vary multiple experimental factors simultaneously, and (ii) explicit class-level weighting methods (W-CE, CB) have not been compared with focal loss under controlled conditions on this benchmark. The following section describes our methodology for addressing these gaps. Results are not directly comparable because of differences in dataset versions, split strategies, preprocessing, and evaluation protocols. Our results report test-set evaluation with TTA. CE = cross-entropy; FL = focal loss; W-CE = weighted CE; CB = class-balanced loss. Method abbreviations: STL = standard transfer learning; T-R hyb. = Transformer–ResNet hybrid; IA Softmax = imbalance-aware Softmax loss; MS fusion = multi-scale feature fusion; CNN ens. = ensemble of CNNs; Best ens. = best ensemble variant; Ctrl. 4-loss = controlled four-loss study.

TABLE I. CONTEXTUAL COMPARISON WITH REPRESENTATIVE STUDIES ON CASSAVA DISEASE CLASSIFICATION

Study	Year	Architecture	Loss	Acc (%)	Params (M)	Method
[22]	2022	VGG16	CE	88.5	138	STL
		Inception-v3	CE	88.0	27	
		Xception	CE	88.1	23	
		ResNet-50	CE	88.9	25.6	
		DenseNet-121	CE	88.5	8.0	
		T-RNet	CE	90.6	14	T-R hyb.
		T-RNet	FAMP-Softmax	91.1	14	IA Softmax
[23]	2022	MSFM-EfficientNet	CE	88.1	~5	MS fusion
[24]	2025	EfficientNet-B0	CE	89.5	5.3	CNN ens.
		MobileNetV2	CE	89.5	3.5	
		ResNet50V2	CE	89.5	25.6	
		InceptionV3	CE	88.2	27	
		DenseNet169+EffNetB0	CE	89.9	~16	Best ens.
Ours	2026	ConvNeXt-Base	FL+TTA	91.67	88.2	Ctrl. 4-loss

III. Methodology

A. Problem Formulation

For an input image x, the backbone g_φ extracts a feature vector $h = g_\varphi(x)$. The classifier head c_ψ maps h to logits $\mathbf{z} = c_\psi(\mathbf{h}) \in \mathbb{R}^K$, where K is the number of classes. Class probabilities are then obtained as p = Softmax(z), and the predicted label is $\hat{y} = arg\ max_k p_k$. Training minimizes the regularized empirical risk:

$$\theta^* = \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(f_\theta(\mathbf{x}_i), y_i) + \lambda\|\theta\|_2^2 \qquad (3)$$

where $\theta = \{\varphi, \psi\}$ contains all learnable parameters, N is the number of training samples, and $\mathcal{L}$ denotes one of the four evaluated loss functions (CE, W-CE, CB, or FL). Model selection is based on validation Macro-F1 to reduce bias toward majority class accuracy during early stopping. The term $\lambda\|\theta\|_2^2$ applies L_2 weight decay to reduce overfitting.

B. Network Architectures and Classifier Head

Table II lists the five backbones used in this study. LeNet [25] serves as a basic baseline trained from scratch, using batch normalization [26] after each convolutional layer. We also use ImageNet pre-trained models: ResNet-50 [2], EfficientNet-B0 [27], MobileNetV2 [28], and ConvNeXt-Base [29].

To isolate the effect of the loss function, we replace the native classification layer of each backbone with a standard Multi-Layer Perceptron (MLP) head (Fig. 2). This head maps backbone features of dimension d through two hidden layers of sizes 512 and 256 before the final K-class classifier, with dropout rates of 0.4, 0.3, and 0.2 applied progressively.

TABLE II. Architectural Specifications. Feature Dim is the Output Size before the MLP Head

Backbone	Year	Params (M)	Feature Dim	Pretrain
LeNet (custom)	1998	2.8	4096	No
ResNet-50	2016	24.7	2048	Yes
EfficientNet-B0	2019	4.8	1280	Yes
MobileNetV2	2018	3.0	1280	Yes
ConvNeXt-Base	2022	88.2	1024	Yes

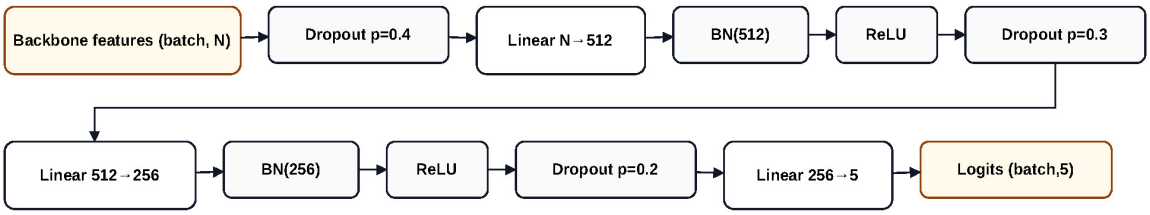


Fig. 2. Shared classification head used across all evaluated backbones.

C. Optimization Protocol

We use the AdamW optimizer [30] with a weight decay of 10^{-2}. AdamW is selected because decoupled weight decay has been shown to improve optimization stability and generalization in modern networks. Learning rates depend on model size: 10^{-3} for LeNet; 10^{-4} for ResNet-50, EfficientNet-B0, and MobileNetV2; and 5×10^{-5} for ConvNeXt-Base.

The schedule starts with a 3-epoch linear warmup to reduce early optimization instability, followed by cosine annealing decay. Label smoothing (0.1) is applied consistently across all runs to keep the optimization setting fixed across backbones and loss functions; its interaction with class imbalance has been studied separately by Müller *et al.* [31]. Early stopping halts training if validation Macro-F1 does not improve for 12 consecutive epochs.

With the methodology established, the next section describes the dataset, augmentation pipeline, and evaluation protocol used to validate the approach.

IV. Experimental Setup

A. Dataset and Imbalance

We evaluate the models on the iCassava 2019 dataset [5], which contains field images captured under unconstrained conditions. Table III and Fig. 3 illustrate the long-tailed class distribution. Cassava Mosaic Disease (CMD) is the majority class, yielding an imbalance ratio of IR = 8.41.

TABLE III. Training Set Composition Demonstrating Class Disparity

Class	Samples	Percentage	Ratio to Min	Category
CMD	2658	47.0%	8.41×	Majority
CBSD	1443	25.5%	4.57×	Moderate
CGM	773	13.7%	2.45×	Moderate
CBB	466	8.2%	1.47×	Minority
Healthy	316	5.6%	1.00×	Minority
Total	5,656	100.0%	—	—

Note: CMD = Cassava Mosaic Disease; CBSD = Cassava Brown Streak Disease; CGM = Cassava Green Mottle; CBB = Cassava Bacterial Blight.

Several disease categories exhibit visually similar symptoms (Fig. 4), which further complicates classification. CMD causes leaf mosaic patterns, CBSD causes vein chlorosis, CGM causes yellow mottling, and CBB causes angular necrotic lesions. Variable lighting and background clutter further complicate feature extraction.

We reserve 1,885 images for testing and use fixed training, validation, and test partitions across all experiments to ensure fair comparison.

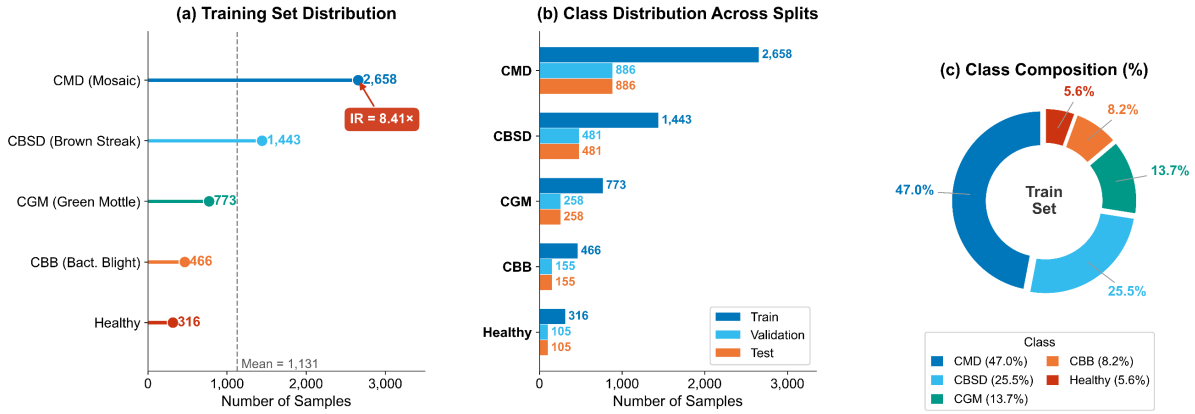


Fig. 3. Distribution analysis showing Cassava Mosaic Disease (CMD) dominance across the train, validation, and test partitions.

Fig. 4. Representative field samples from the iCassava 2019 dataset [5].

B. Data Augmentation

Images are resized to 256×256 pixels. We use a batch size of 32 (Table IV). To reduce overfitting, we apply RandomResizedCrop (scale 0.7–1.0), horizontal and vertical flips, ColorJitter (0.3), and Random Erasing (p = 0.25). Mixup regularization (α = 0.2) [32] is also used to smooth decision boundaries between classes.

TABLE IV. STANDARDIZED TRAINING HYPERPARAMETERS APPLIED TO ALL MODELS

Parameter	Value / Strategy
Image Resolution	256×256
Batch Size	32
Optimizer	AdamW (Weight decay 10^{-2})
Learning Rate	5×10^{-5} to 1×10^{-3}
LR Schedule	Linear Warmup (3 ep) → Cosine Decay
Max Epochs	100
Early Stopping	Patience: 12 epochs (Monitor: Val Macro-F1)
Label Smoothing	0.1 (Applied consistently across all runs)
FL Parameters	γ = 2.5, α = 1.0

C. Evaluation Metrics and Test-Time Augmentation (TTA) Setup

Experiments are conducted on an Intel Core i5-13400F CPU with an NVIDIA RTX 4070 Ti SUPER (16 GB VRAM) using PyTorch 2.6.0. The random seed is fixed at 42 for all runs. We measure performance using overall accuracy, precision, recall, and Macro-F1. Macro-F1 averages the F1 scores across all classes equally, ensuring that strong performance on the majority class does not hide weak performance on minority classes:

$$\text{Macro-F1} = \frac{1}{K}\sum_{k=1}^{K} F1_k, \quad F1_k = \frac{2P_kR_k}{P_k+R_k} \tag{4}$$

where P_k and R_k denote the precision and recall of class k, respectively.

To improve prediction robustness to leaf orientation, we apply horizontal-flip Test-Time Augmentation (TTA). For each test image, probabilities from the original and horizontally flipped versions are averaged as:

$$\bar{\mathbf{p}}(\boldsymbol{x}) = \frac{1}{2}\Big(\text{Softmax}(\mathbf{z}(\boldsymbol{x})) + \text{Softmax}\big(\mathbf{z}(\text{flip}_H(\boldsymbol{x}))\big)\Big) \tag{5}$$

where $\mathbf{z}(\cdot)$ is the output vector before Softmax, and flip_H is the horizontal flip. The final prediction is $\arg max_k\ \bar{p}_k(\mathbf{x})$.

Because all comparisons are reported on a single fixed split and a single random seed, very small differences between configurations, especially those below 0.5 Macro-F1 points, should be interpreted as empirical tendencies rather than as statistically conclusive evidence.

TABLE V. TEST METRICS EVALUATED WITHOUT TTA ACROSS FOUR LOSS FUNCTIONS

Architecture	CE		W-CE		CB		FL	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
LeNet (scratch)	79.95	69.91	71.88	64.39	73.85	65.79	82.65	74.35
ResNet-50	88.59	83.09	87.37	83.14	88.01	82.95	88.44	83.22
EfficientNet-B0	88.01	82.76	87.80	82.75	87.69	82.83	89.39	85.12
MobileNetV2	89.18	83.97	87.37	82.37	86.79	81.63	88.70	83.76
ConvNeXt-Base	90.72	87.07	91.35	88.48	91.72	89.03	91.25	88.86

TABLE VI. PERFORMANCE DELTAS (Δ) INDUCED BY HORIZONTAL-FLIP TTA FOR EACH LOSS FUNCTION

Architecture	CE		W-CE		CB		FL	
	Acc Δ	F1 Δ	Acc Δ	F1 Δ	Acc Δ	F1 Δ	Acc Δ	F1 Δ
LeNet (scratch)	+1.27	+2.26	+0.43	+0.48	−0.16	−0.09	+0.21	+0.27
ResNet-50	+0.53	+0.54	+0.38	+0.53	+0.58	+0.44	+1.00	+1.55
EfficientNet-B0	+1.75	+3.22	+0.79	+1.21	+0.80	+1.43	+0.48	+0.71
MobileNetV2	−0.21	+0.71	+0.11	+0.48	0.00	−0.11	−0.11	−0.14
ConvNeXt-Base	+0.53	+1.21	+0.06	−0.10	−0.10	+0.34	+0.42	+0.53

Having detailed the experimental protocol, we now present the results. Table V reports baseline test results without TTA for all four loss functions. Even without TTA, replacing CE with FL improves the mean accuracy by +0.80% and Macro-F1 by +1.70%; it is the only method that improves performance on average across all five backbones. In contrast, W-CE and CB reduce the mean Macro-F1 by −1.13% and −0.91%, respectively, mainly because of marked degradation on LeNet (−5.52 and −4.12 F1 points). Table VI then reports the gains obtained by adding TTA during inference.

V. RESULTS AND DISCUSSION

A. Overall Performance under TTA

Table VII presents the final test results with TTA for all four loss functions. ConvNeXt-Base with focal loss attains the highest Macro-F1 among the evaluated configurations, 89.39%, while CB loss on ConvNeXt-Base reaches the highest accuracy (91.94%) and a nearly identical Macro-F1 (89.37%). The 0.02-point Macro-F1 gap between FL and CB on ConvNeXt-Base is practically negligible; accordingly, the apparent advantage of FL in this setting should be interpreted as an observed result of the current experiment rather than as a definitive ranking. Across all five backbones, focal loss is the only method that improves the average Macro-F1 relative to CE (+0.70 points), whereas W-CE and CB reduce the average by −2.20 and −2.10 points, respectively, mainly because of drops on LeNet and MobileNetV2. Table VIII breaks down the minority-class gains.

TABLE VII. COMPREHENSIVE EVALUATION ON THE TEST SET WITH HORIZONTAL-FLIP TTA. BEST VALUES PER ARCHITECTURE ARE **BOLD**

Architecture	CE		W-CE		CB		FL	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
LeNet (scratch)	81.22	72.17	72.31	64.87	73.69	65.70	**82.86**	**74.62**
ResNet-50	89.12	83.63	87.75	83.67	88.59	83.39	**89.44**	**84.77**
EfficientNet-B0	**89.76**	**85.98**	88.59	83.96	88.49	84.26	89.87	85.83
MobileNetV2	**88.97**	**84.68**	87.48	82.85	86.79	81.52	88.59	83.62
ConvNeXt-Base	91.25	88.28	91.41	88.38	**91.94**	89.37	91.67	**89.39**

TABLE VIII. MINORITY-CLASS F1 (CBB, HEALTHY) AND MACRO-F1 UNDER FOUR LOSS FUNCTIONS WITH TTA

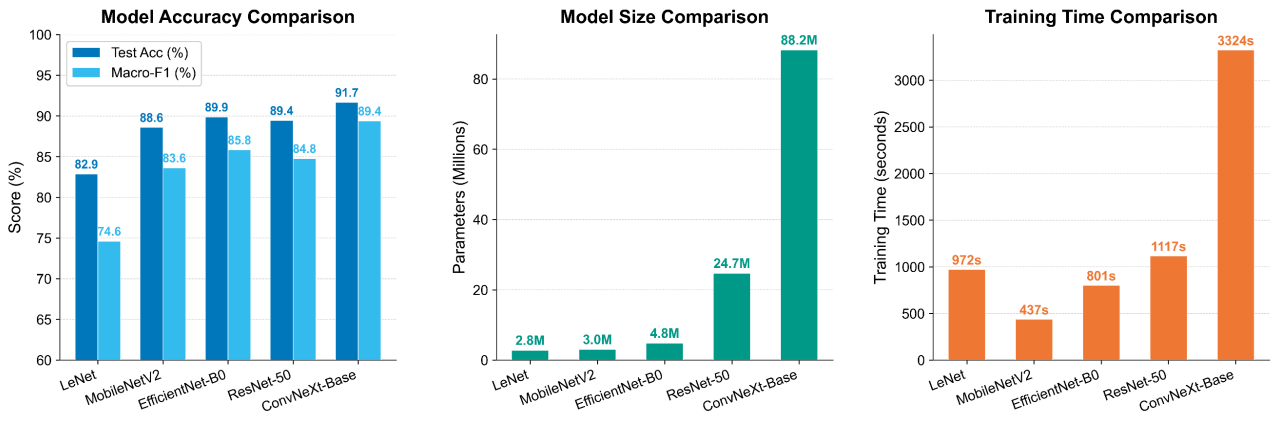
Architecture	CE		W-CE		CB		FL	
	Acc Δ	F1 Δ	Acc Δ	F1 Δ	Acc Δ	F1 Δ	Acc Δ	F1 Δ
LeNet (scratch)	+1.27	+2.26	+0.43	+0.48	−0.16	−0.09	+0.21	+0.27
ResNet-50	+0.53	+0.54	+0.38	+0.53	+0.58	+0.44	+1.00	+1.55
EfficientNet-B0	+1.75	+3.22	+0.79	+1.21	+0.80	+1.43	+0.48	+0.71
MobileNetV2	−0.21	+0.71	+0.11	+0.48	0.00	−0.11	−0.11	−0.14
ConvNeXt-Base	+0.53	+1.21	+0.06	−0.10	+0.22	+0.34	+0.42	+0.53

B. Accuracy vs. Efficiency Trade-Off

Table IX and Fig. 5 compare computational cost and accuracy for the FL+TTA setting. Although ConvNeXt-Base delivers the highest accuracy, EfficientNet-B0 provides a more attractive parameter-to-performance trade-off within the evaluated models. The reported runtimes correspond to training rather than inference; therefore, deployment implications should be interpreted cautiously.

TABLE IX. TRADE-OFF ANALYSIS FOR THE FL+TTA SETTING, RELATING ACCURACY, PARAMETER COMPLEXITY, AND TRAINING LATENCY

Architecture	Params (M)	Test Acc (%)	Macro F1 (%)	Time (s)
LeNet (scratch)	2.8	82.90	74.60	9722
MobileNetV2	3.0	88.60	83.60	437
EfficientNet-B0	4.8	89.90	85.80	8001
ResNet-50	24.7	89.40	84.80	11117
ConvNeXt-Base	88.2	91.67	89.39	3324



(a) (b) (c)

Fig. 5. Comparison of (a) test accuracy vs. Macro-F1, (b) parameter count and (c) training time across architectures in the FL+TTA setting.

Interestingly, LeNet requires longer training despite having fewer parameters, likely because it is trained from scratch and uses less compute-efficient convolutional blocks than MobileNetV2. Faster convergence of the pre-trained MobileNetV2 may also contribute to this difference under early stopping.

C. Optimization Dynamics

Fig. 6 shows convergence behavior across all four loss functions. Under CE (Fig. 6(a)), all pre-trained backbones converge within 25–30 epochs, with ConvNeXt-Base plateauing near 91%. Under FL (Fig. 6(d)), the absolute loss scale is lower (~0.15 to 0.9) because of the $(1 - p_t)\,\gamma$ modulating factor, and LeNet converges more slowly but to a higher final accuracy than under CE.

The W-CE (Fig. 6(b)) and CB (Fig. 6(c)) panels reveal a clear pattern: while ConvNeXt-Base still converges smoothly to ~91%, LeNet exhibits severe oscillation and fails to exceed 70% validation accuracy, a pattern consistent with the interpretation that large class weights can make optimization less stable in a low-capacity feature space. Because weighted losses and focal loss operate on different numerical scales, the absolute loss values should not be compared directly across panels.

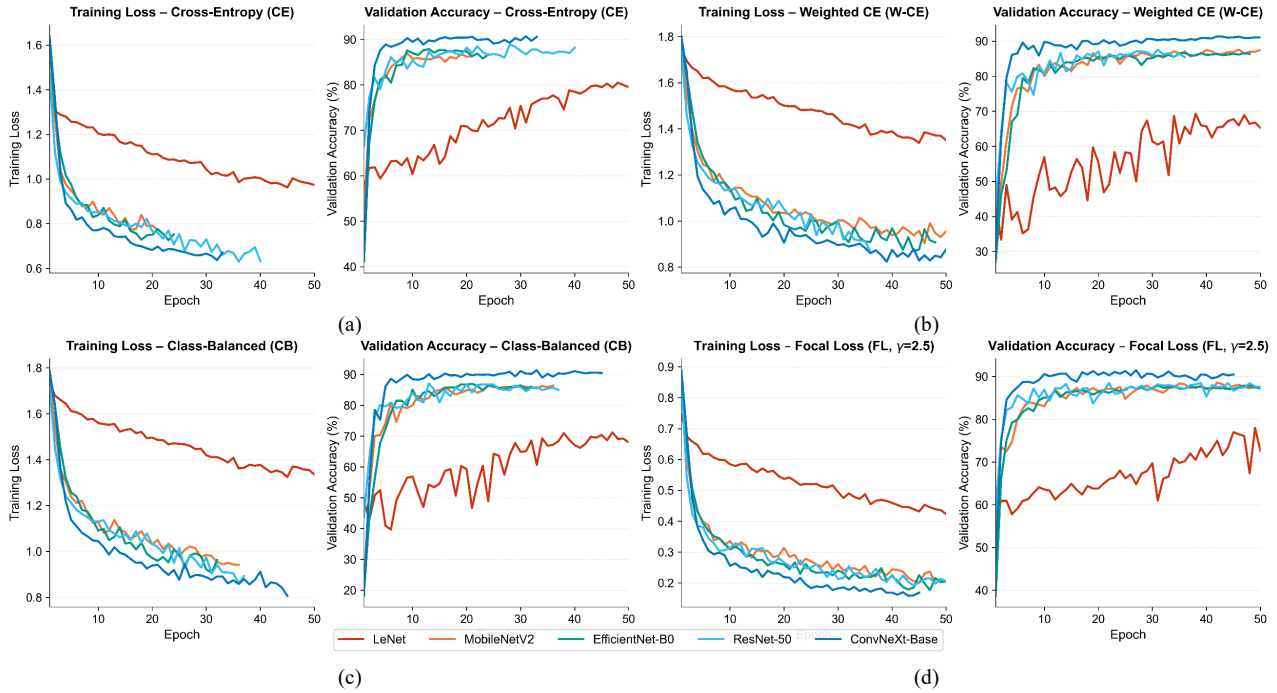


Fig. 6. Training dynamics across four loss functions. Each panel shows training loss (left) and validation accuracy (right) for all five backbones. (a) CE; (b) W-CE; (c) CB; (d) FL.

D. Class-Level Analysis

Table X reports class-specific F1 for ConvNeXt-Base under all four loss functions. Both focal loss and CB improve minority-class F1: CBB improves by +2.57% (FL) and +1.35% (CB), while Healthy improves by +2.38% (FL) and +2.03% (CB). Meanwhile, performance on the CMD class remains stable (≤ +0.24%). The main difference between focal loss and class-balanced loss lies in their mechanism of action: focal loss operates at the sample level, down-weighting easy samples regardless of class membership, whereas CB applies fixed class-level weights. On ConvNeXt-Base, where the feature space is relatively rich, both approaches converge to similar performance (~89.4% Macro-F1).

The confusion matrices in Fig. 7 visually support these patterns across all four loss functions. FL and CB show fewer off-diagonal entries for minority classes (CBB, Healthy), whereas W-CE disperses more predictions away from the diagonal, consistent with its lower Macro-F1. For the two minority classes, FL and CB also achieve the highest diagonal counts (CBB: 118/117; Healthy: 98/100). Fig. 8 provides an inference example for the under-represented Healthy class. In this example, ConvNeXt-Base with focal loss predicts the Healthy class with 79% confidence.

TABLE X. CLASS-WISE F1 (%) BREAKDOWN FOR CONVNEXT-BASE WITH TTA UNDER FOUR LOSS FUNCTIONS

Class	Samples	Category	CE	W-CE	CB	FL
CBB	155	Minority	77.70	77.03	79.05	80.27
CBSD	481	Moderate	90.79	89.96	91.46	90.36
CGM	258	Moderate	85.38	87.47	86.53	86.36
CMD	886	Majority	95.23	95.48	95.47	95.27
Healthy	105	Minority	92.31	91.94	94.34	94.69
Macro-F1	1885	-	88.28	88.38	89.37	89.39

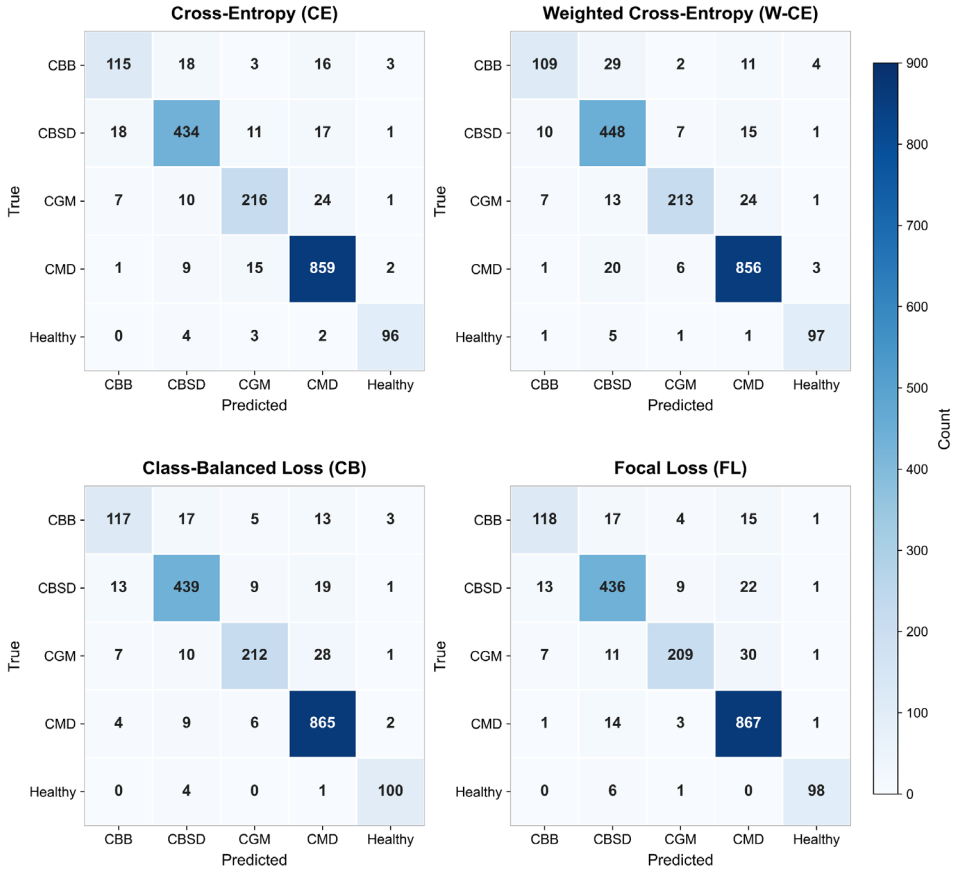


Fig. 7. Confusion matrices for ConvNeXt-Base under four loss functions with horizontal-flip TTA on the test set (N = 1,885).

Fig. 8. Example prediction for a Healthy-class leaf using ConvNeXt-base with focal loss.

E. *Qualitative Error Analysis*

Residual errors are mainly associated with three factors. First, several disease categories share overlapping visual symptoms, especially in early or mild stages, which increases confusion among CBB, CBSD, and CGM. Second, low-contrast lesions and uneven illumination can obscure local disease patterns that are important for reliable recognition. Third, background clutter, leaf overlap, and partial occlusion reduce the visibility of discriminative regions. These factors are consistent with the remaining confusion patterns observed in Fig. 7. Overall, the results suggest that loss reweighting can improve minority sensitivity, but challenging image conditions remain an important source of residual error.

F. *Interaction Between Backbone Capacity and Loss Function*

The results suggest a clear interaction between backbone capacity and loss-function effectiveness. Three regimes emerge:

(a) High-capacity backbone (ConvNeXt-Base): All reweighting methods improve or match CE, with FL and CB yielding comparable Macro-F1 (~89.4%).

(b) Medium-capacity backbone (ResNet-50): FL provides clear gains (+1.14% F1), while W-CE and CB provide marginal improvements.

(c) Low-capacity backbone (LeNet, MobileNetV2): Only FL maintains or improves performance; W-CE and CB cause significant degradation (up to −7.3% F1 on LeNet).

This pattern suggests that explicit class weights require a sufficiently expressive feature space to be effective. When the backbone cannot produce discriminative representations, large class weights may amplify noisy gradients and make training less stable. Focal loss appears to mitigate part of this issue because its modulation factor $(1 - p_t)^{\gamma}$ suppresses the contribution of easy samples and places greater emphasis on hard examples. However,

because the paper does not report multi-seed variance, loss-specific hyperparameter tuning, or direct gradient statistics, this mechanism should be regarded as a plausible interpretation rather than a definitive causal explanation.

From a deployment perspective, EfficientNet-B0 offers the strongest accuracy-efficiency trade-off among the evaluated backbones, whereas ConvNeXt-Base with FL is the more suitable choice when predictive performance is prioritized.

VI. Conclusion

This study compared four loss functions: CE, W-CE, CB, and FL across five CNN backbones for imbalanced cassava leaf disease classification (IR = 8.41). Under a controlled protocol with shared classifier heads and matched hyperparameters, the key findings are:

- Focal loss attains the highest observed Macro-F1, 89.39%, together with 91.67% accuracy for ConvNeXt-Base with TTA, while improving minority-class F1 for CBB (+2.57%) and Healthy (+2.38%) without degrading CMD performance.
- Class-balanced loss achieves near-equivalent performance on ConvNeXt-Base (89.37% Macro-F1), indicating that the gap between FL and CB on this backbone is very small in the present experiment.
- Explicit class-level weighting (W-CE, CB) degrades performance on low-capacity architectures (LeNet: −6.5 to −7.3% Macro-F1), suggesting that fixed class-level reweighting may be more sensitive to limited representational capacity.
- The effectiveness of loss reweighting depends on backbone capacity: in this setup, sample-level modulation (focal loss) shows a more stable improvement trend than class-level weighting across heterogeneous architectures.
- EfficientNet-B0 offers the strongest accuracy–efficiency trade-off among the evaluated backbones, making it attractive for resource-constrained agricultural deployments, while ConvNeXt-Base with FL provides the highest absolute predictive performance.

These results suggest that practitioners should consider backbone capacity jointly with loss-function choice rather than applying reweighting methods indiscriminately. At the same time, these recommendations should be validated through multi-seed experiments and additional datasets before being generalized more broadly.

This study has several limitations. Experiments are conducted on a single dataset (iCassava 2019) and a single random seed (42), so the paper does not provide confidence intervals or significance tests for small observed differences. Loss hyperparameters are not tuned symmetrically for each backbone and each loss function: focal loss is fixed at $\gamma = 2.5$ and $\alpha = 1.0$, while W-CE and CB are likewise not tuned separately per model. The interaction among focal loss, label smoothing, and mixup is not isolated in a dedicated ablation. The manuscript also does not analyze potential leakage across partitions at the plant, leaf, or near-duplicate image level, and it does not release the exact split indices or training scripts needed for full reproducibility. Other imbalance remedies, such as LDAM, Balanced Softmax, and resampling-based methods, were not evaluated in the present comparison. Deployment metrics such as inference latency and memory footprint are not measured directly. Extending the evaluation to multiple random seeds and to other imbalanced agricultural datasets, such as PlantVillage and Rice Disease, would help test the robustness of the conclusions and improve generalizability.

Data Availability

The iCassava 2019 dataset used in this study is publicly available through the Kaggle repository reported in [5]. The main training configuration and hyperparameters are described in Section IV. Additional implementation details are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no competing interests.

Author Contributions

Conceptualization, methodology, experimentation, and original draft preparation: Thanh-Hai Tong-Le. Review, editing and supervision: Thanh-Nghi Doan. Both authors have read and approved the final manuscript.

Acknowledgment

We acknowledge the time and facilities provided by Tien Giang University, Tra Vinh University and An Giang University for this study.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.,* vol. 25, pp. 1097–1105, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

[3] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers in Plant Science*, vol. 8, 1852, 2017.

[4] Food and Agriculture Organization (FAO), *Save and Grow: Cassava*, Rome, Italy: FAO, 2013.

[5] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, and J. Tusubira, "iCassava 2019 Fine-Grained Visual Categorization Challenge," arXiv preprint arXiv:1908.02900, 2019.

[6] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw*., vol. 106, pp. 249–259, 2018.

[7] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, "The class imbalance problem in deep learning," *Machine Learning*, vol. 113, 2022.

[8] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," *in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 9260–9269.

[9] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *Adv. Neural Inf. Process. Syst*., vol. 32, 2019.

[10] T.-H. Tong-Le and N.-G. Pham, "The problem of data imbalances and some methods of processing imbalanced data in deep learning models," *CTU J. Sci*., vol. 60, no. 5A, pp. 50–58, 2024.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[12] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, 27, 2019.
[13] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A survey on imbalanced learning," *Artif. Intell. Rev*., vol. 57, 2024.
[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority oversampling technique," *J. Artif. Intell. Res*., vol. 16, pp. 321–357, 2002.
[15] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Hong Kong, China, 2008, pp. 1322–1328.
[16] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, 60, 2019.
[17] X. Liu, L. Wang, L. Ma, and C. Wang, "DRFL: Dynamic-Recall Focal Loss for image classification and segmentation," *Appl. Artif. Intell.*, vol. 38, no. 1, 2024. doi: 10.1080/08839514.2024.2411845.
[18] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, 2020.
[19] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun, "Distribution alignment: A unified framework for long-tail visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, 2021, pp. 2361–2370.
[20] N. Charoenphakdee, J. Vongkulbhisal, N. Chairatanakul, and M. Sugiyama, "On focal loss for class-posterior probability estimation: A theoretical perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, 2021, pp. 5202–5211.
[21] I. Paçal, I. Kunduracıoğlu, M. H. Alma, M. Deveci, S. Kadry, J. Nedoma, and R. Martinek, "A systematic review of deep learning techniques for plant diseases," *Artif. Intell. Rev.*, vol. 57, no. 11, 304, 2024. doi: 10.1007/s10462-024-10944-7
[22] Y. Zhong, B. Huang, and C. Tang, "Classification of cassava leaf disease based on a non-balanced dataset using transformer-embedded ResNet," *Agriculture*, vol. 12, no. 9, 1360, 2022. doi: 10.3390/agriculture12091360
[23] M. Liu, H. Liang, and M. Hou, "Research on cassava disease classification using the multi-scale fusion model," *Front. Plant Sci.*, vol. 13, 1088531, 2022.
[24] G. Sambasivam *et al.*, "A hybrid deep learning model approach for automated detection and classification of cassava leaf diseases," *Sci. Rep*., vol. 15, 7009, 2025.
[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML),* Lille, France, 2015, pp. 448–456.
[27] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114.
[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.
[29] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 11966–11976.
[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019.
[31] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 4696–4705, 2019.
[32] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vancouver BC, Canada, 2018.