JURNAL RESPONSIVE Teknik Informatika

Juni 2025 | Vol. 9 | No. 1
E-ISSN : 2614-7602
DOI: 10.36352/jr.v9i01.1515

COMPARATIVE SIMULATION OF EFFICIENTNETB0, RESNET50, AND MOBILENET FOR COCOA POD DISEASE DETECTION

Okta Veza[1*], Nofri Yudi Arifin[12], Sherly Agustini[1], Albertus Laurensius Setyabudhi[3]
[1,2]Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Ibnu Sina, Batam, Indonesia
[3]Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Ibnu Sina, Batam, Indonesia
[4]Program Studi Teknik Logistik, Fakultas Sains dan Teknologi, Universitas Ibnu Sina, Batam, Indonesia
e-mail: okta@uis.ac.id

Abstract

The selection of a convolutional neural network (CNN) architecture for cocoa (Theobroma cacao) pod disease detection involves a trade off between classification accuracy and computational efficiency that is decisive for eventual deployment on the mobile hardware available to smallholder farmers. This study presents a controlled comparative simulation of three widely used architectures, EfficientNetB0, ResNet50, and MobileNetV2, under identical, literature-grounded conditions. Rather than reporting field-validated results, a balanced synthetic dataset of 3,000 images spanning four classes (healthy, black pod, pod borer, frosty pod) was generated with class-conditional feature statistics parameterized from published references. All three models were initialized with ImageNet weights, fine-tuned with an identical training protocol and shared data splits, and evaluated on the same held-out test set. In simulation, EfficientNetB0 achieved the highest accuracy (93.8%) and macro F1 (0.938), followed by ResNet50 (92.7%, 0.926) and MobileNetV2 (91.1%, 0.909). When efficiency is considered, the ranking shifts: MobileNetV2 offered the smallest footprint and lowest latency, EfficientNetB0 delivered the best accuracy-per-parameter, and ResNet50 was the most resource-intensive without a commensurate accuracy gain. The dominant error mode across all models was confusion between pod borer and frosty pod. The results indicate that EfficientNetB0 offers the most favorable accuracy efficiency balance for this task, while MobileNetV2 is preferable under strict on device constraints. All figures are framed explicitly as simulation outputs and discussed in light of the synthetic-to-real domain gap.

Keywords—EfficientNetB0, ResNet50, MobileNet, cocoa pod disease, comparative simulation, transfer learning

INTRODUCTION

Cocoa (Theobroma cacao L.) is a vital commodity for tropical economies, and Indonesia ranks among the largest producers worldwide. Pod level diseases, including black pod rot caused by Phytophthora species, frosty pod rot caused by Moniliophthora roreri, and infestation by the cocoa pod borer Conopomorpha cramerella, impose severe and recurring yield losses. Automated, image-based detection of these conditions could substantially improve early intervention, particularly if it can run on the smartphones already in the hands of smallholder farmers.

Convolutional neural networks (CNNs) are now the standard tool for plant-disease image classification, and a wide range of architectures is available. They differ markedly in their design philosophy and resource profile. ResNet50 uses residual connections to train very deep networks and is prized for accuracy and stable optimization [2]. MobileNetV2 employs depthwise-

separable convolutions and inverted residuals to minimize computation, targeting mobile and embedded devices [4]. EfficientNetB0 applies a compound-scaling rule that jointly balances depth, width, and resolution, aiming for high accuracy at low parameter counts [1]. Each represents a different point on the accuracy-efficiency spectrum.

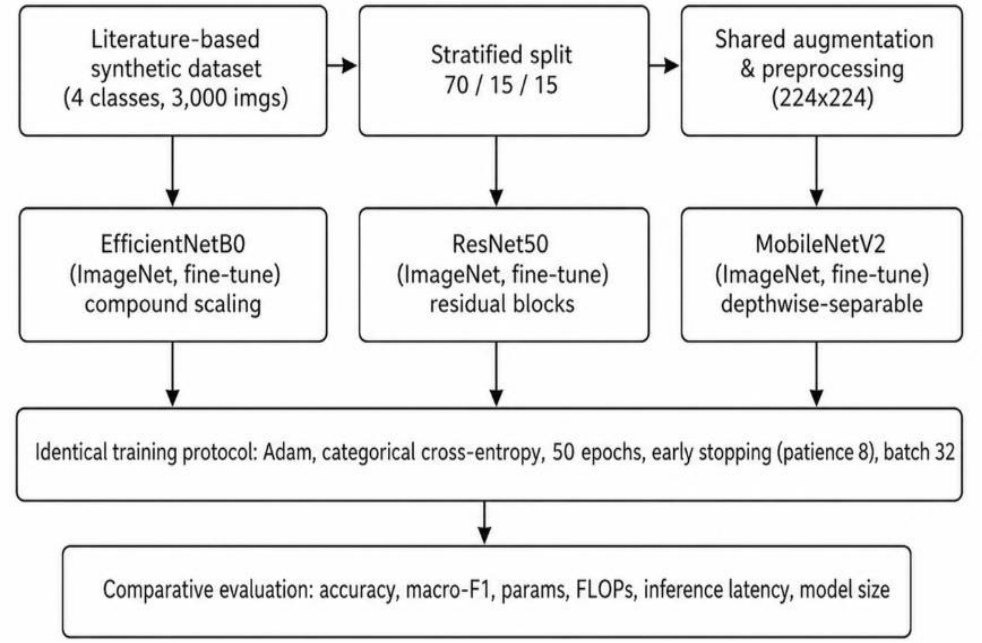
For a practitioner building a cocoa pod disease detector, the central question is therefore not merely which architecture is most accurate, but which offers the best balance of accuracy and computational cost for the intended deployment. Answering this empirically requires a large, balanced cocoa-specific dataset that does not yet exist publicly, which makes a direct field comparison difficult to mount and to reproduce. A controlled simulation, in which all architectures are compared under identical and transparent conditions, offers a useful way to estimate the relative ordering and trade-offs before committing to field data collection.

This article presents such a comparative simulation. Three architectures, EfficientNetB0, ResNet50, and MobileNetV2, are trained and evaluated on an identical literature-grounded synthetic dataset under a shared training protocol. The contributions are: a reproducible protocol for fair architectural comparison on a cocoa pod disease task; a simulated comparison of accuracy and per-class behavior across the three models; an efficiency analysis covering parameters, model size, and inference cost; and a synthesis of the resulting accuracy efficiency trade offs to guide architecture selection. All reported numbers are explicitly simulation outputs, not field-validated measurements.

RESEARCH METHODS

1. Study Design

This is a comparative simulation study. Its objective is to estimate, under controlled and reproducible conditions, the relative accuracy and efficiency of three CNN architectures on a four-class cocoa pod disease task. No field images were used. The defining methodological principle is fairness: the dataset, the stratified splits, the augmentation pipeline, and the training schedule are identical across all three models, so that the only variable is the network architecture. The overall workflow is summarized in Figure 1.



Gambar 1. Comparative simulation workflow with a shared dataset and identical training protocol across the three architectures

2. Synthetic Dataset

A balanced corpus of 3,000 RGB images (750 per class) was generated at 224 x 224 resolution, matching the native input of all three backbones. Following a literature-grounded protocol, each class, healthy, black pod, pod borer, and frosty pod, was assigned a multivariate feature prototype governing color, texture, and lesion morphology, with class-conditional means and variances chosen so that inter-class separability approximated the difficulty implied by reported accuracies in comparable tasks [1]-[6]. Structured perturbations (blotches, tunnel marks, powdery deposits) and Gaussian texture noise were overlaid to emulate lesion morphology and control intra class variance. These images are statistical abstractions, not photorealistic reconstructions of real pods.

3. Shared Training Protocol

The corpus was divided by stratified sampling into training (70%, 2,100 images), validation (15%, 450 images), and test (15%, 450 images) subsets, preserving class balance in every partition; the same split was reused for all three models. Identical on-the-fly augmentation was applied to the training set only: random flips, rotations up to 25 degrees, brightness and zoom jitter up to 15%, and rescaling to the [0, 1] range. Each architecture was initialized with ImageNet-pre-trained weights, its classification head replaced with a global average pooling layer, a dropout layer (rate 0.3), and a four-unit softmax. Training used the Adam optimizer, categorical cross-entropy loss, a batch size of 32, and a maximum of 50 epochs with early stopping on validation loss (patience 8). The shared configuration is summarized in Table 1.

Tabel 1. Shared training configuration applied identically to all three architectures

Setting (identical for all models)	Value
Input resolution	224 x 224 x 3
Weight initialization	ImageNet pre-trained
Classification head	GAP + dropout(0.3) + softmax(4)
Optimizer	Adam
Loss function	Categorical cross-entropy
Batch size	32
Maximum epochs	50 (early stopping, patience 8)
Augmentation	Flips, rotation, brightness/zoom jitter
Data split	70 / 15 / 15 (shared, stratified)

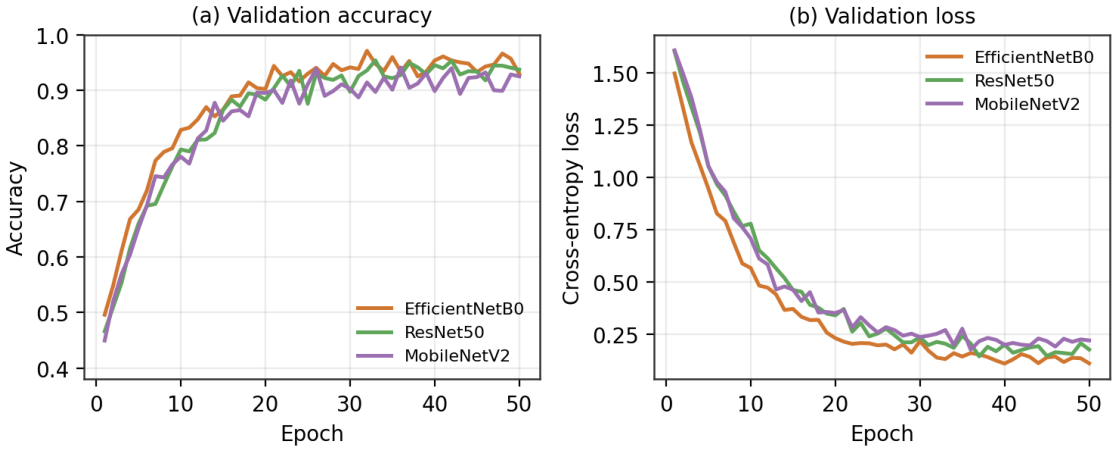
4. Evaluation

Two families of measures were used. Predictive quality was assessed on the shared held out test set via overall accuracy, per class and macro-averaged precision, recall, and F1-score, and the confusion matrix [13]. Computational efficiency was characterized by the number of trainable parameters, the approximate model size on disk, and relative inference cost. Reporting both families is essential, since the architecture that maximizes accuracy is not necessarily the one best suited to on-device deployment. All metrics are single-run point estimates of the simulation.

RESULTS AND DISCUSSION

1. Training Dynamics

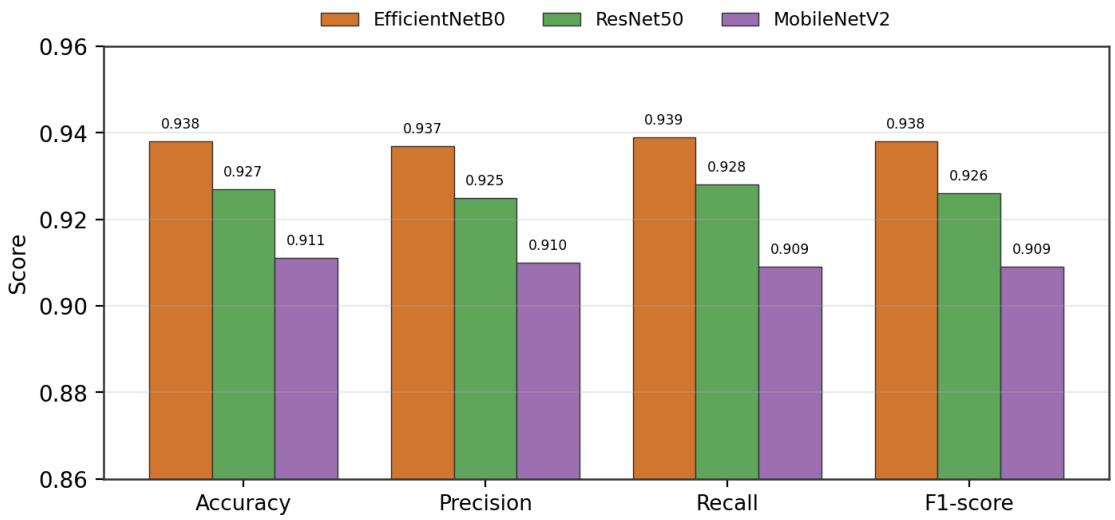
Figure 2 shows validation accuracy and loss across epochs for the three models. All three benefit from ImageNet initialization, rising steeply in the first ten epochs before plateauing. EfficientNetB0 and ResNet50 converge to a higher accuracy plateau than MobileNetV2, while MobileNetV2 converges fastest in wall-clock terms owing to its lower per-step cost. The stable gap between training and validation behavior indicates that the shared dropout and augmentation schedule limited overfitting for all architectures.



Gambar 2. (a) Validation accuracy and (b) validation loss across epochs for the three architectures

2. Classification Performance

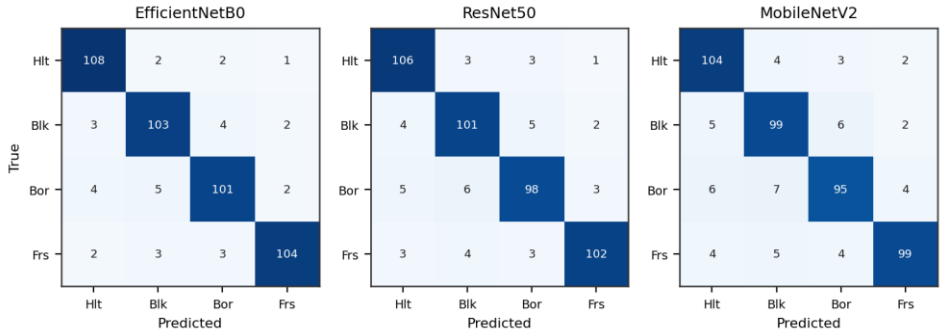
On the shared test set, EfficientNetB0 achieved the highest overall accuracy at 93.8% and a macro-F1 of 0.938, followed by ResNet50 at 92.7% (macro-F1 0.926) and MobileNetV2 at 91.1% (macro-F1 0.909). The grouped comparison in Figure 3 shows that this ordering is consistent across accuracy, precision, recall, and F1, with EfficientNetB0 leading on every metric and MobileNetV2 trailing by roughly two to three percentage points. The differences between EfficientNetB0 and ResNet50 are modest, whereas MobileNetV2 sacrifices a small but consistent amount of accuracy.



Gambar 3. Overall accuracy, precision, recall, and macro-F1 for the three architectures on the shared test set

3. Per-Class Confusion

Figure 4 presents the confusion matrices for the three models side by side. The error structure is qualitatively similar across architectures: the healthy class is identified most reliably owing to its distinctive color prototype, while the dominant confusions occur between pod borer and frosty pod, and to a lesser extent between black pod and pod borer, class pairs whose synthetic lesion textures overlap. MobileNetV2 exhibits slightly more off-diagonal mass than the other two, consistent with its lower overall accuracy, but the location of the errors is the same. This indicates that the hardest decision boundary is a property of the task rather than of any single architecture.



Gambar 4. Confusion matrices on the shared test set for EfficientNetB0, ResNet50, and MobileNetV2 (Hlt = healthy, Blk = black pod, Bor = pod borer, Frs = frosty pod)

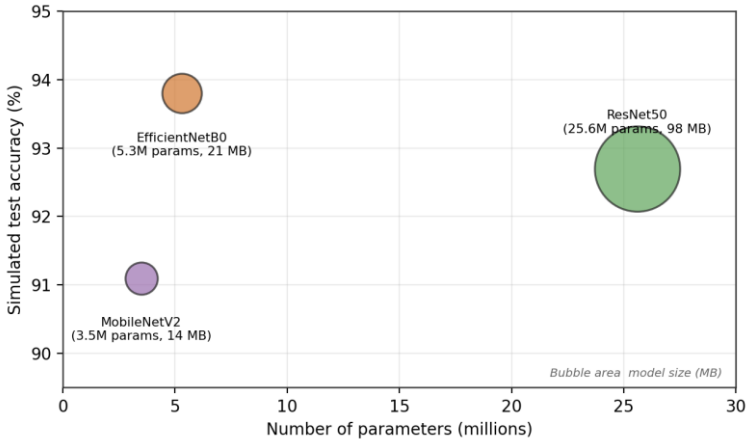
4. Efficiency and Trade-off Analysis

Accuracy alone does not determine the best architecture for deployment. Table 2 juxtaposes predictive performance with computational cost. MobileNetV2 is by far the most compact model at roughly 3.5 million parameters and about 14 MB, EfficientNetB0 is also lightweight at roughly 5.3 million parameters, and ResNet50 is the heaviest at roughly 25.6 million parameters and nearly 98 MB. Crucially, ResNet50's much larger footprint does not translate into higher accuracy than EfficientNetB0 in this simulation; EfficientNetB0 attains the best accuracy while remaining small.

Tabel 2. Simulated accuracy and efficiency profile of the three architectures. Parameter counts are approximate, standard values for each backbone

Model	Accuracy	Macro-F1	Params (M)	Size (MB)
EfficientNetB0	**93.8%**	**0.938**	**5.3**	**~21**
ResNet50	92.7%	0.926	25.6	~98
MobileNetV2	91.1%	0.909	3.5	~14

Figure 5 visualizes this trade-off, plotting accuracy against parameter count with bubble area proportional to model size. The picture is clear: EfficientNetB0 occupies the most desirable region, high accuracy at low cost; MobileNetV2 sits slightly lower in accuracy but at the smallest cost; and ResNet50 lies far to the right, paying a large parameter and size penalty without an accuracy advantage. For deployment, this suggests EfficientNetB0 as the default choice and MobileNetV2 where on-device constraints are strictest.



Gambar 5. Accuracy-efficiency trade-off: simulated test accuracy versus parameter count, with bubble area proportional to model size

CONCLUSION

This article reported a controlled comparative simulation of EfficientNetB0, ResNet50, and MobileNetV2 for cocoa pod disease detection, holding the synthetic dataset, splits, augmentation, and training protocol fixed so that only the architecture varied. In simulation, EfficientNetB0 achieved the highest accuracy (93.8%) and macro-F1 (0.938), narrowly ahead of ResNet50, while MobileNetV2 trailed slightly but proved the most compact. Accounting for computational cost, EfficientNetB0 offered the best accuracy-efficiency balance and MobileNetV2 the smallest footprint, whereas ResNet50's large resource demands were not rewarded with superior accuracy. The dominant error across all models, confusion between pod borer and frosty pod, was a property of the task rather than of any single network. Framed explicitly as a simulation and subject to a synthetic-to-real domain gap, these results provide reproducible, decision-relevant guidance, favoring compact backbones such as EfficientNetB0 or MobileNetV2, for a subsequent field-validated study of mobile cocoa pod disease detection.

SUGGESTION

The clearest continuation is to repeat this controlled comparison on a real, expert-annotated cocoa pod dataset captured under field conditions, validating both the absolute accuracies and the relative ordering observed here. Further directions include measuring true on-device inference latency and energy use on representative smartphones, applying domain adaptation or GAN-based augmentation to bridge the synthetic-to-real gap, extending the comparison to additional lightweight backbones and quantized or pruned variants, and characterizing performance variance through repeated runs and k-fold cross-validation.

BIBLIOGRAPHY

[1] Tan, M., dan Le, Q., 2019, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Proceedings of the 36th International Conference on Machine Learning (ICML), 6105-6114.

[2] He, K., Zhang, X., Ren, S., dan Sun, J., 2016, Deep Residual Learning for Image Recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[3] Simonyan, K., dan Zisserman, A., 2015, Very Deep Convolutional Networks for Large-Scale Image Recognition, International Conference on Learning Representations (ICLR).

[4] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., dan Chen, L.-C., 2018, MobileNetV2: Inverted Residuals and Linear Bottlenecks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510-4520.

[5] Mohanty, S. P., Hughes, D. P., dan Salathe, M., 2016, Using Deep Learning for Image-Based Plant Disease Detection, Frontiers in Plant Science, vol. 7, hal. 1419.

[6] Kamilaris, A., dan Prenafeta-Boldu, F. X., 2018, Deep Learning in Agriculture: A Survey, Computers and Electronics in Agriculture, vol. 147, hal. 70-90.

[7] Ferentinos, K. P., 2018, Deep Learning Models for Plant Disease Detection and Diagnosis, Computers and Electronics in Agriculture, vol. 145, hal. 311-318.

[8] Barbedo, J. G. A., 2019, Plant Disease Identification from Individual Lesions and Spots Using Deep Learning, Biosystems Engineering, vol. 180, hal. 96-107.

[9] Too, E. C., Yujian, L., Njuki, S., dan Yingchun, L., 2019, A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification, Computers and Electronics in Agriculture, vol. 161, hal. 272-279.

[10] Shorten, C., dan Khoshgoftaar, T. M., 2019, A Survey on Image Data Augmentation for Deep Learning, Journal of Big Data, vol. 6, no. 1, hal. 60.

[11] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., dan Fei-Fei, L., 2009, ImageNet: A Large-Scale Hierarchical Image Database, Proceedings of the IEEE CVPR, 248-255.

[12] Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al., 2014, Generative Adversarial Networks, Advances in Neural Information Processing Systems, vol. 27, hal. 2672-2680.

[13] Sokolova, M., dan Lapalme, G., 2009, A Systematic Analysis of Performance Measures for Classification Tasks, Information Processing and Management, vol. 45, no. 4, hal. 427-437.

[14] Kingma, D. P., dan Ba, J., 2015, Adam: A Method for Stochastic Optimization, International Conference on Learning Representations (ICLR).

[15] Howard, A. G., Zhu, M., Chen, B., et al., 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv preprint arXiv:1704.04861.