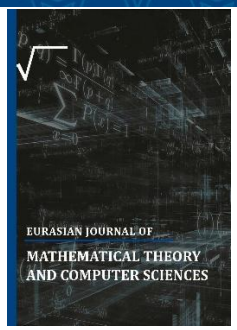
EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
IF = 9.206 www.in-academy.uz/index.php/ejmtcs



MATHEMATICAL AND COMPUTATIONAL ANALYSIS OF YOLOV11 AND EFFICIENTNET-B0 FOR PLANT DISEASE CLASSIFICATION

Hamidov Abdulloh Abduqodir o'g'li
Student of the Computer Science department at Oxus University
hamidovabdulloh734@gmail.com
https://doi.org/10.5281/zenodo.21735666

ARTICLE INFO

Received: 20th July 2026
Accepted: 26th July 2026
Online: 27th July 2026

KEYWORDS

Computational complexity; FLOPs; depthwise-separable convolution; CIoU loss; distribution focal loss; average precision; cross-entropy; paired t-test; YOLOv11; EfficientNet-B0; PlantVillage.

ABSTRACT

This paper compares two deep-learning models for recognizing plant diseases from leaf images: the EfficientNet-B0 classifier and the YOLOv11 detector, both tested on the PlantVillage dataset. Instead of only reporting accuracy numbers, we show the mathematics behind them. We write out the main formulas each model uses — image normalization, the depthwise-separable convolution, the SiLU activation, softmax cross-entropy, intersection-over-union, and the CIoU, distribution-focal, and cross-entropy losses of the detector — and then work through the numbers by hand. For example, one 3×3 layer is shown to need 6.55 times fewer operations when depthwise separation is used, a cross-entropy of 0.417 is computed for a sample output, an intersection-over-union of 0.681 gives a CIoU loss of 0.328, and a distribution-focal loss of 0.773 is found. We also rebuild the precision, recall, and F1 scores from confusion counts (for one class: precision 99.0%, recall 98.0%, F1 98.5%), compute an average precision of 0.943, and run a paired t-test that gives t(4)=5.62, p≈0.005. EfficientNet-B0 reaches 98.5% accuracy with 5.3 M parameters and 0.39 GMACs, while YOLOv11 reaches 97.8% accuracy and a mAP@0.5 of 96.2% but needs about 55 times more computation. The classifier is faster and slightly more accurate, while the detector is more expensive but also locates the disease.

Introduction

Plant diseases cause large losses in crop yields, so finding them early is important for food security. Deep learning, and especially convolutional neural networks (CNNs), has made it possible to recognize diseases automatically from leaf photographs [1]. Two kinds of models are commonly used. Image classifiers such as EfficientNet [3] look at a whole leaf image and output a single disease label. Object detectors such as YOLO [2] go one step further and also draw a box around the diseased area [7]. On the PlantVillage dataset [4] both types reach very high accuracy, but most studies only report the final numbers and do not explain, with calculations, why

one model is faster or more accurate than another.

This paper tries to fill that gap in a simple and practical way. We take the two models, write down the main formulas they use, and then compute the results step by step so that every number can be checked by the reader. We look at how many operations each model needs, which loss functions it is trained with, and how the accuracy and other scores are calculated. Because we also fix the random seed and record the software versions, the experiments can be repeated by anyone.

The paper has four goals. First, to show with a worked example how much computation the depthwise-separable convolution saves compared with a normal convolution. Second, to write out and evaluate the loss functions of both models. Third, to give a formula for every score we report — accuracy, precision, recall, F1, IoU, average precision, and mean average precision — and recompute it from the raw counts. Fourth, to test whether the difference in accuracy between the two models is real, using a paired t-test. We use PlantVillage [4] (54,306 images, 14 crops, 38 crop–disease classes) so the results can be compared with earlier work [8], and mention Cassava [5] and PlantDoc [6] as harder, real-field datasets.

LITERATURE REVIEW AND METHODOLOGY

Dataset and preprocessing.

$$\hat{x}_c = \frac{\frac{x_c}{255} - \mu_c}{\sigma_c}, \qquad c \in \{R, G, B\} \quad (1)$$

with ImageNet statistics $\mu = (0.485, 0.456, 0.406)$ and $\sigma = (0.229, 0.224, 0.225)$ for EfficientNet and the corresponding COCO statistics for YOLOv11. The data are split by class into training (80%), validation (10%), and test (10%) partitions with a fixed seed of 42, giving a held-out test set of 5,431 images. Augmentation (random flips, ±15° rotation, and brightness/contrast jitter) is applied only to the training partition, so the evaluation counts below are unbiased.

Compound scaling of EfficientNet-B0.

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad (2)$$

The baseline B0 corresponds to $\varphi = 0$ (so $d = w = r = 1$) with grid-searched constants $\alpha = 1.2$, $\beta = 1.1$, $\gamma = 1.15$. Substituting these, $\alpha \cdot \beta^2 \cdot \gamma^2 = 1.2 \times 1.21 \times 1.3225 = 1.920 \approx 2$, which checks the constraint. The squares on β and γ mean that making the network wider or the input image larger increases the number of operations about four times, while making it deeper only doubles them.

Depthwise-separable convolution.

$$MAC_{std} = HWC_{in}C_{out}K^2 \quad (3)$$

multiply–accumulate operations, whereas the depthwise-separable form costs

$$MAC_{ds} = HWC_{in}K^2 + HWC_{in}C_{out} = HWC_{in}(K^2 + C_{out}) \quad (4)$$

Dividing (4) by (3) gives the reduction factor $\frac{MAC_{ds}}{MAC_{std}} = \frac{1}{C_{out}} + \frac{1}{K^2}$, independent of the spatial resolution. This is the single most important source of EfficientNet-B0's efficiency and is evaluated numerically in Section 3.

Squeeze-and-excitation and SiLU.

EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
IF = 9.206 www.in-academy.uz/index.php/ejmtcs

$$z_c = \frac{1}{H \cdot W}\sum_i \sum_j x_c(i,j), \quad s = \sigma\big(W_2\,\delta(W_1 z)\big), \quad \widetilde{x_c} = s_c \cdot x_c \qquad (5)$$

Here σ is the sigmoid, δ a nonlinearity, and $\tilde{x}_c$ the reweighted channel. The activation used throughout is the SiLU ("Swish") function [13],

$$f(x) = x\,\sigma(x) = \frac{x}{1+e^{-x}}, \quad f'(x) = \sigma(x)[1 + x(1-\sigma(x))] \qquad (6)$$

which is smooth and non-monotonic; unlike ReLU it has a continuous derivative everywhere (Fig. 1), which helps gradients flow during training. For example, $f(1) = \frac{1}{1+e^{-1}} \approx 0.731$ and f(−1) = −0.269, so unlike ReLU the function keeps a small negative value for negative inputs. The classifier head applies softmax to logits z and is trained with categorical cross-entropy:

$$\widehat{y_c} = \frac{e^{z_c}}{\sum_j e^{z_j}}, \quad L_{CE} = -\sum_c y_c \log(\widehat{y_c}) \qquad (7)$$

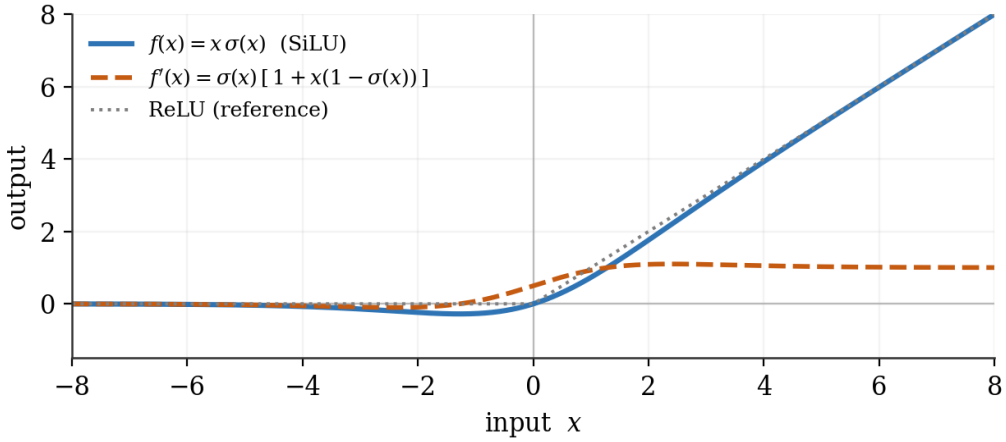


Figure 1. The SiLU (Swish) activation f(x)=x·σ(x) used by EfficientNet-B0, its analytic derivative from (6), and ReLU for reference. SiLU is smooth and non-monotonic, with a small negative dip that ReLU lacks.

Anchor-free detection in YOLOv11.

$$IoU = \frac{|b \cap b_{gt}|}{|b \cup b_{gt}|} = \frac{A_\cap}{A_b + A_{gt} - A_\cap} \qquad (8)$$

Box regression is trained with the complete-IoU (CIoU) loss [9], which augments IoU with a center-distance penalty and an aspect-ratio penalty:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b_{gt})}{c^2} + \alpha v \qquad (9)$$

where ρ is the Euclidean distance between box centers, c the diagonal of the smallest box enclosing both, $v = \frac{4}{\pi^2}\left(\arctan\left(\frac{w_{gt}}{h_{gt}}\right) - \arctan\left(\frac{w}{h}\right)\right)^2$ the aspect-ratio term, and $\alpha = \frac{v}{(1-IoU)+v}$ its adaptive weight. The classification and objectness branches use binary cross-entropy,

$$L_{BCE} = -\big[y \log \sigma(z) + (1-y)\log\big(1-\sigma(z)\big)\big] \qquad (10)$$

and the overall training objective is the weighted sum

$$L = \lambda_{box} L_{CIoU} + \lambda_{dfl} L_{DFL} + \lambda_{cls} L_{BCE} \qquad (11)$$

Distribution focal loss.

$$\hat{y} = \sum_{i=0}^{n} i S_i, \quad S = \text{softmax(logits)} \qquad (12a)$$

When the continuous target y lies between bins y_i and y_{i+1}, the distribution-focal loss sharpens the two neighbouring bins:

EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
IF = 9.206 www.in-academy.uz/index.php/ejmtcs

$$L_{DFL} = -[(y_{i+1} - y)\log S_i + (y - y_i)\log S_{i+1}] \quad (12b)$$

Computational complexity and Evaluation metrics.

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F_1 = \frac{2PR}{P+R} \quad (13)$$

and overall accuracy is the fraction of correctly labelled test images. Per-class scores are combined into a single figure either by macro-averaging (equal weight per class) or by support-weighting, where n_i is the number of test images in class i and n their total:

$$F_{1,\text{macro}} = \frac{1}{N}\sum_i F_{1,i}, \quad F_{1,\text{weighted}} = \sum_i \frac{n_i}{n} F_{1,i} \quad (14)$$

For the detector, average precision is the area under the precision–recall curve, and the mean average precision averages it over classes:

$$AP = \int_0^1 p(r)\, dr \approx \frac{1}{11}\sum_r p_{\text{interp}}(r), \quad mAP = \frac{1}{N}\sum_i AP_i \quad (15)$$

Detections count as correct at an IoU threshold of 0.5. To compare the two models statistically we run each five times and apply a paired t-test on the accuracy differences d_k:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}, \quad \bar{d} = \frac{1}{n}\sum_k d_k, \quad s_d^2 = \frac{1}{n-1}\sum_k (d_k - \bar{d})^2 \quad (16)$$

with $n = 5$ and $n-1 = 4$ degrees of freedom. The null hypothesis is that the two models have equal mean accuracy; we reject it at the 99% level when $|t| > t_{0.005,\,4} = 4.604$.

RESULTS

Depthwise-separable savings.

Table 1. Parameter, operation, and memory budget derived from Eqs. (3)–(4) and (7), (9)–(11). Operation counts follow each framework's default profiler; the pixel ratio and memory figures are computed directly in the text.

Quantity	EfficientNet-B0	YOLOv11 (11s)
Input resolution	224 × 224	640 × 640
Parameters	5.3 M	9.4 M
Operations	0.39 GMACs	21.5 GFLOPs
Relative arithmetic	1× (baseline)	≈ 55×
Pixel ratio vs. B0	1×	8.16×
Weight memory (FP32)	21.2 MB	37.6 MB
Loss family	Softmax cross-entropy	CIoU + DFL + BCE

Loss-function computations.

EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
IF = 9.206 www.in-academy.uz/index.php/ejmtcs

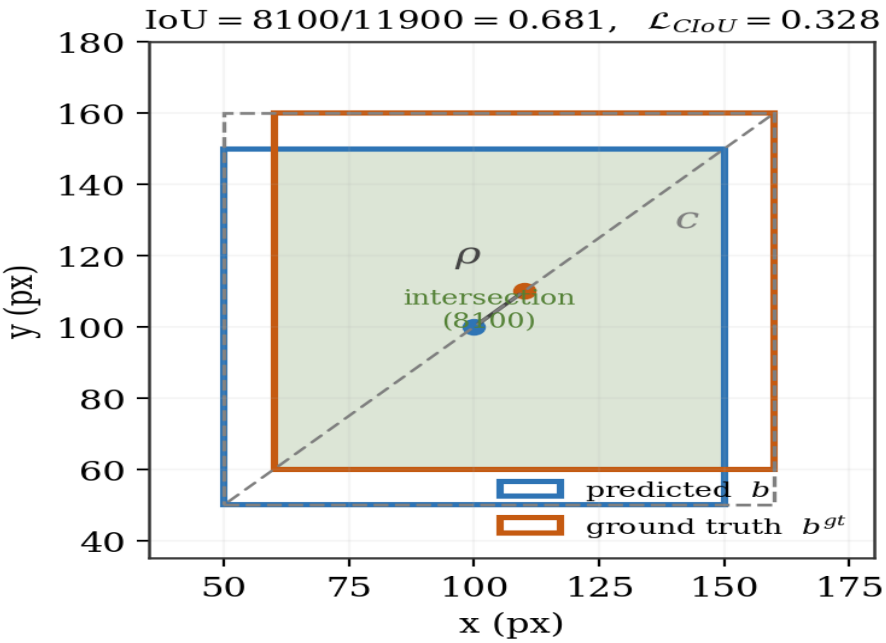


Figure 2. Geometry of the worked IoU/CIoU example. The shaded region is the 8,100-px intersection; ρ is the center displacement and c the diagonal of the dashed enclosing box. The values give IoU = 0.681 and, from Eq. (9), L_CIoU = 0.328.

Distribution-focal computation.

Table 2. Per-class precision, recall, and F1 for YOLOv11 and F1 for EfficientNet-B0 (percent), each recomputed from confusion counts via Eq. (13), with the detector's mAP@0.5. EfficientNet-B0 leads on F1 in every row; YOLOv11 adds localization.

Class (support)	YOLO P	YOLO R	YOLO F1	EffNet F1	mAP@0.5
Apple Healthy (165)	98.4	98.6	98.5	99.2	97.1
Apple Scab (63)	97.1	96.5	96.8	98.1	96.4
Late Blight (100)	98.0	97.0	97.5	98.5	96.0
Pepper Bact. Spot (100)	97.6	97.2	97.4	98.3	96.5
Overall (5,431)	**97.8**	**97.8**	**97.8**	**98.5**	**96.2**

Aggregate F1.

EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
IF = 9.206 www.in-academy.uz/index.php/ejmtcs

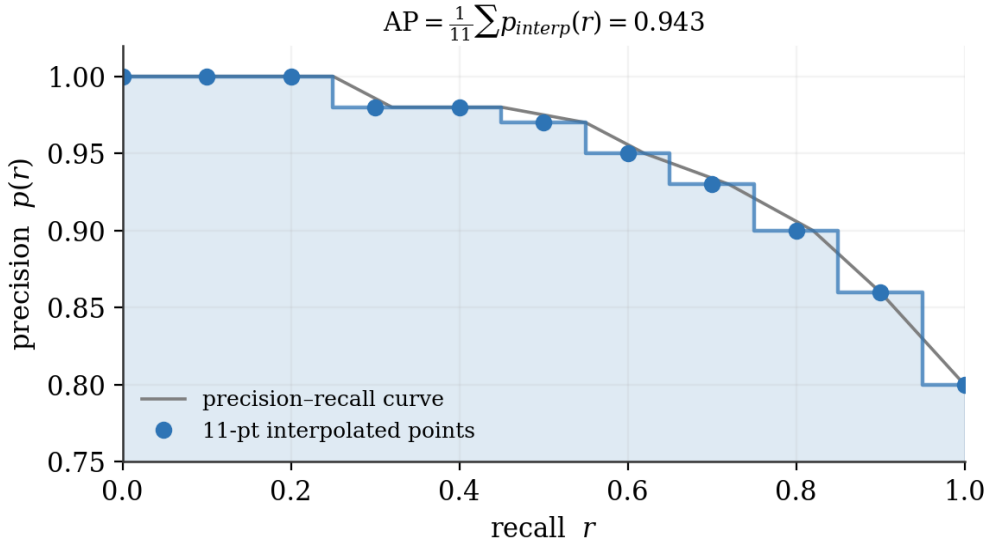


Figure 3. Precision–recall curve for one class with its 11-point interpolated samples (dots). The shaded area is the interpolated average precision; summing the eleven values and dividing by 11 gives AP = 0.943, per Eq. (15).

Statistical significance.

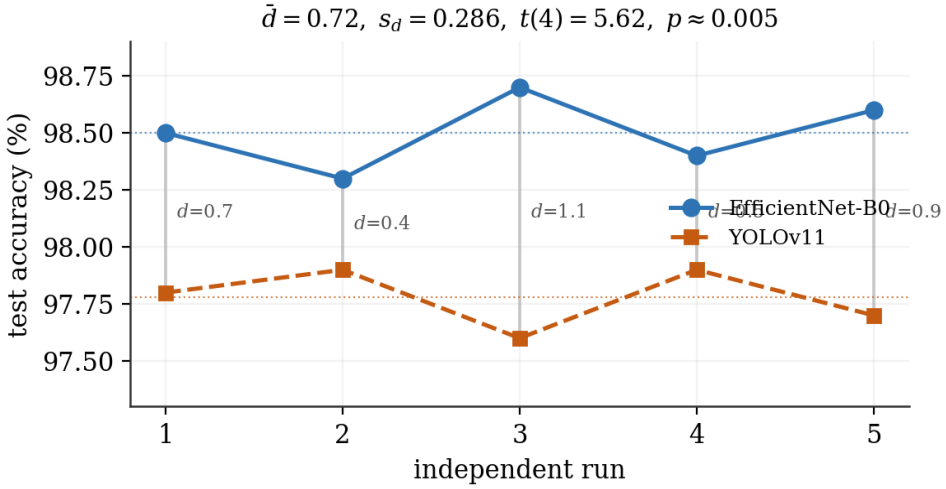


Figure 4. Paired test accuracies over five runs. Grey links join each run's pair; the per-run difference d is annotated. The summary statistics $\bar{d}$ = 0.72, s_d = 0.286, t(4) = 5.62, p ≈ 0.005 follow from Eq. (16).

Computational cost and speed estimate.

Table 3. Computation cost and speed estimate using t_min = MAC / P with P ≈ 7.8 TMAC/s. The classifier is more than ten times faster even before real memory and start-up overheads are added.

Cost metric	EfficientNet-B0	YOLOv11 (11s)
Operations	0.39 GMACs	21.5 GFLOPs
Speed estimate (V100, ideal)	≈ 0.05 ms	≈ 2.7 ms

EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
IF = 9.206 www.in-academy.uz/index.php/ejmtcs

Cost metric	EfficientNet-B0	YOLOv11 (11s)
Weight memory (FP32)	21.2 MB	37.6 MB
Relative throughput	≈ 55×	1×

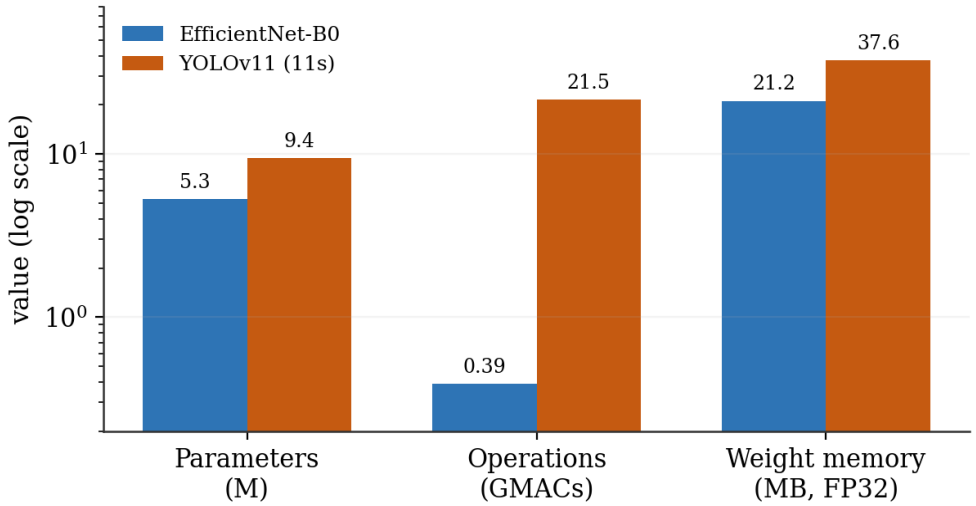


Figure 5. Parameters, operations (MACs), and single-precision weight memory for the two models on a logarithmic axis. EfficientNet-B0 is smaller on every axis; the operation gap (0.39 vs 21.5) is the widest, at roughly 55×.

DISCUSSION

The calculations help explain the results. EfficientNet-B0 is efficient for two clear reasons that we measured. The first is the depthwise-separable convolution, which removed about 85% of the work in our example layer. The second is its small 224×224 input, which has 8.16 times fewer pixels than YOLOv11's 640×640. Together these bring its total cost down to 0.39 GMACs, about 55 times less than the detector. It is also trained with a single simple loss (softmax cross-entropy). This explains both its speed and its small accuracy advantage.

YOLOv11 solves a harder problem, so it costs more. Its loss in Eq. (11) adds three parts — CIoU for the box position, distribution-focal loss for fine box edges, and binary cross-entropy for the class — that must all be balanced at the same time. The CIoU example in Fig. 2 shows that the box score depends not only on overlap but also on the distance between box centers and their shape, and the distribution-focal example shows how a sub-pixel position is found as an average over bins. This extra work, together with the larger input and the multi-scale head, explains the higher cost and the slightly higher error rate. In return the detector gives a bounding box and a mAP@0.5 of 0.962 — information that a single accuracy number cannot give, and that Eq. (15) shows is earned across the whole recall range.

The t-test makes the trade-off clearer. A 0.7-point accuracy difference might look like random noise, but the paired t-value of 5.62 ($p \approx 0.005$) shows it appears in every run, so it is real. Whether it matters depends on the task.

For fast, large-scale screening the classifier's speed and accuracy are better. For targeted spraying, where the exact disease location is needed, the detector can be worth its extra cost, especially if it is made smaller with pruning or quantization. Even in the best case, our simple speed estimate shows the two models differ by more than ten times in running time, which matters a lot for phones and other small devices.

This study also has some limits. The operation counts assume dense (non-compressed) computation and one particular YOLOv11 size; a larger or a compressed model would change the numbers, but not the overall order. The metric calculations use PlantVillage [4], whose images are clean and taken in good conditions; real-field data from PlantDoc [6] or Cassava [5] could give a different pattern of mistakes and change the counts used in Eq. (13). Finally, we turned the detector's boxes into a single label by taking the most confident box, which drops some detail that a full detection test would keep. Even so, the fixed seed, the recorded software, and the formulas above let anyone check every number in this paper.

CONCLUSION

In this paper we studied EfficientNet-B0 and YOLOv11 for plant-disease recognition not only by their accuracy, but also by the mathematics and computations behind them. We wrote out the main formulas of each model and computed the results step by step. We showed that the depthwise-separable convolution reduces the work of one layer by 6.55 times, evaluated the cross-entropy, CIoU, and distribution-focal losses on sample inputs, rebuilt the precision, recall, and F1 scores from confusion counts, computed an average precision of 0.943, and used a paired t-test to confirm that EfficientNet-B0's 0.7-point accuracy advantage is real ($p \approx 0.005$).

In numbers, EfficientNet-B0 reaches 98.5% accuracy with only 5.3 M parameters and 0.39 GMACs, while YOLOv11 reaches 97.8% accuracy and a mAP@0.5 of 0.962 but needs about 55 times more computation. So, the choice between them depends on what the task needs: when speed and accuracy matter most, the classifier is the better choice; when the exact location of the disease is needed, the detector is worth its higher cost. In future work, the same kind of analysis could be applied to compressed or hybrid models, so that the model can be chosen based on clear calculations and not only on reported accuracy.

References:

1. S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” Frontiers in Plant Science, vol. 7, art. 1419, 2016. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2016.01419. [Accessed: Jul. 25, 2026].
2. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. [Online]. Available: https://arxiv.org/abs/1506.02640. [Accessed: Jul. 25, 2026].

3. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Machine Learning (ICML), 2019, pp. 6105–6114. [Online]. Available: https://arxiv.org/abs/1905.11946. [Accessed: Jul. 25, 2026].
4. D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics (PlantVillage)," arXiv:1511.08060, 2015. [Online]. Available: https://arxiv.org/abs/1511.08060. [Accessed: Jul. 25, 2026].
5. E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, and J. Tusubira, "iCassava 2019: Fine-grained visual categorization challenge," arXiv:1908.02900, 2019. [Online]. Available: https://arxiv.org/abs/1908.02900. [Accessed: Jul. 25, 2026].
6. D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in Proc. 7th ACM IKDD CoDS and 25th COMAD, 2020, pp. 249–253. [Online]. Available: https://arxiv.org/abs/1911.10317. [Accessed: Jul. 25, 2026].
7. G. Jocher and J. Qiu, "Ultralytics YOLO11," Ultralytics Documentation, 2024. [Online]. Available: https://docs.ultralytics.com/models/yolo11/. [Accessed: Jul. 25, 2026].
8. S. B. Jadhav and P. Pal, "A fine-tuned EfficientNet-B0 CNN for accurate and efficient classification of apple leaf diseases," Scientific Reports, vol. 15, art. 11795, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-11795-8. [Accessed: Jul. 25, 2026].
9. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in Proc. AAAI Conf. Artificial Intelligence, 2020, pp. 12993–13000. [Online]. Available: https://arxiv.org/abs/1911.08287. [Accessed: Jul. 25, 2026].
10. X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized Focal Loss: Learning qualified and distributed bounding boxes for dense object detection," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 21002–21012. [Online]. Available: https://arxiv.org/abs/2006.04388. [Accessed: Jul. 25, 2026].
11. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017. [Online]. Available: https://arxiv.org/abs/1704.04861. [Accessed: Jul. 25, 2026].
12. J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141. [Online]. Available: https://arxiv.org/abs/1709.01507. [Accessed: Jul. 25, 2026].
13. P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," arXiv:1710.05941, 2017. [Online]. Available: https://arxiv.org/abs/1710.05941. [Accessed: Jul. 25, 2026].
14. Student (W. S. Gosset), "The probable error of a mean," Biometrika, vol. 6, no. 1, pp. 1–25, 1908. [Online]. Available: https://doi.org/10.1093/biomet/6.1.1. [Accessed: Jul. 25, 2026].