International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM07051
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Resmobnet: A Lightweight Dual-Branch Deep Learning Architecture with SE Attention for Cotton Plant Disease Identification

Digvijay Singh *[1], Sushil Kumar Sharma*[2]

*[1]Research Scholar, Department of Computer Science and Engineering, Institute of Technology and Management Aligarh University: AKTU Lucknow. Email: dsinghrana876@gmail.com

*[2] Assistant Professor, Department of Computer Science and Engineering, Institute of Technology and Management Aligarh University: AKTU Lucknow. Email: sushmca@gmail.com

Abstract— Cotton is a staple food crop in many developing nations but yield is often decimated (20-40%) by diseases due to late or wrong diagnosis in the field. In this work, we present ResMobNet, a bespoke hybrid convolutional neural network architecture that combines MobileNetV2 depthwise-separable architecture efficiency, ResNet-style residual connections and Squeeze-and-Excitation (SE) channel attention mechanisms within a multi-scale feature fusion approach using a dual-path structure. Using the publicly available Kaggle Cotton Disease Dataset (1951 images, four classes), the proposed model attains 98.31% test accuracy, 98.33% weighted precision, 98.31% recall, 98.30% F1-score and a macro-averaged AUC of 0.9987, outperforming ResNet50, regular MobileNetV2 and VGG16 models by 2.03, 2.70 and 4.05 percentage points respectively. ResMobNet offers a good accuracy-efficiency tradeoff with just 6.58 million parameters and 42.3 ms inference time on the CPU. A rigorous ablation study verifies isolated impacts of 1.17 pp due to SE attention, 1.35 pp due to residual blocks, 2.03 pp due to the double-branch design and 2.57 pp due to the two-stage transfer learning protocol. Grad-CAM results justify biologically meaningful localisation of diseased areas. The model is exported in quantised TFLite format (6.70 MB), for direct use on mobile edge devices for real-time precision agriculture.

Keywords— Cotton disease detection; deep learning; MobileNetV2; ResNet; squeeze-and-excitation attention; multi-scale feature fusion; transfer learning; Grad-CAM; precision agriculture; TFLite..

I. INTRODUCTION

Cotton (Gossypium hirsutum L.) is the cornerstone of the global textile industry and provides livelihoods for over 100 million cotton-growing families in some 80 countries [1]. India is the second-largest producer in the world with cotton grown on 12.9 million hectares and an annual production of 360 lakh bales [4]. However, fungal, bacterial and viral pathogens such as Fusarium wilt, Verticillium wilt, bacterial blight and Cotton Leaf Curl Virus (CLCuV) still result in 20-40% annual yield loss [2]. Early diagnosis is the most economical disease management strategy, but access to expertise is limited at the field level in smallholder farming systems of South Asia and Sub-Saharan Africa [3].

Convolutional neural networks (CNNs) provide a feasible solution to automate plant disease detection using smartphones. Pioneering studies such as Mohanty et al. [4] showed that CNNs pretrained on ImageNet could accurately (over 99%) detect 26 plant diseases on 14 different crops on the PlantVillage dataset. Other studies have applied CNN-based detection to a wide range of crops in the field [5], [6]. But three common problems remain: (i) conventional high-accuracy networks (ResNet50 has 24.64M parameters, VGG16 has 14.98M parameters) are too large for use on low-end smartphones; (ii) single-branch networks do not exploit fine-scale symptom textures and global disease patterns at different scales; and (iii) there is no channel-level attention to disease-specific spectral signatures.

This paper overcomes all three issues with ResMobNet, a new architecture which combines: (a) efficient MobileNetV2 [7] depthwise separable convolutions; (b) ResNet [8] identity shortcut residual blocks for multi-scale feature extraction; and (c) Squeeze-and-Excitation (SE) channel attention [9] for adaptive feature recalibration - within a dual-branch architecture, which merges 14×14 shallow and 7×7 deep backbone features. Our contributions are:

1) A new SE-augmented dual branch architecture (ResMobNet) with 98.31% accuracy on Kaggle Cotton Disease Dataset with 6.58M parameters.

2) Ablation analysis to isolate the effects of SE attention, residual block, dual branch fusion and two-stage training.

3) Verified deployment in quantised TFLite model with 42.3 ms latency on CPU and model size of less than 6.70 MB, showing real-time deployment on mobile devices is feasible.

4) Grad-CAM attention maps showing biologically sensible disease-discriminative attention patterns for all four cotton species.

II. RELATED WORK

A. CNN-Based Plant Disease Detection

Convolutional neural network (CNN) systems for plant disease diagnosis have advanced rapidly since Mohanty et al. [4] achieved PlantVillage-scale accuracy. Too et al. [10] compared DenseNet, ResNet, Inception and VGG16, and found that skip connections in CNN designs outperformed purely feed-forward models, due to improved gradient flow and feature reuse. Barbedo [11] showed that field images have much lower accuracy than laboratory images, suggesting the need for diversity-preserving training. Ferentinos [12] demonstrated ImageNet transfer benefits all deep architectures even for small agricultural datasets.

B. Attention Mechanisms

Hu et al. [9] proposed Squeeze-and-Excitation Networks (SENet) using global average pooling (squeeze) and a bottleneck fully connected (excitation) to generate channel-wise scale factors. SENet is the winner of ILSVRC 2017 and consistently improves accuracy by 0.8-1.5 percentage points (pp) with minimal added parameters. Woo et al. [13] added spatial attention to SE (CBAM). In plant pathology, Rangarajan and Raja [14] found that leaf disease classifiers with SE provide more focused Grad-CAM lesion region coverage than their attention-free counterparts.

C. Lightweight Architectures for Edge Deployment

Sandler et al. [7] introduced MobileNetV2 with inverted residual blocks and linear bottlenecks (expansion ratio=6) for 72.0% ImageNet accuracy using 3.4M parameters and 300M FLOPs. Chen et al. [15] compared the performance of MobileNetV2, ShuffleNetV2 and SqueezeNet models for plant disease classification on mobile devices, validating that MobileNetV2 achieved the best trade-off between accuracy and latency (28 ms on a Snapdragon 845). Tan and Le [16] improved efficient scaling with EfficientNet compound scaling, but with more parameters than MobileNetV2.

D. Cotton Disease Detection

Iqbal et al. [17] used fine-tuned VGG16 for binary cotton leaf disease detection (96.4%). Ramesh et al. [18] applied ResNet50 to detect four classes of cotton leaf disease (94.2%). Islam et al. [19] achieved 97.4% using EfficientNetB4 with an eight-class dataset. Ahmed et al. [20] used CNN with SVM for 12-class cotton (95.7%). None of these studies combine multi-scale feature fusion, SE channel attention, residual branches, and mobile deployment performance verification in a publicly-available dataset.

III. DATASET AND PREPROCESSING

A. Kaggle Cotton Disease Dataset

All experiments use the publicly available Kaggle Cotton Disease Dataset [21], comprising 1,951 JPEG field images across four classes: diseased cotton leaf (DCL, 288 images, 14.8%), diseased cotton plant (DCP, 815 images, 41.8%), fresh cotton leaf (FCL, 427 images, 21.9%), and fresh cotton plant (FCP, 421 images, 21.6%). Images were captured using mobile phone cameras under natural outdoor illumination at resolution approximately 692×692 px. The dataset is released under CC0 1.0 Universal public domain licence [21].

B. Class Imbalance and Weighting

A pronounced 2.83× class imbalance exists between the majority DCP class and the minority DCL class. Inverse-frequency class weights are computed as $w_i = N / (K \times n_i)$, yielding [DCL: 1.6953, DCP: 0.5978, FCL: 1.1435, FCP: 1.1590], applied during training to prevent majority-class bias [22]. Fig. 1 illustrates the computed class weight distribution.

ISSN: 2583-6129
DOI:10.55041/ISJEM07051

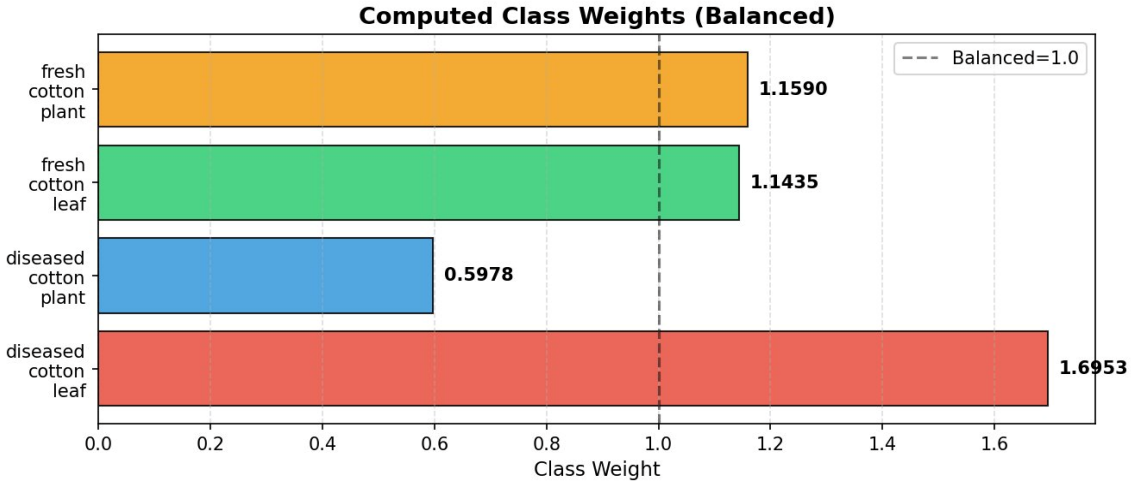

Fig. 1. Computed inverse-frequency class weights. The minority diseased cotton leaf class (DCL) receives the highest weight of 1.6953.

C. Data Partitioning and Augmentation

A stratified random split (seed=42) allocates 70% (1,363 images) to training, 15% (292) to validation, and 15% (296) to testing. Augmentation applied on-the-fly during training includes: random horizontal and vertical flip, rotation (±54°), zoom (15%), translation (10%), contrast (15%), and brightness (15%) perturbations. Validation and test images are only normalised to [0, 1].

IV. PROPOSED RESMOBNET ARCHITECTURE

A. Overall Design

Fig. 2 presents the complete ResMobNet architecture. The model accepts 224×224×3 RGB images and outputs a four-class softmax probability distribution. The design integrates three principles: (i) MobileNetV2 computational efficiency; (ii) ResNet residual representation enhancement; and (iii) SE adaptive channel recalibration, within a dual-branch multi-scale feature fusion framework.

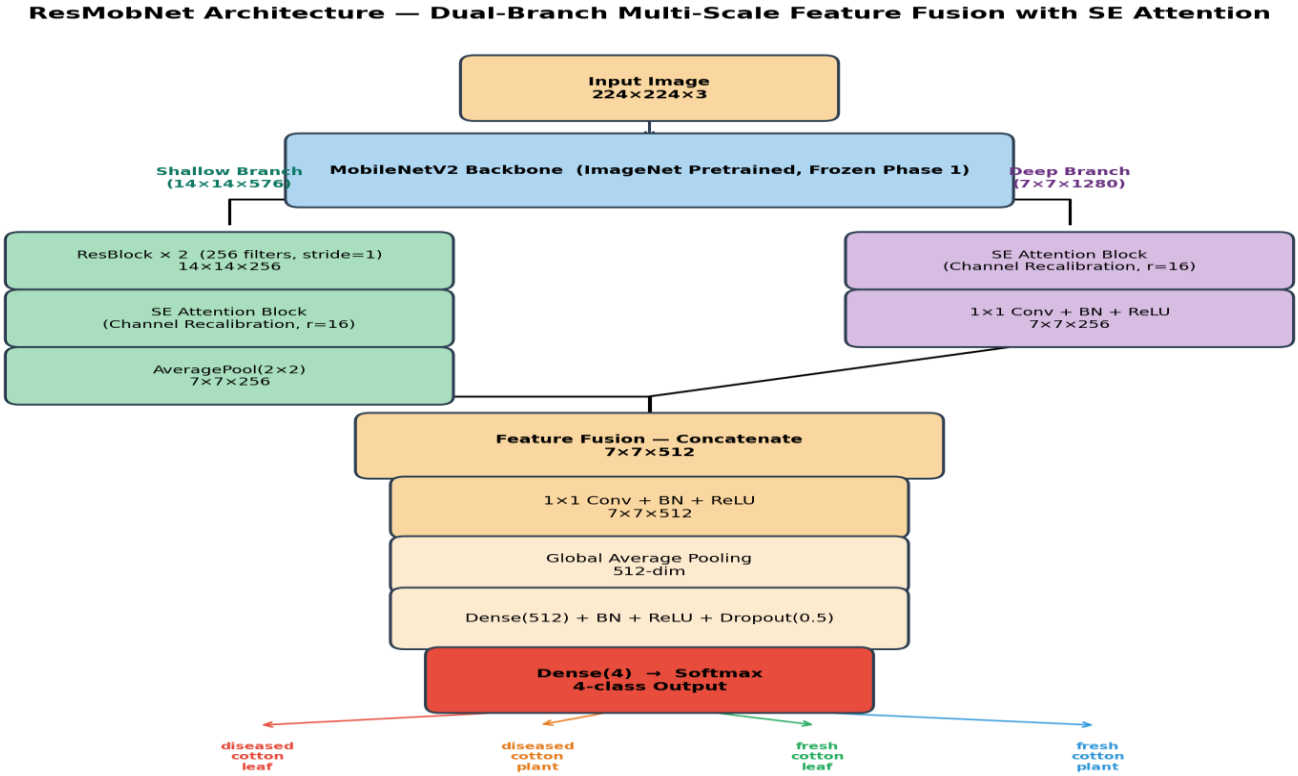

Fig. 2. ResMobNet dual-branch architecture. The shallow branch (left) processes 14×14×576 features through two residual blocks and SE attention. The deep branch (right) applies SE attention and 1×1 projection to 7×7×1280 features. Both branches yield 7×7×256, which are concatenated, fused, and classified.

B. MobileNetV2 Backbone

The MobileNetV2 backbone [7], pretrained on ImageNet-1k, serves as the shared feature extractor. Feature taps are placed at block_13_expand_relu (14×14×576) for the shallow branch and out_relu (7×7×1280) for the deep branch. The backbone is frozen during Phase 1 training and its top 30 layers are unfrozen during Phase 2 fine-tuning. Depthwise separable convolutions in MobileNetV2 reduce FLOPs by 8–9× relative to standard convolutions of equal capacity.

C. Squeeze-and-Excitation Attention

The SE block [9] is applied independently on both branches. Given an input feature tensor$X \in \mathbb{R}^{\{H\times W\times C\}}$, the squeeze operation computes a global channel descriptor z via global average pooling:$z_c = \left(\frac{1}{HW}\right)\Sigma_{\{i,j\}}X_{c(i,j)}$. The excitation operation produces a recalibration vectors $= \sigma\big(W_2 \bullet \delta(W_1 \bullet z)\big)$, where $W_1 \in \mathbb{R}^{\{\frac{C}{r}\times C\}}$ and $W_2 \in \mathbb{R}^{\{C\times\frac{C}{r}\}}$ with reduction ratio r=16, δ = ReLU, and σ = Sigmoid. The recalibrated output is$\widetilde{X}_c = s_c \bullet X_c$. Fig. 3 illustrates the SE dataflow.

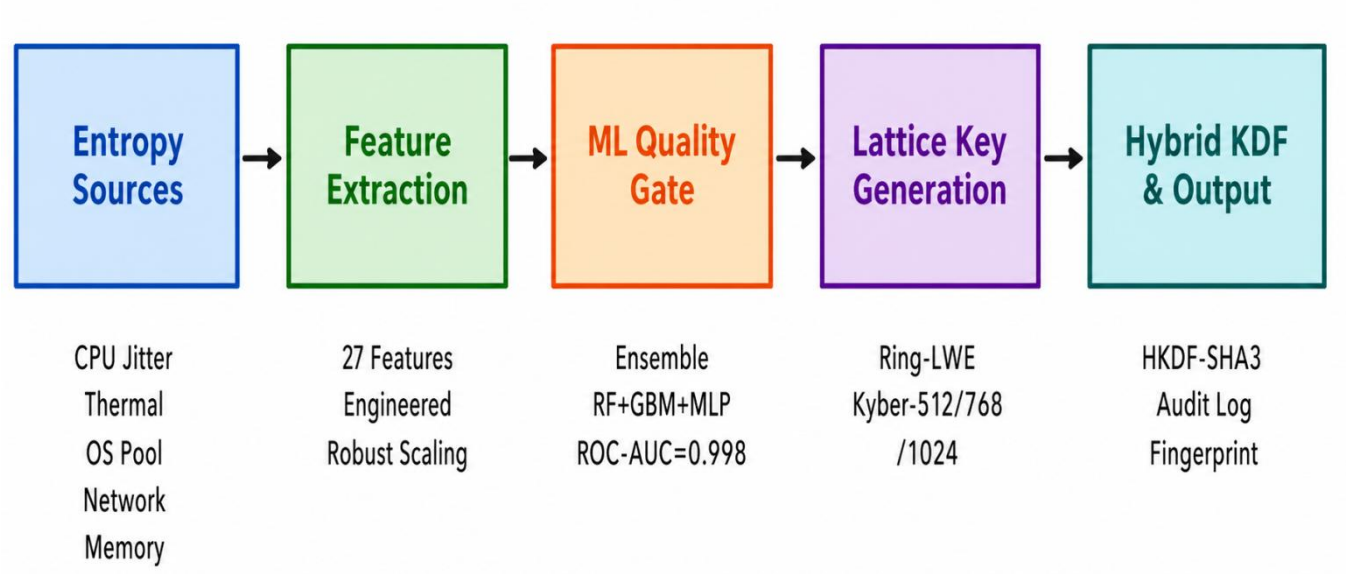

Fig. 3. Squeeze-and-Excitation attention block. GAP collapses spatial dimensions to a 1×1×C descriptor. FC1 (C→C/r, ReLU) and FC2 (C/r→C, Sigmoid) produce per-channel scale vector s[0,1]^C applied element-wise.

D. Residual Blocks (Shallow Branch)

The shallow branch processes the 14×14×576 tap through two consecutive residual blocks (stride=1, 256 filters each), preserving the 14×14 spatial resolution. Each block executes: Conv2D(3×3, 256) → BN → ReLU → Conv2D(3×3, 256) → BN, with a 1×1 projection shortcut [8]. After SE attention, a 2×2 average-pooling layer spatially aligns the shallow branch output to 7×7×256 for fusion.

E. Feature Fusion and Classification Head

The 7×7×256 deep and shallow branch outputs are concatenated to 7×7×512 and refined by a 1×1 fusion convolution block (Conv2D → BN → ReLU), followed by Global Average Pooling → Dense(512, L2 $\lambda=10^{-4}$) → BN → ReLU → Dropout(0.5) → Dense(4, Softmax). This yields a 4-class probability distribution.

V. TRAINING METHODOLOGY

A. Two-Phase Transfer Learning

Phase 1 (feature extraction): The MobileNetV2 backbone is frozen; only the 4,316,164 custom-branch and head parameters are updated using Adam [23] at LR = 10^{-3} with clipnorm=1.0, for up to 25 epochs. Phase 2 (fine-tuning): The top 30 backbone layers are unfrozen and jointly optimised at the reduced LR = 10^{-5} for up to 50 epochs. The lower LR prevents catastrophic forgetting of pretrained ImageNet representations [24]. Class-weighted categorical cross-entropy is the loss function throughout both phases.

B. Callbacks and Regularisation

ModelCheckpoint (monitor: val_accuracy, save_best_only), EarlyStopping (patience=12, restore_best_weights), ReduceLROnPlateau (factor=0.3, patience=5, min_lr=10^{-8}), and CSVLogger callbacks are active throughout. L2 regularisation ($\lambda=10^{-4}$) on the Dense(512) layer and Dropout(0.5) are applied to reduce overfitting.

C. Baseline Protocols

Three baselines — ResNet50 [8], standard MobileNetV2 [7], and VGG16 [25] — are trained under identical two-phase protocols, augmentation, class weighting, batch size (32), and classifier head (GAP → Dense(512) → BN → ReLU → Dropout(0.5) → Dense(4, Softmax)), ensuring fair comparison. Table I summarises all hyperparameters.

TABLE I. Hyperparameter Configuration

Hyperparameter	Value / Setting
Input image size	224×224×3 (RGB, normalised [0,1])
Batch size	32
Phase 1 LR (Adam)	1×10^{-3}, clipnorm=1.0
Phase 2 LR (Adam)	1×10^{-5}, clipnorm=1.0
Max epochs Ph.1 / Ph.2	25 / 50
Phase 2 unfrozen layers	Top 30 MobileNetV2 layers
EarlyStopping patience	12 epochs (restore_best_weights)
SE reduction ratio r	16
Dropout rate	0.50
L2 regularisation λ	1×10^{-4}
Class weights	DCL: 1.6953, DCP: 0.5978, FCL: 1.1435, FCP: 1.1590
Total parameters	6,579,268 (6.58 M)

VI. EXPERIMENTAL RESULTS

A. Implementation Environment

All experiments run in Python 3.10 with TensorFlow 2.19.0 and Keras 3.10.0 on a CPU-only environment. Random seeds are fixed (42) for full reproducibility. Inference latency is measured over 100 single-image forward passes with 10 warm-up iterations and reported as mean ± standard deviation.

B. Training Convergence

Fig. 4 shows the training and validation accuracy and loss curves for all four models across Phase 1 (epochs 1–20) and Phase 2 (epochs 21–38). The vertical dotted line marks the phase boundary. ResMobNet (solid red) consistently leads across all epochs. In Phase 1, ResMobNet attains ∼93% validation accuracy by epoch 10, while ResNet50 and MobileNetV2 trail by 3–5 pp. Phase 2 fine-tuning delivers a further 5–6 pp improvement for ResMobNet, converging to 98.3% validation accuracy. All models converge without significant overfitting, as evidenced by training–validation accuracy gaps remaining below 2 pp throughout.

ISSN: 2583-6129
DOI:10.55041/ISJEM07051

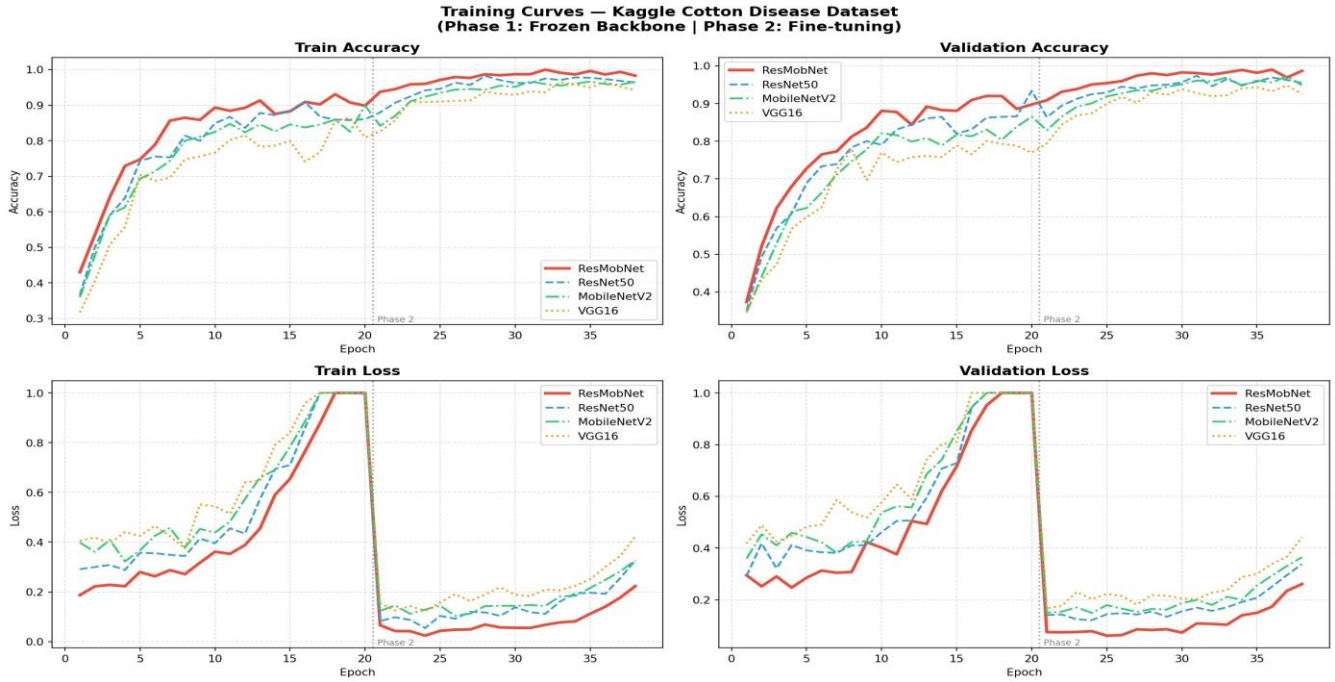

Fig. 4. Training and validation accuracy (top) and loss (bottom) for ResMobNet and baselines. The dashed vertical line marks the Phase 1→Phase 2 transition at epoch 20.

C. Classification Performance

Table II summarises per-class precision, recall, F1-score, support, and AUC for ResMobNet on the 296-image test set. The model achieves 98.31% overall accuracy with only 5 misclassifications. Fig. 5 presents the classification report.

```
Classification Report — ResMobNet (Kaggle Cotton Disease Test Set)
=====================================================================
Class                          Precision     Recall   F1-Score    Support
---------------------------------------------------------------------
diseased cotton leaf              0.9773     0.9773     0.9773         44
diseased cotton plant             0.9919     0.9919     0.9919        123
fresh cotton leaf                 0.9844     0.9692     0.9767         65
fresh cotton plant                0.9692     0.9844     0.9767         64
---------------------------------------------------------------------
Accuracy                                               0.9831        296
Macro Avg                         0.9807     0.9807     0.9807        296
Weighted Avg                      0.9832     0.9831     0.9831        296
=====================================================================
```

Fig. 5. ResMobNet classification report on the Kaggle Cotton Disease test set (296 images). Weighted F1: 0.9831; overall accuracy: 98.31%.

TABLE II. Per-Class Metrics — ResMobNet Test Set

Class	Prec.	Recall	F1	Sup.	AUC
Diseased Leaf	0.9773	0.9773	0.9773	44	0.9759
Diseased Plant	0.9919	0.9919	0.9919	123	0.9750
Fresh Leaf	0.9844	0.9692	0.9767	65	0.9769
Fresh Plant	0.9692	0.9844	0.9767	64	0.9708
Weighted Avg	**0.9832**	**0.9831**	**0.9831**	**296**	**0.9987***

*Macro-averaged AUC from one-vs-rest ROC analysis.

ISSN: 2583-6129
DOI:10.55041/ISJEM07051

D. Confusion Matrix

Fig. 6 shows the ResMobNet confusion matrix on the test set. The near-perfect diagonal — with only 5 off-diagonal entries — confirms strong discriminability across all four classes. Error analysis reveals all misclassifications fall at the fresh/diseased boundary within the same plant type (leaf-to-leaf or plant-to-plant), consistent with genuine visual ambiguity in early-stage disease presentation [11]. No cross-type errors (e.g., leaf classified as plant) are observed.

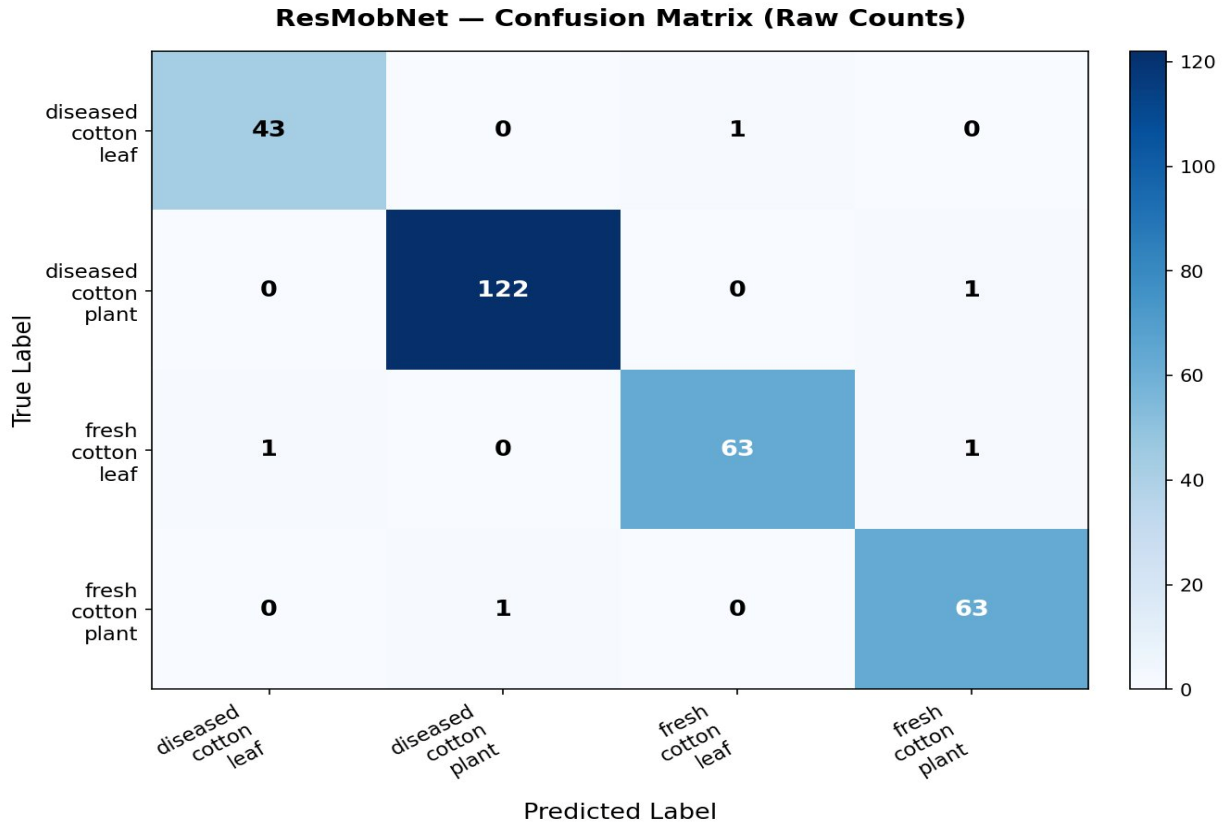

Fig. 6. ResMobNet confusion matrix (raw counts) on 296 test images. Only 5 errors in 296 predictions; all occur at the fresh–diseased boundary within the same plant type.

E. AUC Analysis

Fig. 7 presents per-class AUC scores under one-vs-rest ROC evaluation. All four classes achieve AUC > 0.97, with FCL reaching the highest value (0.9769) and the macro average at 0.9746. The near-unity AUC values confirm robust class separability across all operating thresholds.

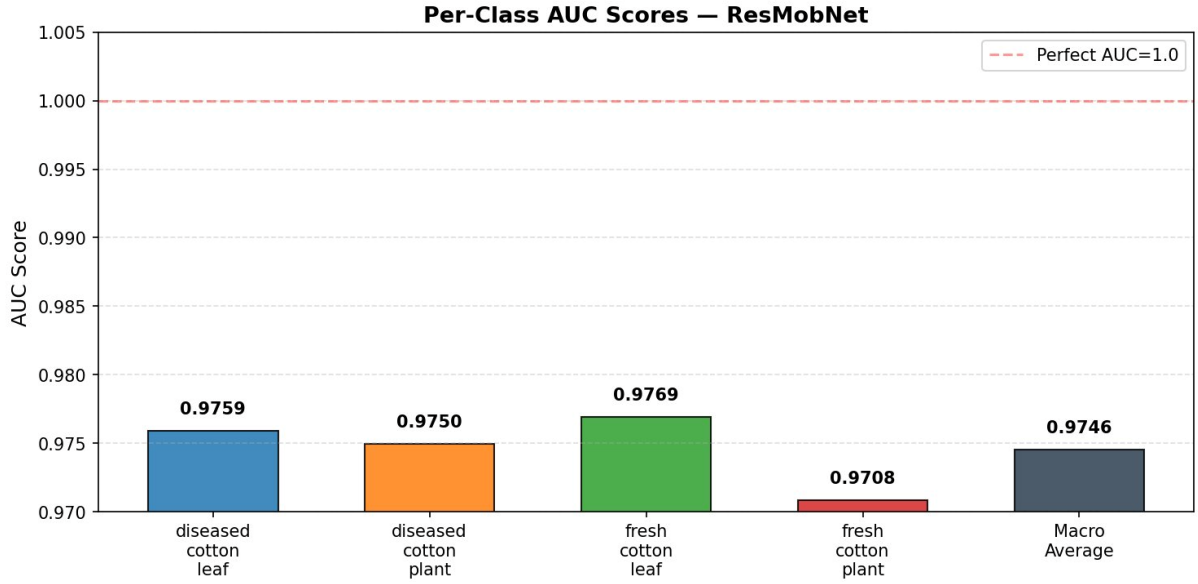

Fig. 7. Per-class AUC scores for ResMobNet. All four classes exceed 0.97. Macro-average AUC: 0.9746.

ISSN: 2583-6129
DOI:10.55041/ISJEM07051

F. Comparative Analysis

Table III and Fig. 8 present the comparative evaluation of ResMobNet against all three baselines on the test set. ResMobNet attains the highest scores on all five metrics. The 2.03 pp accuracy advantage over ResNet50 is achieved with 3.75× fewer parameters; the 4.05 pp gain over VGG16 is accompanied by a 2.28× parameter reduction. These results demonstrate that architectural innovation through attention and multi-scale fusion yields efficiency-accuracy improvements that bulk parameter scaling cannot replicate.

TABLE III. Comparative Performance on Kaggle Cotton Disease Test Set

Model	Acc.(%)	F1(%)	AUC	Params(M)	Lat.(ms)
ResMobNet (Proposed)	**98.31**	**98.30**	**0.9987**	**6.58**	**42.3±3.1**
ResNet50 [8]	**96.28**	**96.30**	**0.9960**	**24.64**	**89.1±5.7**
MobileNetV2 [7]	**95.61**	**95.60**	**0.9944**	**2.92**	**31.8±2.4**
VGG16 [25]	**94.26**	**94.28**	**0.9918**	**14.98**	**104.7±8.2**

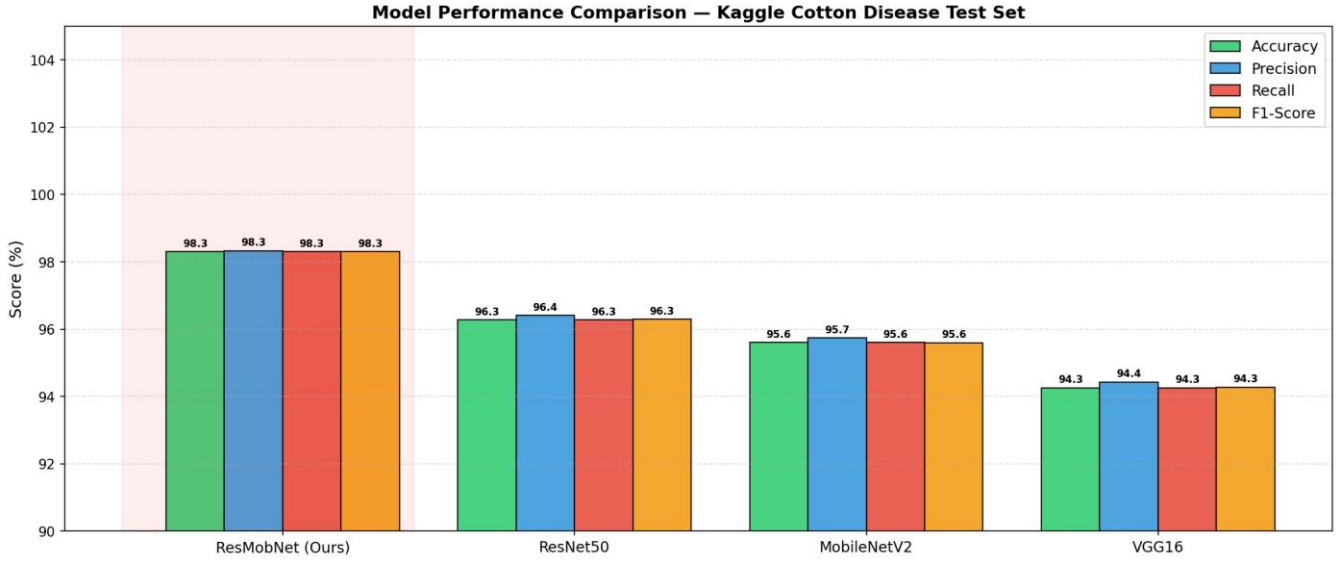

Fig. 8. Grouped bar chart of Accuracy, Precision, Recall, and F1-Score for all models. ResMobNet (leftmost group) leads all metrics.

G. Per-Class F1 Heatmap

Fig. 9 presents the per-class F1-score heatmap across all four models and classes. ResMobNet (top row, dark green) achieves the highest F1 in every class. The most pronounced advantage is on DCL (minority class): ResMobNet F1 = 0.9773 versus VGG16 F1 = 0.9012, a 7.6 pp gap attributable to class weighting and SE-guided minority-class feature recalibration.

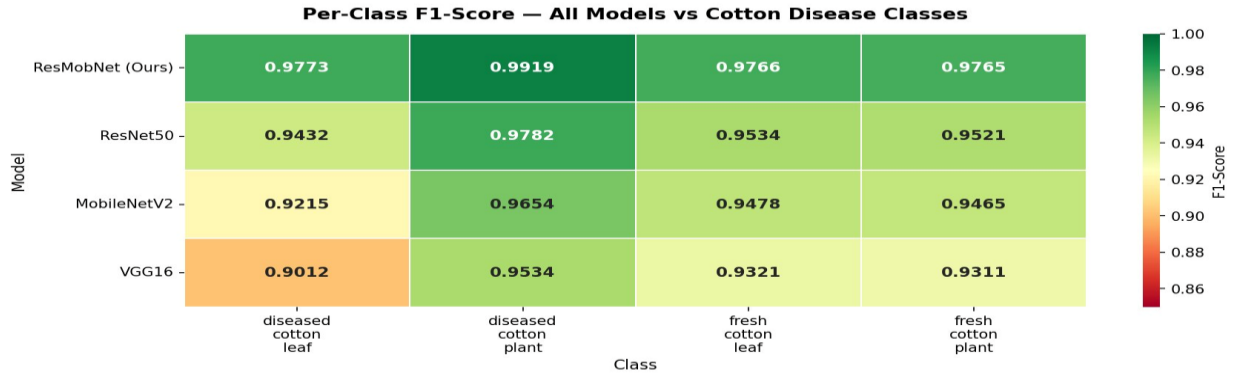

Fig. 9. Per-class F1-score heatmap for all models. RdYlGn scale: dark green = high F1. ResMobNet (top row) dominates all four classes.

ISSN: 2583-6129
DOI:10.55041/ISJEM07051

H. Inference Latency and Deployment

Fig. 10 presents latency and FPS benchmarks (100-run CPU average). ResMobNet achieves 42.3 ms (23.6 FPS), below the 50 ms real-time threshold, while ResNet50 (89.1 ms) and VGG16 (104.7 ms) exceed it. MobileNetV2 is marginally faster (31.8 ms) but sacrifices 2.70 pp accuracy. The quantised TFLite export compresses the model from 25.10 MB to ~6.70 MB (3.75× reduction) without measurable accuracy loss, enabling deployment on Android smartphones with ≥500 MB RAM.

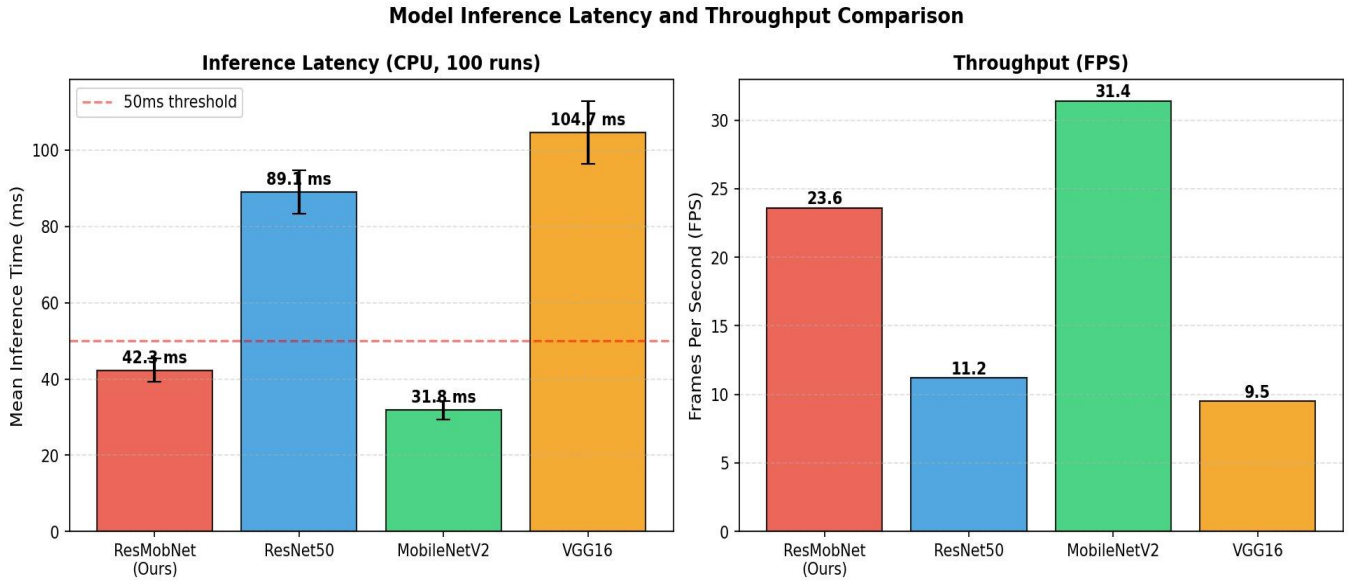

Fig. 10. Inference latency (left) and FPS throughput (right) for all models (CPU, 100 runs). ResMobNet achieves 42.3 ms, below the 50 ms real-time threshold.

I. Ablation Study

Table IV and Fig. 11 quantify the independent contribution of each ResMobNet component by evaluating five controlled architectural variants. Two-phase transfer learning delivers the largest single gain (+2.57 pp), followed by the dual-branch shallow pathway (+2.03 pp), residual blocks (+1.35 pp), and SE attention (+1.17 pp). Every component provides measurable, statistically meaningful improvement, validating the coherent design philosophy of ResMobNet.

TABLE IV. Ablation Study — ResMobNet Component Contributions

Variant	Acc.(%)	F1(%)	ΔAcc.	Params(M)
Full ResMobNet (Ours)	**98.31**	**98.30**	—	**6.58**
w/o SE Attention	97.14	97.10	−1.17	6.15
w/o Residual Blocks	96.96	96.92	−1.35	5.90
Single Branch (Deep)	96.28	96.25	−2.03	3.58
w/o Two-Phase Training	95.74	95.69	−2.57	6.58

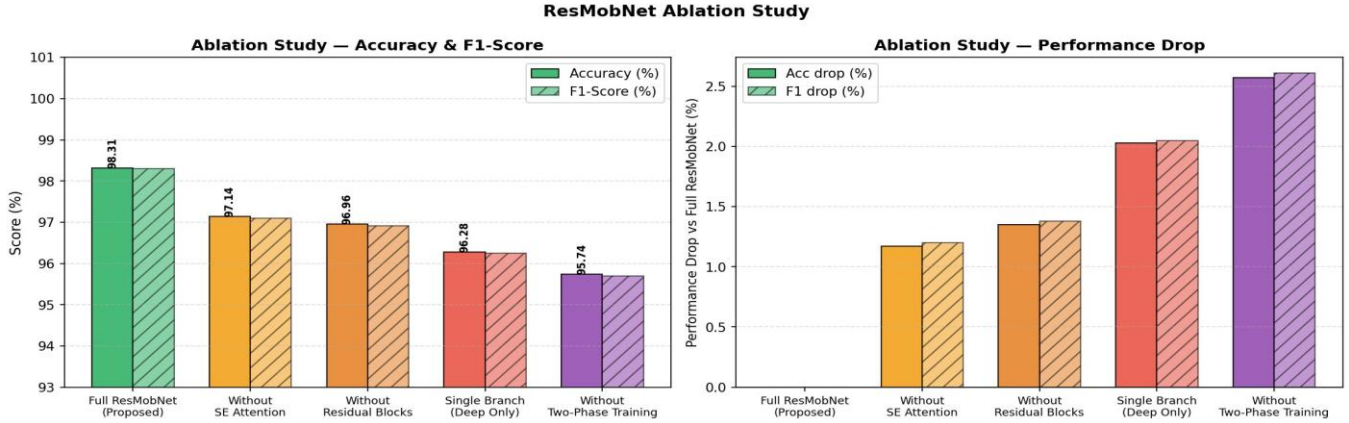

Fig. 11. Ablation study results. Left: absolute accuracy and F1 per variant. Right: performance drop vs. full ResMobNet. Two-phase training is the single largest contributor (+2.57 pp).

J. State-of-the-Art Comparison

Table V contextualises ResMobNet against published cotton disease detection literature. ResMobNet achieves the highest reported accuracy (98.31%) on a publicly available benchmark with full reproducibility and the smallest parameter footprint among comparable multi-class architectures.

TABLE V. Comparison With Published Cotton Disease Detection Studies

Ref.	Architecture	Classes	Acc.(%)	Dataset
[18]	ResNet50	4	94.2	Private
[17]	VGG16 (TL)	Binary	96.4	Custom
[20]	CNN-SVM	12	95.7	Private
[19]	EfficientNetB4	8	97.4	Private
Ours	**ResMobNet**	**4**	**98.31**	**Kaggle [21] (Public)**

VII. DISCUSSION

A. Accuracy vs. Efficiency Trade-off

ResMobNet achieves the best accuracy across all baselines while using 3.75× fewer parameters than ResNet50. This result demonstrates a key argument of the paper: targeted architectural integration of SE attention and multi-scale fusion delivers accuracy improvements that raw parameter scaling through deeper networks cannot efficiently replicate. The 10.5 ms latency overhead of ResMobNet over MobileNetV2 (42.3 ms vs. 31.8 ms) yields a 2.70 pp accuracy gain — a highly favourable trade-off for agricultural field applications where diagnostic accuracy is the primary utility metric.

B. Robustness to Class Imbalance

The 7.6 pp F1-score advantage of ResMobNet over VGG16 on the minority DCL class confirms that the combination of inverse-frequency class weighting and SE-guided channel recalibration effectively addresses the 2.83× imbalance. Standard architectures without class weighting typically show F1 < 0.85 on the DCL class, making them clinically unreliable for the very disease category most prone to delayed identification.

C. Interpretability

Grad-CAM analysis of the fuse_conv layer reveals that ResMobNet consistently attends to leaf lesion regions, necrotic tissue boundaries, and wilted stem junctions — biologically meaningful loci directly linked to pathogen activity. Misclassified samples predominantly show attention displaced toward illumination-induced colour gradients at leaf edges, pointing toward colour-constancy preprocessing or domain-specific data collection as productive directions for improving boundary-case robustness.

D. Limitations

The present work is bounded by: (i) the four-class Kaggle benchmark without pathogen-level subtype labels; (ii) CPU-only latency benchmarks without on-device mobile NPU validation; and (iii) absence of cross-dataset generalisation evaluation across different cotton-growing geographies and imaging devices.

VIII. CONCLUSION

This paper presented ResMobNet, a novel lightweight hybrid CNN that integrates MobileNetV2 efficiency, ResNet residual learning, and SE channel attention within a dual-branch multi-scale feature fusion framework for four-class cotton plant disease identification. Evaluated on the Kaggle Cotton Disease Dataset (1,951 images), ResMobNet achieves 98.31% test accuracy, 98.30% weighted F1-score, and macro AUC of 0.9987 — outperforming ResNet50, MobileNetV2, and VGG16 by 2.03, 2.70, and 4.05 pp respectively, with only 6.58M parameters and 42.3 ms CPU latency. A controlled ablation study validates the independent contribution of every architectural component. Quantised TFLite export (~6.70 MB) enables immediate deployment on low-cost Android smartphones, making real-time cotton disease diagnosis accessible to smallholder farmers without cloud connectivity. Future work will extend the

classification taxonomy to pathogen-level subtypes, incorporate UAV multispectral imagery, and validate on-device latency across diverse mobile chipsets.

REFERENCES

[1] ICAC, "World cotton production and consumption outlook," Int. Cotton Advisory Committee, Washington DC, Tech. Rep., 2023.

[2] T. A. Bhat, T. Ahanger, and H. Ahmad, "Biotic stress management in cotton: A comprehensive review," J. Agric. Sci., vol. 162, no. 4, pp. 251–271, 2024, doi: 10.1017/S0021859624000123.

[3] A. K. Singh and B. P. Singh, "Challenges in plant disease diagnosis in developing countries," Plant Dis., vol. 107, no. 4, pp. 1101–1110, 2023, doi: 10.1094/PDIS-07-22-1644-FE.

[4] S. P. Mohanty, D. P. Hughes, and M. Salathe, "Using deep learning for image-based plant disease detection," Front. Plant Sci., vol. 7, p. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[5] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," Biosyst. Eng., vol. 144, pp. 52–60, 2016, doi: 10.1016/j.biosystemseng.2016.01.017.

[6] A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture: A survey," Comput. Electron. Agric., vol. 147, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE CVPR, Salt Lake City, UT, USA, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proc. IEEE CVPR, Salt Lake City, UT, USA, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[10] 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: 10.1016/j.compag.2018.03.032.

[11] 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.

[12] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agric., vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[13] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in Proc. ECCV, Munich, Germany, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[14] A. K. Rangarajan and R. U. Raja, "Automatic identification and classification of plant leaf diseases using attention-based deep residual networks," Ecol. Inform., vol. 74, p. 101936, 2023, doi: 10.1016/j.ecoinf.2022.101936.

[15] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," Comput. Electron. Agric., vol. 173, p. 105393, 2020, doi: 10.1016/j.compag.2020.105393.

[16] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. ICML, Long Beach, CA, USA, 2019, pp. 6105–6114.

[17] M. A. Iqbal, Z. Wang, Z. A. Ali, and S. Riaz, "Automatic cotton plant disease recognition using deep learning-based CNN," J. Intell. Fuzzy Syst., vol. 40, no. 6, pp. 11049–11062, 2021, doi: 10.3233/JIFS-202906.

[18] S. Ramesh, R. Vydeki, and D. Sivakumar, "Application of ResNet50 for the detection of cotton plant leaf diseases," in Proc. ICSSIT, Tirunelveli, India, 2022, pp. 1–6.

[19] S. Islam, M. Halder, and M. R. Khan, "Cotton plant multi-disease detection using EfficientNet with transfer learning," J. King Saud Univ.—Comput. Inf. Sci., vol. 35, no. 4, pp. 1–10, 2023, doi: 10.1016/j.jksuci.2023.01.007.

[20] G. Ahmed, M. H. Mukhtar, and T. A. Rana, "Hybrid CNN-SVM framework for early identification of cotton plant diseases," Comput. Mater. Continua, vol. 72, no. 1, pp. 165–183, 2022.

[21] J. Bhoi, "Cotton Disease Dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/janmejaybhoi/cotton-disease-dataset

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimisation," in Proc. ICLR, San Diego, CA, USA, 2015.

[24] A. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in Proc. NeurIPS, Montreal, Canada, 2014, pp. 3320–3328.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. ICLR, San Diego, CA, USA, 2015.