Vol.7 No.1 Tahun 2026
Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi
https://ejournal.unuja.ac.id/index.php/core
E-ISSN : 2774-7875 and P-ISSN : 2775-0124

Comparison of MobileNetV2, EfficientNet-B0, and ResNet50 for Fruit Freshness Classification Based on Accuracy and F1-Score

Nadiyah[1], Fathur Rizal[2], Andi Wijaya,[3], Zainal Arifin,[4]
[1,2,3,4] Universitas Nurul Jadid, Probolinggo, Indonesia

Article Info

Article history:

Received : mm dd, yyyy
Revised : mm dd, yyyy
Accepted : mm dd, yyyy

Keywords:

Architecture comparison;
Convolutional neural networks;
Fruit freshness classification;
Transfer learning;

Abstract

Freshness is a key determinant of fruit quality and safety, while manual assessment is subjective and slow. This study aims to compare three convolutional neural network architectures, namely MobileNetV2, EfficientNet-B0, and ResNet50, for the freshness classification of apples, bananas, and oranges in fresh and rotten conditions. The Food Freshness Dataset from Kaggle with 29,502 images was used and divided with a ratio of 70:20:10 into six classes. All three models were built using a transfer learning scheme with feature extraction on ImageNet pre-trained weights, an input size of 224×224, and an identical training configuration for 10 epochs. The main difference between the models lies in the specific preprocessing functions of each architecture to ensure a fair comparison. Evaluation was carried out on test data using accuracy and F1-score macro and weighted. The results show that ResNet50 achieved the highest performance with an accuracy of 0.9780 and a macro F1-score of 0.9787, followed by EfficientNet-B0 (0.9759; 0.9774) and MobileNetV2 (0.9726; 0.9732). Class-by-class analysis revealed that the Rotten Orange class was the most difficult for all models. EfficientNet-B0 and MobileNetV2 performed comparable to ResNet50 but with a much smaller number of parameters, making them more efficient for resource-constrained applications. This study emphasizes the importance of reporting F1-scores alongside accuracy on class-imbalanced data.

Corresponding Author:

Fathur Rizal,
fathurrizal@unuja.ac.id

INTRODUCTION

Fruit is a horticultural commodity that is susceptible to quality degradation due to its high water content and ongoing metabolic processes after harvest. Freshness is a key determinant of market value and a marker of food safety, so errors in assessing fresh and spoiled fruit directly impact economic losses and consumer health risks. In Indonesia, food loss and waste are significant, with the fruit and vegetable sector being the least efficient category in its supply chain (FAO, 2022). This situation emphasizes the need for a freshness assessment system that is fast, consistent, and does not rely entirely on manual inspection.

Manual freshness assessment relies on human visual observation, making it susceptible to fatigue, subjectivity, and speed limitations when dealing with large fruit volumes. Computer vision-based automation offers an alternative to standardize assessments and reduce labor costs during the sorting stage (Hossain et al., 2019). Convolutional neural networks (CNNs) have become a dominant approach in image classification due to their

DOI: https://doi.org/10.33650/coreai.v6i2

ability to learn hierarchical visual features directly from data without manual feature engineering (Krizhevsky et al., 2012).

Several previous studies have applied CNNs to fruit freshness classification with promising results. The application of MobileNetV2 to apple, banana, and orange images reported high validation accuracy for distinguishing fresh from rotten fruit (Singh et al., 2023). The CNN approach has also proven effective for freshness assessment of other food commodities, for example, Android-based egg freshness detection achieved high accuracy (Akbar et al., 2023), as well as the use of other machine learning algorithms such as K-Nearest Neighbor for data-driven classification tasks (Jasri et al., 2024). Other studies comparing several pretrained architectures have shown that architecture choice significantly impacts final performance, and no single architecture consistently outperforms all datasets (Ghosh & Singh, 2025; Iqbal et al., 2025; Palakodati et al., 2020). Lightweight models like MobileNetV2 are even capable of achieving competitive accuracy with low computational requirements (Shahi et al., 2022). These findings demonstrate that comparing architectures on similar datasets and experimental conditions is a crucial step before selecting a model for implementation. However, most previous research still focuses on accuracy metrics. Accuracy can be misleading when the class distribution is imbalanced, as a model favoring the majority class can still achieve high accuracy while failing to recognize the minority class (Sokolova & Lapalme, 2009). In the context of rotten fruit detection, failing to recognize the minority class is the most costly error, so an F1-score metric that balances precision and recall is necessary to complement accuracy.

Based on this description, this study compares three CNN architectures: MobileNetV2, EfficientNet-B0, and ResNet50, for the freshness classification of apples, bananas, and oranges. The three architectures were chosen because they represent distinct characteristics: MobileNetV2 as an efficient, lightweight model (Sandler et al., 2018), EfficientNet-B0 with balanced scaling between depth, width, and resolution (Tan & Le, 2019), and ResNet50 as a deeper residual model and a benchmark for image classification (He et al., 2016). The contribution of this research is twofold: it provides a comparison of three architectures with equivalent per-architecture preprocessing so that differences in results can be attributed to the architectures, and it includes macro and weighted F1-scores as a basis for evaluation in addition to accuracy so that performance under imbalanced class conditions can be assessed more precisely.

METHOD

This is an experimental study comparing the performance of three CNN architectures on the task of fruit freshness classification. The research stages included dataset collection, image preprocessing, model training using transfer learning, and evaluation using accuracy and F1-score. All experiments were run on the Google Collaboratory using the TensorFlow/Keras framework.

The dataset used is the Food Freshness Dataset, publicly available on Kaggle (ULNN Project, 2025). From the dataset, images of three types of fruit were selected: apples, bananas, and oranges, each in fresh and rotten condition, thus forming six classes: Fresh Apple, Fresh Banana, Fresh Orange, Rotten Apple, Rotten Banana, and Rotten Orange. The data was divided in a 70:20:10 ratio into 20,650 training images, 5,901 validation images, and 2,951 test images. The number of images between classes is unbalanced, with the Fresh Orange class having the most images, so the use of the F1-score as a companion metric is relevant. Details of the data division are presented in Table 1.

Table 1. Fruit Freshness Dataset Distribution

Subset	Ratio	Total Image	Percentage
Training	70	20.650	70%
Validation	20	5.901	20%
Testing	10	2.951	10%
Total	100	29.502	100%

All images were resized to 224x224 pixels to fit the standard input of all three architectures and maintain fairness in comparisons. A key aspect of this research was the use of specific preprocessing functions for each architecture. Each ImageNet pretrained architecture expects a different input normalization scheme, so each model's default preprocess_input function was applied instead of uniform scaling. Applying uniform preprocessing, such as dividing pixel values by 255, can disadvantage certain architectures and make comparisons unfair. Augmentation in the form of rotation, horizontal and vertical flips, and zooming was applied only to the training data to suppress overfitting, while the validation and test data were preprocessed without augmentation.

All three architectures were used as baselines with ImageNet pretrained weights (Deng et al., 2009) through a transfer learning scheme using feature extraction, where the base layers are frozen and only the classification layers are trained. This feature extraction approach is commonly used in deep convolutional architectures such as VGG (Simonyan & Zisserman, 2015). On top of each base, an identical classification head was added, consisting of global average pooling, a dense layer of 256 neurons with ReLU activation, a dropout of 0.5, and a dense output layer with softmax activation for the number of classes. The use of identical classification heads ensures that performance differences stem from the base architecture. All models were trained with the Adam optimizer (Kingma & Ba, 2015), a learning rate of 0.0001, a categorical crossentropy loss function, and 10 epochs. Configuration details are presented in Table 2.

Table 2. Model Training Configuration

Parameter	Value
Input image size	224 × 224 piksel
Initial boot (pretrained)	ImageNet
Transfer learning scheme	Feature extraction (basis dibekukan)
Optimizer	Adam
Learning rate	0,0001
Epoch	10
Loss function	Categorical crossentropy
Dropout	0,5
Output activation function	Softmax
Input preprocessing	preprocess_input per-architecture

Performance is evaluated on the test data using accuracy, precision, recall, and F1-score. F1-score is reported in both macro and weighted variants. The macro F1-score is used as the primary basis for ranking because it gives equal weight to each class and is more sensitive to minority classes. Equations (1) to (3) present the formulas for precision, recall, and F1-score.

$$Precision = TP / (TP + FP) \quad (1)$$

$$Recall = TP / (TP + FN) \quad (2)$$

$$F1\text{-}score = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (3)$$

DOI: https://doi.org/10.33650/coreai.v6i2

where TP (true positive), FP (false positive), and FN (false negative) respectively represent the number of correct positive predictions, false positive predictions, and false positive data that were not recognized. A confusion matrix is also used to analyze the classification error patterns of each model. In summary, all stages of the research, from dataset collection to comparison and analysis of results, are shown in Figure 1.

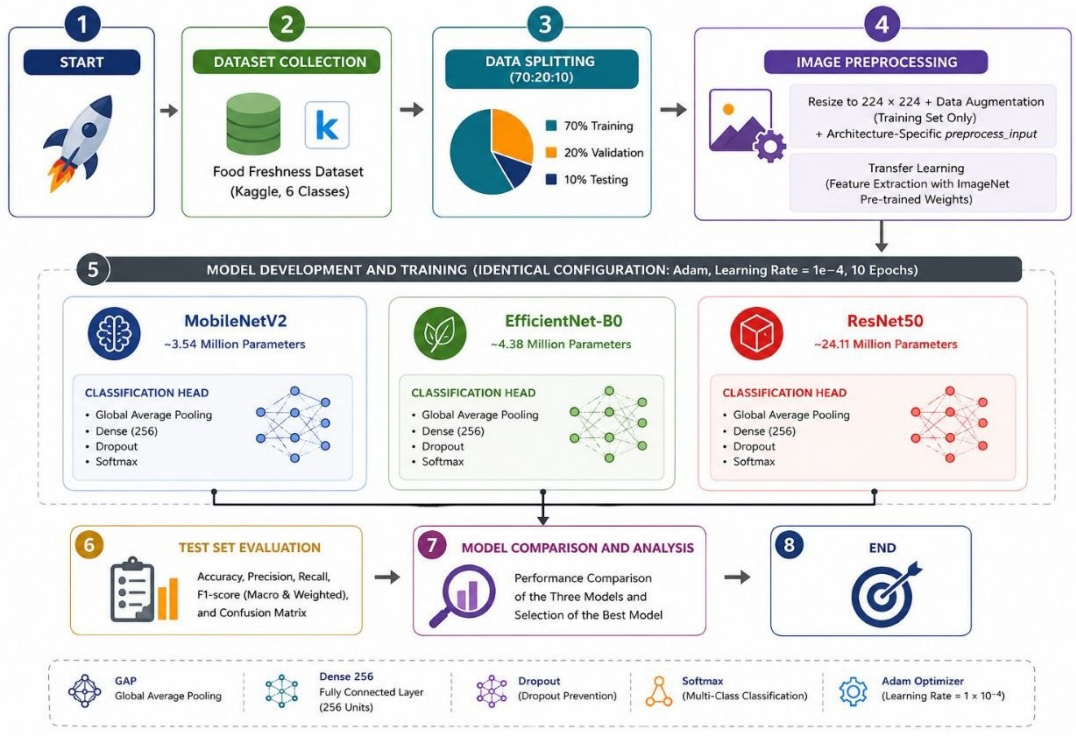

Figure 1. Research flow diagram

RESULTS AND ANALYSIS

All three models were trained for 10 epochs under identical configurations. In the first epoch, training accuracy for all three models ranged from 0.84-0.87 with a relatively high initial loss (0.38-0.47), which is expected since the classification head was just beginning to adapt to the features from the frozen base. Accuracy increased consistently over subsequent epochs and loss decreased without meaningful *overfitting*, as the gap between training and validation accuracy remained small throughout training. In the final epoch, MobileNetV2 achieved a training accuracy of 0.9692 with a training loss of 0.0750, and a validation accuracy of 0.9675 with a validation loss of 0.0674. EfficientNet-B0 achieved a training accuracy of 0.9732 with a training loss of 0.0607, and a validation accuracy of 0.9742 with a validation loss of 0.0570 - the lowest validation loss among the three models, indicating slightly better generalization. ResNet50, meanwhile, achieved the highest training accuracy (0.9760) with the lowest training loss (0.0540), although its validation loss (0.0581) was slightly higher than EfficientNet-B0's, suggesting a slightly stronger tendency to fit the training data, although still within a reasonable range.

In addition to accuracy, computation time during training was also recorded as a practical comparison dimension. Table 3 summarizes the average time per epoch under steady-state conditions, namely epochs 2 through 10; the first epoch was excluded from the calculation because it includes data loading and *caching* overhead that does not reflect the actual training speed. The results show that MobileNetV2 was the fastest, while ResNet50 required almost twice the time of MobileNetV2 per epoch, consistent with its much larger parameter count and network depth.

Table 3. Average Training Time per Epoch (Steady State, Epochs 2-10)

Model	Time per Epoch (seconds)	Time per Epoch (minutes)	Ratio to MobileNetV2
MobileNetV2	≈ 454	≈ 7.6	1.00×
EfficientNet-B0	≈ 548	≈ 9.1	1.21×
ResNet50	≈ 853	≈ 14.2	1.88×

Note: times were recorded on the Google Colaboratory environment and may vary depending on hardware allocation; the values above illustrate the relative comparison among models within the same training session, not a standardized benchmark.

Evaluation on the 2,951 test images serves as the primary basis for comparison, as this data was never seen during training. Table 4 summarizes accuracy, precision macro, recall macro, F1-score macro, F1-score weighted, and the number of parameters for each model. ResNet50 achieved the highest accuracy and macro F1-score (0.9780 and 0.9787), supported by the highest macro precision (0.9811), meaning its positive predictions were, on average, the least likely to be wrong. EfficientNet-B0, however, recorded the highest macro recall (0.9778), slightly surpassing ResNet50 (0.9766), meaning EfficientNet-B0 was, on average, the least likely to miss (fail to recognize) samples from their true class. This finding shows that ResNet50's advantage on aggregate metrics does not hold uniformly across every precision and recall component, and the choice of the "best model" may depend on which metric is prioritized for a given deployment context. MobileNetV2 consistently ranked third across all metrics, although the gap relative to the other two models was not large.

Table 4. Model Performance Comparison on the Test Set

Model	Accuracy	Precision macro	Recall macro	F1 macro	F1 weighted	Parameters (M)
MobileNetV2	0.9726	0.9748	0.9717	0.9732	0.9724	2.59
EfficientNet-B0	0.9759	0.9770	0.9778	0.9774	0.9760	4.38
ResNet50	0.9780	0.9811	0.9766	0.9787	0.9778	24.11

Figure 2 presents a comparison of accuracy and macro F1-score for the three models. For each model, the accuracy and macro F1-score values are very close, indicating balanced performance across classes despite the uneven data distribution.

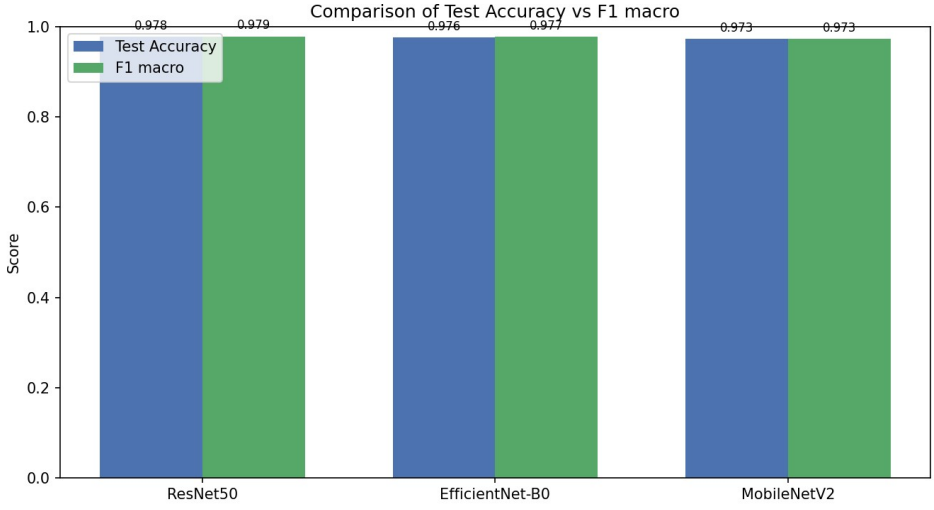

Figure 2. Comparison of accuracy and macro F1-score across the three models

Per-class analysis provides a far more detailed picture than aggregate metrics, since aggregate metrics can mask weaknesses in specific classes. Table 5 presents the F1-score for

DOI: https://doi.org/10.33650/coreai.v6i2

each class across the three models, along with the number of test samples (*support*). For classes with clearly distinguishable images, such as Fresh Banana and Rotten Banana, all three models achieved F1-scores close to or equal to 1.00, indicating that these classes are easily distinguished based on color and surface texture. The Fresh Apple and Rotten Apple classes also show high F1-scores (above 0.98) across all three models. In contrast, the Rotten Orange class consistently proved to be the most difficult class for all models, with F1-scores ranging only from 0.906-0.916, far below the other classes. The Fresh Orange class, despite having the largest number of samples (1,052 images), also showed relatively lower F1-scores (0.967-0.973) compared to the other non-orange classes.

Table 5. F1-score per Class on the Test Set

Class	Support	MobileNetV2	EfficientNet-B0	ResNet50
Fresh Apple	344	0.9841	0.9957	0.9956
Fresh Banana	340	0.9985	1.0000	0.9985
Fresh Orange	1052	0.9674	0.9681	0.9726
Rotten Apple	479	0.9844	0.9937	0.9896
Rotten Banana	383	0.9987	1.0000	1.0000
Rotten Orange	353	0.9059	0.9068	0.9159

Figure 3 shows the *confusion matrix* for the three models. The most prominent error pattern occurs between the Fresh Orange and Rotten Orange classes: for MobileNetV2, 38 out of 353 Rotten Orange images (10.8%) were misclassified as Fresh Orange; for EfficientNet-B0, 31 images (8.8%); and for ResNet50, 33 images (9.3%). The reverse error, Fresh Orange images misclassified as Rotten Orange, occurred far less often (25 out of 1,052 for MobileNetV2, 34 for EfficientNet-B0, and 21 for ResNet50). In other words, the errors are asymmetric: all three models more frequently fail to recognize truly rotten oranges as rotten than they mistakenly flag fresh oranges as rotten. The visual similarity between fresh oranges and oranges in the early stages of spoilage, combined with the relatively small number of Rotten Orange training images, is the most likely explanation for this difficulty.

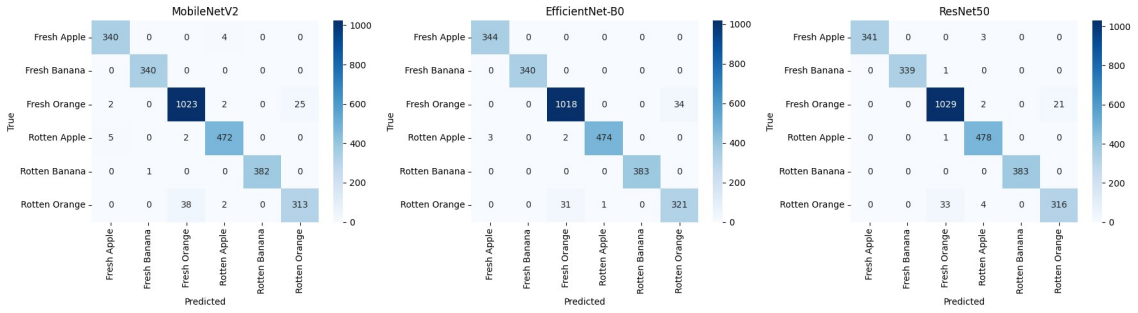

Figure 3. Confusion matrix for MobileNetV2, EfficientNet-B0, and ResNet50 on the test set

To better understand the nature of the errors in the Rotten Orange class, Table 6 separates the precision and recall values for that class. Across all three models, recall is consistently lower than precision, indicating that the errors are dominated by *false negatives* (Rotten Orange images that failed to be recognized) rather than *false positives* (images from other classes mistakenly predicted as Rotten Orange). EfficientNet-B0 has the highest recall (0.9093), although its precision is slightly lower than the other two models, while ResNet50 has the highest precision (0.9377) but lower recall than EfficientNet-B0.

Table 6. Precision and Recall for the Rotten Orange Class (Support = 353)

Model	Precision	Recall	Misclassified as Fresh Orange
MobileNetV2	0.9260	0.8867	38 (10.8%)
EfficientNet-B0	0.9042	0.9093	31 (8.8%)
ResNet50	0.9377	0.8952	33 (9.3%)

Overall, the experimental results show that all three architectures can classify fruit freshness very well, with small differences in performance at the aggregate level. ResNet50 has a slight edge in accuracy and macro F1-score, but this advantage comes with a much larger number of parameters, approximately 24.11 million, compared to EfficientNet-B0 (4.38 million) and MobileNetV2 (2.59 million), as well as a training time per epoch nearly twice that of MobileNetV2 (Table 3). ResNet50 therefore requires the greatest computational resources, both in terms of parameter count and time, for a relatively small performance gain.

EfficientNet-B0 offers an interesting balance: its performance is only slightly below ResNet50 in accuracy and macro F1-score, and it even leads in macro recall, while having less than a fifth of the parameters and a much shorter training time per epoch. MobileNetV2, as the lightest and fastest model, remains competitive with a macro F1-score gap of only about 0.005 relative to ResNet50. This finding is consistent with previous research indicating that no single architecture is always superior in absolute terms, and model selection should consider the balance between accuracy and efficiency (Ghosh & Singh, 2025; Iqbal et al., 2025; Palakodati et al., 2020). The per-class analysis and the precision-recall breakdown for the Rotten Orange class carry an important practical implication. Because the errors in this class are dominated by low recall, meaning rotten fruit images predicted as fresh, the risk that arises in real-world deployment is that fruit which is actually unfit may pass through the sorting process undetected. This risk is practically more damaging than the reverse error, fresh fruit mistakenly flagged as rotten, because the former can affect food safety while the latter only affects sorting efficiency. This finding underscores that reporting F1-score and per-class precision-recall analysis, not aggregate accuracy alone, is essential for deployment cases involving food safety, and opens opportunities for further research such as class weighting or decision threshold adjustment specifically for the Rotten Orange class to reduce the false negative rate.

CONCLUSIONS

This study compared three CNN architectures, namely MobileNetV2, EfficientNet-B0, and ResNet50, for the freshness classification of apple, banana, and orange images using accuracy, precision, recall, and F1-score. On the test set, ResNet50 achieved the highest accuracy and macro F1-score (0.9780 and 0.9787) as well as the highest macro precision (0.9811), but EfficientNet-B0 led in macro recall (0.9778), showing that the "best model" can differ depending on which metric is prioritized. ResNet50's advantage also came with a much higher computational cost, approximately 24.11 million parameters and a training time per epoch nearly twice that of MobileNetV2, making EfficientNet-B0 and MobileNetV2 more efficient for deployment on resource-constrained devices without sacrificing much accuracy. Per-class analysis and the precision-recall breakdown show that the Rotten Orange class was the most difficult for all models, with F1-scores ranging only from 0.906-0.916. Errors in this class were dominated by low recall, with 31-38 out of 353 rotten orange images (8.8-10.8%) mistakenly recognized as fresh oranges, while the reverse error was far less common. This asymmetry matters in practice because it means the system is more likely to let genuinely unfit fruit pass through than to flag fresh fruit as rotten, so reporting accuracy alone, without

F1-score and per-class recall analysis, risks masking weaknesses that are directly relevant to food safety. Future research could explore *fine-tuning* of the base layers, addressing class imbalance through *class weighting* or decision threshold adjustment, adding data for the weaker classes, and testing the models on *mobile* devices to directly assess computational efficiency under real deployment conditions.

REFERENCES

Akbar, R. R. M., Rizal, F., & Shudiq, W. J. (2023). Implementasi algoritma Convolutional Neural Network (CNN) untuk deteksi kesegaran telur berbasis Android [Implementation of the Convolutional Neural Network (CNN) algorithm for Android-based egg freshness detection]. Jusikom: Jurnal Sistem Komputer Musirawas, 8(1), 1–10. https://doi.org/10.32767/jusikom.v8i1.1949

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Fadhlurrahman, M. F., Wirayuda, T. A. B., & Purnama, B. (2024). Prediction of fruit freshness level using convolutional neural network. Proceedings of the 12th International Conference on Information and Communication Technology (ICoICT), 137–143. https://doi.org/10.1109/ICoICT61617.2024.10698538

FAO. (2022). FAO and MoA start the Food Loss Study in Indonesia. Food and Agriculture Organization. https://www.fao.org/indonesia/news/detail/FAO-and-MoA-Start-the-Food-Loss-Study-in-Indonesia/en

Ghosh, D., & Singh, J. P. (2025). A hybrid CNN-LSTM model for accurate fruit freshness classification using deep learning. Discover Artificial Intelligence. https://doi.org/10.1007/s44163-025-00750-7

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90

Hossain, M. S., Al-Hammadi, M., & Muhammad, G. (2019). Automatic fruit classification using deep learning for industrial applications. IEEE Transactions on Industrial Informatics, 15(2), 1027–1034. https://doi.org/10.1109/TII.2018.2875149

Iqbal, M., et al. (2025). Canned apple fruit freshness detection using hybrid convolutional neural network and transfer learning. Journal of Food Quality, 2025. https://doi.org/10.1155/jfq/8522400

Jasri, M., Rahmadan, I., & Shudiq, W. J. (2024). Increasing student interest in learning through the implementation of the K-Nearest Neighbor algorithm in classifying learning preferences at SMAN 1 Kraksaan. Journal of Computer Networks, Architecture and High Performance Computing, 6(4). https://doi.org/10.47709/cnahpc.v6i4.4526

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105.

Palakodati, S. S. S., Chirra, V. R. R., Dasari, Y., & Bulla, S. (2020). Fresh and rotten fruits classification using CNN and transfer learning. Revue d'Intelligence Artificielle, 34(5), 617–622. https://doi.org/10.18280/ria.340512

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Shahi, T. B., Sitaula, C., Neupane, A., & Guo, W. (2022). Fruit classification using attention-based MobileNetV2 for industrial applications. PLoS ONE, 17(2), e0264586. https://doi.org/10.1371/journal.pone.0264586

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR), 1–14. https://arxiv.org/abs/1409.1556

Singh, A., Gupta, R., & Kumar, A. (2023). Fresh and rotten fruit detection using deep CNN and MobileNetV2. In A. Mishra, D. Gupta, & G. Chetty (Eds.), Advances in IoT and Security with Computational Intelligence (ICAISA 2023), Lecture Notes in Networks and Systems (Vol. 755, pp. 219–231). Springer. https://doi.org/10.1007/978-981-99-5085-0_22

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning (ICML), 97, 6105–6114. https://arxiv.org/abs/1905.11946

ULNN Project. (2025). Food Freshness Dataset. Kaggle. https://www.kaggle.com/datasets/ulnnproject/food-freshness-dataset