Journal of Electrical Engineering and Computer (JEECOM)
Vol. 7, No. 2 (2025), DOI: 10.33650/jeecom.v4i2
p-ISSN: 2715-0410; e-ISSN: 2715-6427

Improving Tomato Ripeness Classification Using Knowledge Distillation and Hyperparameter Optimization with Optuna

Iasya Sholihin [1], Andi Sunyoto [2]
[1, 2] Magister Informatika, Universitas Amikom Yogyakarta, Sleman, Indonesia

Article Info

Article history:

Diterima 5 Juli 2025
Revisi 14 Juli 2025
Diterbitkan 15 September 2025

Keywords:

Tomato Ripeness Classification
Knowledge Distillation
Optuna
Hyperparameter Tuning
ResNet50
MobileNetV2

ABSTRACT

Automatic classification of tomato ripeness plays a crucial role in ensuring post-harvest quality and efficiency in the horticultural industry. This study proposes a combined strategy of Knowledge Distillation (KD) and hyperparameter optimization using Optuna to improve the accuracy of the ResNet50 student model by leveraging the performance of a MobileNetV2 teacher model.We used a publicly available Kaggle dataset containing 8,540 images, categorized into four ripeness levels (green, red, ripe, and rotten), comprising 7,157 training images and 1,383 validation images.Each image was resized to 224×224 pixels; light augmentation techniques (random rotation, brightness–contrast adjustment, flipping, and Gaussian blur) were applied only to the training set to prevent overfitting while maintaining consistency during evaluation.The MobileNetV2 teacher model was initially fine-tuned on the last 20 layers using manual hyperparameters (freeze_until = 20, dropout = 0.6), achieving an accuracy of 85.8%.Subsequent tuning via Optuna identified the optimal configuration (freeze_until = 91, dropout_rate = 0.5055), which improved the teacher's performance to 89.6%.The resulting teacher model was then used to distill knowledge into the ResNet50 student: under manual settings, the student's accuracy improved from 55.24% to 73.25%; when the student model was also optimized using Optuna, its accuracy surged to 85.54% nearly matching the teacher.Further evaluation using a confusion matrix and ROC curves revealed an increase in per-class AUC to the range of 0.91–0.99 in the KD + Optuna student model, confirming that this method effectively closes the performance gap between student and teacher.These findings demonstrate that combining KD with Optuna-based hyperparameter optimization is an effective approach for producing a lightweight, fast, and highly accurate tomato ripeness classification model ready for deployment in field applications to support post-harvest decision-making.

Corresponding Author:
Iasya Shilihin
Universitas Amikom Yogyakarta,
Jl. Ring Road Utara, Ngringin, Condongcatur, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281
Email: iasyasholihin@students.amikom.ac.id

1. INTRODUCTION

The classification of tomato (Solanum lycopersicum) ripeness is a fundamental aspect of the horticultural industry, closely tied to the quality and economic value of the product.As one of the most popular agricultural commodities, tomatoes play a vital role in the human diet due to their rich nutritional content, particularly lycopene, which is known for its significant health benefits [1].Tomato ripeness greatly influences taste, texture, and post-harvest durability [2].Currently, the classification of ripeness is still largely performed manually by farmers who rely on visual color perception, a method that is often subjective and prone to inconsistencies [2].This presents a significant challenge in maintaining product quality throughout the supply chain.

With the advancement of technology, automated approaches utilizing computer vision and deep learning algorithms have begun to be applied to improve the accuracy of tomato ripeness classification **[3].**Convolutional Neural Network (CNN) models such as ResNet50 and MobileNetV2 have demonstrated remarkable effectiveness in image classification due to their ability to extract visual features such as color and texture [4].However, the use of models like ResNet50 and MobileNetV2 often demands high computational resources and is prone to overfitting, especially under varying visual conditions and limited datasets. To address this issue, a Knowledge Distillation (KD) approach has been proposed, which enables the transfer of knowledge from a more powerful model (teacher) to a simpler model (student), allowing for good classification accuracy with reduced computational complexity.

This study focuses on the use of the same dataset employed in a previous study by Hetharua et al., which explored the automation of tomato ripeness classification. In this research, more advanced image augmentation techniques were also applied such as contrast enhancement and the implementation of Gaussian blur to improve model robustness against lighting variations and background noise, thereby addressing limitations identified in the previous study [5].Experimental results showed that applying the KD method successfully increased the student model's accuracy from 57% to 72% using a dataset consisting of 8,540 tomato images.Although the student model did not reach the teacher model's accuracy level of 86.84%, the results demonstrate the effectiveness of the proposed approach in improving classification quality.

By leveraging modern technologies such as deep learning and computer vision, this study not only has the potential to enhance the efficiency of tomato ripeness classification but also contributes to the sustainability and productivity of the agricultural sector as a whole [5].The successful implementation of this method is expected to assist farmers in better harvest management and strengthen their competitiveness in increasingly demanding markets [6].Accordingly, the objective of this study is to develop and evaluate a deep learning-based tomato ripeness classification model that incorporates Knowledge Distillation (KD) and Optuna hyperparameter optimization, aiming to improve model accuracy while maintaining computational efficiency. Furthermore, this research provides a foundation for future developments in automated fruit classification using advanced technologies, contributing to innovation in the agricultural industry and food security.

2. METHOD

This section outlines the complete methodological workflow applied in this study.First, tomato image data were acquired from the public Kaggle repository and then split into a training set (7,157 images) and a validation set (1,383 images).All images were resized to 224×224 pixels and normalized, while only the training set underwent light augmentation techniques including random rotation, brightness contrast adjustment, flipping, and blurring to enhance sample diversity.The next step involved training the teacher model, MobileNetV2, which was fine-tuned on its last 20 layers. This was followed by hyperparameter optimization, targeting parameters such as learning rate, early stopping, number of unfrozen layers, and dropout rate, in order to identify the optimal configuration.The results from this tuning process were then used in the Knowledge Distillation phase, where the student model, ResNet50, was trained to mimic the soft targets generated by the teacher.

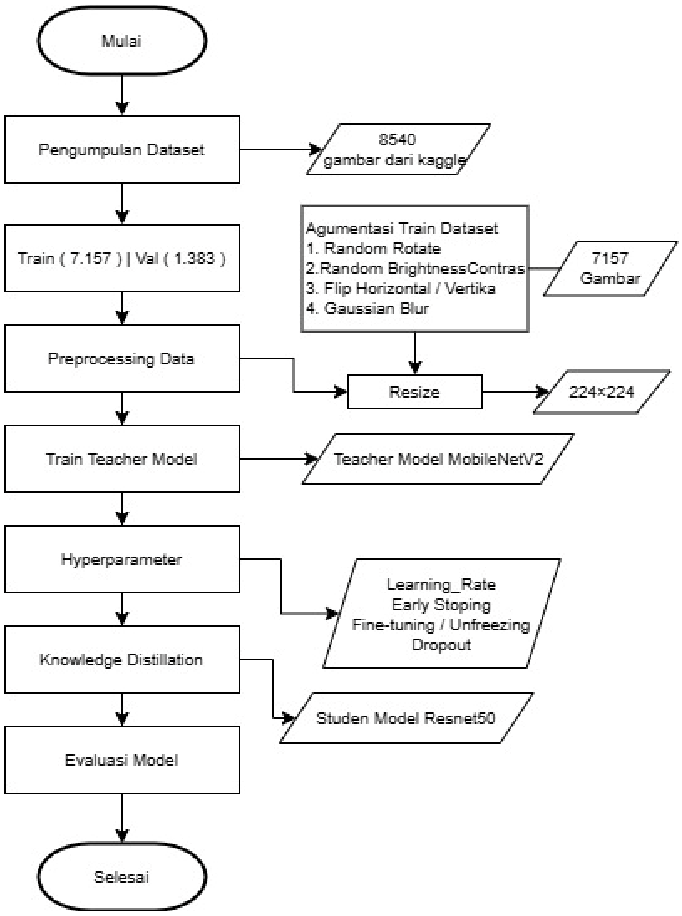
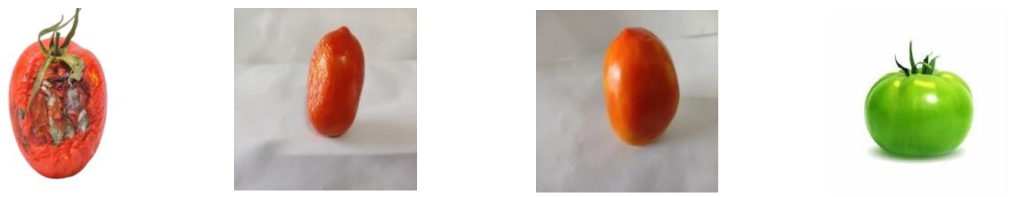


Figure 1: Research Stages

2.1. Dataset

This study utilized a dataset obtained from Kaggle, consisting of 8,540 tomato images categorized into four classes: overripe, rotten, red, and green tomatoes [7].The training set included 2,197 images of overripe tomatoes, 1,044 of rotten tomatoes, 2,173 of red tomatoes, and 1,742 of green tomatoes. Meanwhile, the validation set consisted of 427 images of overripe tomatoes, 202 of rotten tomatoes,418 of red tomatoes, and 336 of green tomatoes.:



(a) Rotten Tomato (b) Overripe Tomato (c) Ripe Red Tomato (d) Unrip Green Tomato

Figure 2. Tomato Fruit Dataset

2.2. Preprocessing and Augmentasi

Each image in the dataset was first resized to a resolution of 224×224 pixels, which is the standard input size for both ResNet50 and MobileNetV2 architectures.Subsequently, various light augmentation techniques were applied to the training set to enrich sample diversity and prevent overfitting.The augmentation transformations included: Random Rotation (rotating images by random multiples of 90°) [8], [9], Random Brightness and Contrast, Random Horizontal/Vertical Flip, and Gaussian Blur [10], [11].It is important to note that all augmentation techniques were applied exclusively to the training set, while the validation set underwent only resizing and normalization without any augmentation, in order to evaluate model performance on "untouched" images [12].Once preprocessing and augmentation were completed, the resulting data batches were then fed into the teacher model (MobileNetV2) for initial training.

2.3. Traning Teacher Model

The teacher model was trained using MobileNetV2 pretrained on ImageNet, where only the last 20 layers were fine-tuned while the earlier layers were frozen.A classification head was added on top of the feature extractor, consisting of a GlobalAveragePooling2D layer, two Dense layers (512 and 256 units with ReLU activation and 0.6 dropout), and a softmax output layer with four classes to reduce the risk of overfitting [13].The model was compiled using the SGD optimizer (learning rate 1×10^{-4}, momentum 0.9), with categorical cross-entropy as the loss function and accuracy as the evaluation metric.Training was conducted for 40 epochs on the lightly augmented training set (rotation, brightness–contrast adjustment, flipping, blurring), with callbacks including EarlyStopping (patience = 5), ReduceLROnPlateau (patience = 3), ModelCheckpoint, and a cosine annealing learning rate schedule.The validation set, which underwent only resizing, was used to monitor validation performance.After training, the best-performing weights were saved for use in the subsequent Knowledge Distillation process.This process aims to transfer knowledge from the teacher model to a smaller student model, which typically has lower performance, in order to produce a lightweight model with high accuracy [14].

2.4. Hyperparameter Tuning

Following preprocessing and dataset splitting, we applied Bayesian optimization (Optuna) to search for the best hyperparameter configuration for the ResNet50 model [15], [16].The search space included a learning rate ranging from 1×10^{-4} to 1×10^{-2}, dropout rates between 0.1 and 0.5, and the number of units in the first fully connected layer ranging from 16 to 64.Each combination was tested through a short training session of 5 epochs and evaluated based on validation accuracy.Once the optimal configuration was identified, the model was retrained for a longer duration of 40 epochs.

2.5. Knowledge Distillation and Studen Training

In the Knowledge Distillation stage, the student model—based on ResNet50 with the last 91 layers fine-tuned was equipped with a classification head identical to that of the teacher and trained using a combination of hard loss (categorical cross-entropy) and soft loss (KL divergence between the teacher's and student's soft targets at temperature $T = 3$), with a loss weight $\alpha = 0.5$.The training was conducted for 40 epochs using the Adam optimizer (learning rate 1×10^{-4}), along with callbacks including EarlyStopping (patience = 5), ReduceLROnPlateau (patience = 3), a cosine annealing scheduler, and ModelCheckpoint [17], [18].The training process utilized the lightly augmented training set, while validation was performed exclusively on resized data.The best-performing student weights were then restored for final evaluation.

2.6. Model Evaluation

The model was evaluated using a confusion matrix to gain deeper insights into misclassification patterns.Each element in the confusion matrix indicates the number of correct or incorrect predictions for each class, providing a clear overview of where the model most frequently makes classification errors [19].The use of a confusion matrix is particularly important in the context of machine learning, as it enables detailed analysis of the model's performance on a per-class basis, thereby facilitating the identification of classes that are often misclassified.

3. RESULTS AND DISCUSSION

In this experiment, the researchers utilized a personal computer equipped with an Intel Core i7 12th-generation processor, 64 GB DDR4 RAM, and Windows 11 Pro, along with Google Colab Pro as a supporting platform. The conducted experiments yielded several significant findings related to tomato ripeness image classification using a CNN-based approach, with MobileNetV2 serving as the teacher model and ResNet50 as the student model.

3.1. Scenario performed

In the various scenarios conducted, as shown in Table 1, the first experiment was performed using the original dataset obtained from the public Kaggle repository. The second experiment was carried out using a preprocessed dataset enhanced through augmentation. This study focuses on improving the classification performance of ResNet50 by applying the Knowledge Distillation (KD) method in combination with Optuna-based hyperparameter optimization.

Table 1. Research Scenario

Category	Teacher	Studen	Hyperparameter	Dataset
Baseline	MobileNetV2	ResNet50	Manual	Original
Augmetasi	MobileNetV2	ResNet50	Manual	Augmentasi
Tuned Teacher	MobileNetV2	-	Optuna	Augmentasi
KD + Tunde Studen	MobileNetV2	ResNet50	Manual	Augmentasi
KD + Tunde Studen	MobileNetV2	ResNet50	Optuna	Augmentasi

3.2. Hyperparameter Search

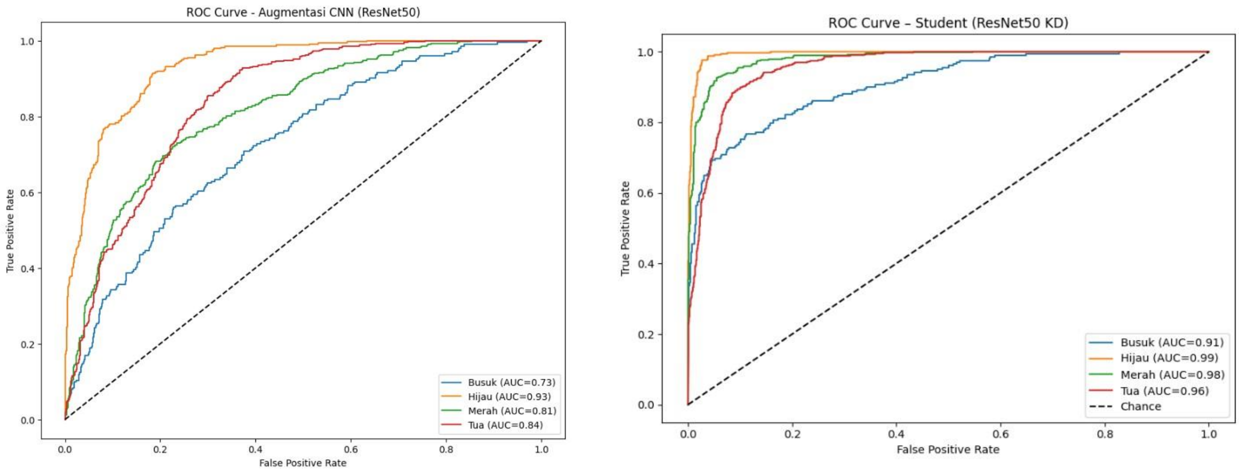
During the hyperparameter tuning stage, Bayesian Optimization (Optuna) was employed to identify the optimal values for two key parameters in the ResNet50 model—namely, the number of layers to unfreeze (freeze_until), within the range of [10, 100], and the dropout_rate, within the range of [0.2, 0.7].Each trial involved a short training run of 5 epochs and was evaluated based on validation accuracy.Out of 30 trials, the highest validation accuracy of 1.0 was achieved in the first trial, with the best configuration being: freeze_until = 91 and dropout_rate = 0.5055.These parameters were then used for the final training phase prior to the Knowledge Distillation process.

Table 2. Optuna hyperparameter search

Iter	Val Accuracy	Freeze_until	Dropout_rate
0	1.000	91	0.5055114881
1	0.9992	11	0.5797830371
2	1.000	12	0.5959558928
3	0.9992	70	0.6209434752
4	1.000	54	0.3115274355
5	0.9992	76	0.2009426668
6	1.000	37	0.5688968754
7	1.000	46	0.2847101267

3.3. Model Performance

After the application of Knowledge Distillation, a performance comparison between the "old" ResNet50 model (without KD) and the "new" ResNet50 model (with KD and hyperparameter tuning) revealed a sign improvement.The original ResNet50 model without distillation achieved an accuracy of approximately 57%, whereas the KD-enhanced ResNet50 model reached 73% accuracy on the validation set. Figure 3 (left) displays the ROC curve for the baseline ResNet50 model: the curve is relatively closer to the diagonal, reflecting lower average AUC values (approximately 0.75–0.85 per class). In contrast, Figure 4 (right) presents the ROC curve for the KD-based ResNet50, which bends sharply toward the upper-left corner, with class-wise AUC values increasing to a range of 0.90–0.99 indicating the improved ability of the new model to distinguish between the four classes.



(a) Original ResNet50

(b) Improved ResNet50

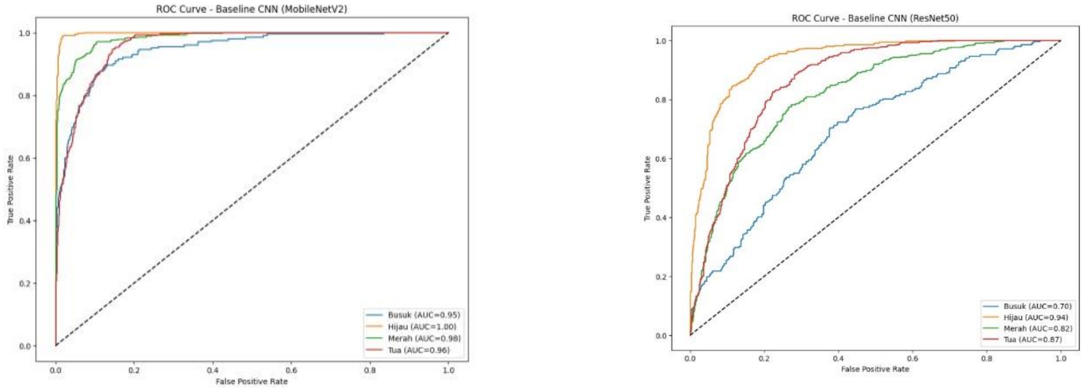
Figure 3. Comparison Result Chart

The improvement can be primarily attributed to two key factors:

1. **Soft-target distillation** teaches the student model to mimic the probability distribution of the teacher model, enabling the student to capture "soft knowledge" such as inter-class relationships that cannot be learned from hard labels alone.
2. **Hyperparameter tuning** (freeze_until, dropout_rate) via Optuna ensures that the student model operates under an optimal configuration, thereby improving its generalization performance.

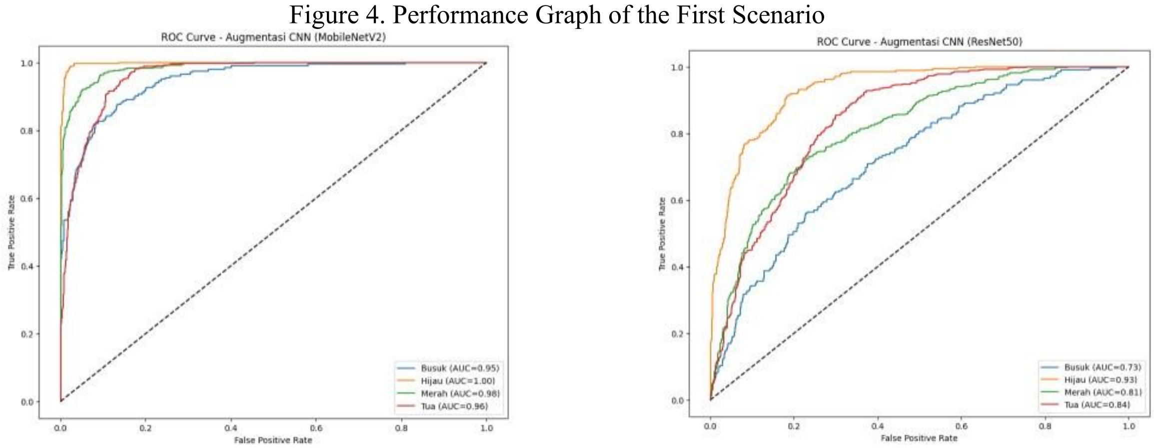
3.4. Results

To evaluate the effectiveness of augmentation, hyperparameter optimization, and Knowledge Distillation (KD) on the ResNet50 architecture, we designed five distinct experimental scenarios. The first scenario ("Baseline Original") utilized a MobileNetV2 teacher and a ResNet50 student built with manually defined hyperparameters (freeze_until = 20, dropout_rate = 0.6) on the original dataset without augmentation; MobileNetV2 achieved an accuracy of 85.76%, while ResNet50 only reached 55.24%. In the second scenario ("Baseline Augmentation"), a light augmentation pipeline (rotation, brightness–contrast, flipping, blurring) was applied to the training set, but the student's hyperparameters remained manual; this improved MobileNetV2's performance to 86.26%, but caused ResNet50's accuracy to drop to 53.87%. For the third scenario ("Tuned Teacher"), MobileNetV2 was re-trained on the augmented dataset using Optuna-optimized hyperparameters targeting val_accuracy = 1.000 and yielding freeze_until = 91 and dropout_rate = 0.5055 resulting in an accuracy of 89.58%.The fourth scenario ("KD Manual") employed this tuned and augmented teacher model to distill knowledge into a ResNet50 student with manual hyperparameters (freeze_until = 20, dropout_rate = 0.6), boosting the student's accuracy from 53.87% to 73.25%. Finally, in the fifth scenario ("KD + Tuned Student"), the ResNet50 student adopted the same Optuna-optimized hyperparameters (freeze_until = 91, dropout_rate = 0.5055) during the KD process, achieving a final accuracy of 85.54% nearly matching the performance of the tuned teacher.These five scenarios confirm that the combination of augmentation, automated hyperparameter tuning, and knowledge distillation can dramatically improve the classification accuracy of the ResNet50 student model, significantly narrowing the gap with its teacher counterpart.



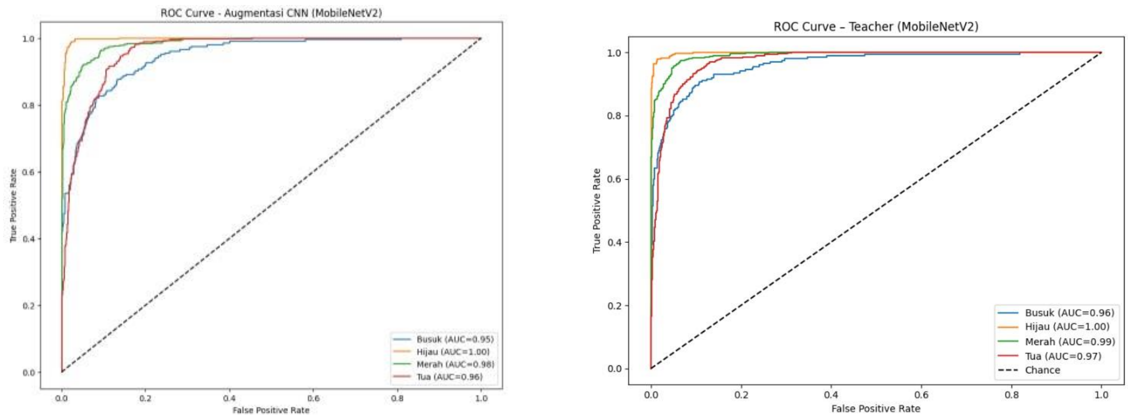
(a) MobileNetV2 (85.76 %) (b) ResNet50 (55.24 %)

Figure 4. Performance Graph of the First Scenario



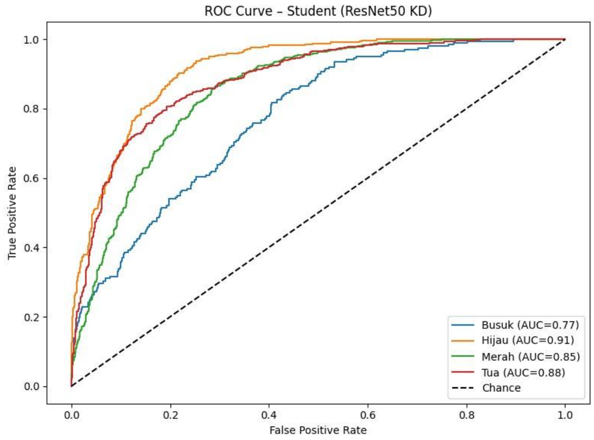
(a) MobileNetV2 (86.26 %) (b) Resnet50 (53.87 %)

Figure 5. Result Graph of the Second Scenario

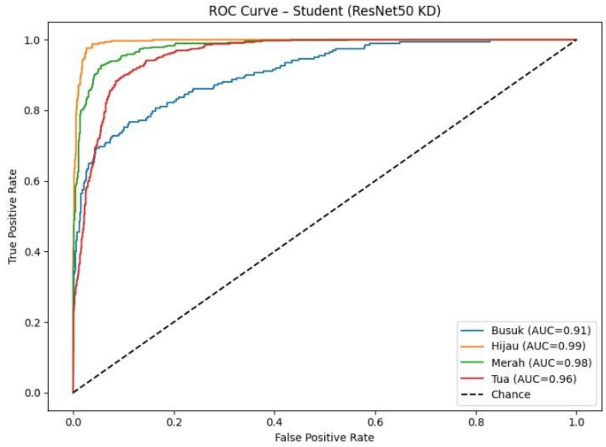


(a) MobileNetV2 Manual (86.26 %) (b) MobileNetV2 Optuna (89.58 %)

Figure 6. Result Graph of the Third Scenario



(a) Resnet50 KD Manual (73.25 %)



(b) Resnet50 KD Optuna (85.54 %)

Figure 7. Result Graphs of the Fourth and Fifth Scenarios

4. CONCLUSION

Based on the experiments conducted, this study demonstrates that the combination of light augmentation, hyperparameter optimization using Bayesian Optimization (Optuna), and Knowledge Distillation (KD) significantly enhances the performance of ResNet50 in classifying the ripeness of large tomatoes**.**Without augmentation and using manual settings, ResNet50 achieved only 55.24% accuracy, and its performance did not improve even after augmentation.By tuning the MobileNetV2 teacher model using Optuna (freeze_until = 91, dropout_rate = 0.5055), its accuracy increased to 89.58%.Applying KD to the ResNet50 student with manual hyperparameters boosted its accuracy to 73.25% (+17.38%), and further, KD with Optuna-optimized hyperparameters pushed the student's accuracy to 85.54% effectively closing the performance gap between student and teacher.

Preprocessing steps including resizing, normalization, and light augmentation proved to be crucial for enriching data variation and preventing overfitting. Automated hyperparameter tuning ensured that the model operated under optimal configurations, while Knowledge Distillation (KD) enabled the student model to capture the "soft knowledge" from the teacher—knowledge that cannot be obtained through hard labeling alone.Moving forward, this research can be expanded by enlarging or balancing the dataset, applying more advanced augmentation techniques, or testing lighter student architectures for real-time applications in the field.It is expected that this method can assist agricultural practitioners and the tomato processing industry in performing automatic quality classification, thereby reducing losses due to spoilage and increasing processing efficiency.

UCAPAN TERIMAKASIH

On behalf of the authors, we would like to express our sincere gratitude to our advisors, Dr. Andi Sunyoto, M.Kom., M. Hanafi, M.Eng., Ph.D., and Tonny Hidayat, S.Kom., M.Kom., Ph.D., for their invaluable guidance throughout this research**.**Their insights, suggestions, and encouragement have been instrumental in supporting us during the course of this study.

Journal of Electrical Engineering and Computer (JEECOM)
Vol. 7, No. 2 (2025), DOI: 10.33650/jeecom.v4i2
p-ISSN: 2715-0410; e-ISSN: 2715-6427

REFERENCES

[1] Md. Y. Ali *et al.*, “Nutritional Composition and Bioactive Compounds in Tomatoes and Their Impact on Human Health and Disease: A Review,” *Foods*, vol. 10, no. 1, p. 45, 2020, doi: 10.3390/foods10010045.

[2] W. Zita, S. Bressoud, G. Glauser, F. Kessler, and V. Shanmugabalaji, “Chromoplast Plastoglobules Recruit the Carotenoid Biosynthetic Pathway and Contribute to Carotenoid Accumulation During Tomato Fruit Maturation,” 2022, doi: 10.1101/2022.09.14.507955.

[3] A. F. Powell *et al.*, “A <i>Solanum Lycopersicoides</I> Reference Genome Facilitates Insights Into Tomato Specialized Metabolism and Immunity,” *The Plant Journal*, vol. 110, no. 6, pp. 1791–1810, 2022, doi: 10.1111/tpj.15770.

[4] L. Izzo, L. Castaldo, S. Lombardi, A. Gaspari, M. Grosso, and A. Ritieni, “Bioaccessibility and Antioxidant Capacity of Bioactive Compounds From Various Typologies of Canned Tomatoes,” *Front Nutr*, vol. 9, 2022, doi: 10.3389/fnut.2022.849163.

[5] A. F. Powell *et al.*, “A <i>Solanum Lycopersicoides</I> Reference Genome Facilitates Insights Into Tomato Specialized Metabolism and Immunity,” *The Plant Journal*, vol. 110, no. 6, pp. 1791–1810, 2022, doi: 10.1111/tpj.15770.

[6] R. Braglia *et al.*, “Phytochemicals and Quality Level of Food Plants Grown in an Aquaponics System,” *J Sci Food Agric*, vol. 102, no. 2, pp. 844–850, 2021, doi: 10.1002/jsfa.11420.

[7] L. Da Quach, K. N. Quoc, A. N. Quynh, N. Thai-Nghe, and T. G. Nguyen, “Explainable Deep Learning Models With Gradient-Weighted Class Activation Mapping for Smart Agriculture,” *IEEE Access*, vol. 11, pp. 83752–83762, 2023, doi: 10.1109/ACCESS.2023.3296792.

[8] K. R. Resmi, G. Raju, V. Padmanabha, and J. Mani, “Person Identification by Models Trained Using Left and Right Ear Images Independently,” pp. 281–288, 2023, doi: 10.2991/978-94-6463-110-4_20.

[9] A. Mohamed *et al.*, “The Impact of Data Processing and Ensemble on Breast Cancer Detection Using Deep Learning,” *Journal of Computing and Communication*, vol. 1, no. 1, pp. 27–37, 2022, doi: 10.21608/jocc.2022.218453.

[10] J. Chu and S. Khan, “Transfer Learning and Data Augmentation in Osteosarcoma Cancer Detection,” *J Emerg Invest*, 2023, doi: 10.59720/22-285.

[11] P. R. Togatorop, Y. Pratama, A. M. Sianturi, M. S. Pasaribu, and P. S. Sinaga, “Image Preprocessing and Hyperparameter Optimization on Pretrained Model MobileNetV2 in White Blood Cell Image Classification,” *Iaes International Journal of Artificial Intelligence (Ij-Ai)*, vol. 12, no. 3, p. 1210, 2023, doi: 10.11591/ijai.v12.i3.pp1210-1223.

[12] W. Gouda, N. U. Sama, G. Al-Waakid, M. Humayun, and N. Z. Jhanjhi, “Detection of Skin Cancer Based on Skin Lesion Images Using Deep Learning,” *Healthcare*, vol. 10, no. 7, p. 1183, 2022, doi: 10.3390/healthcare10071183.

[13] C. C. Tseng, V. Lim, and R. W. Jyung, “Use of Artificial Intelligence for the Diagnosis of Cholesteatoma,” *Laryngoscope Investig Otolaryngol*, vol. 8, no. 1, pp. 201–211, 2023, doi: 10.1002/lio2.1008.

[14] A. Banerjee, O. C. Mutlu, A. Kline, S. Surabhi, P. Washington, and D. P. Wall, “Training and Profiling a Pediatric Facial Expression Classifier for Children on Mobile Devices: Machine Learning Study,” *JMIR Form Res*, vol. 7, p. e39917, 2023, doi: 10.2196/39917.

[15] M. Hassanali, M. Soltanaghaei, T. J. Gandomani, and F. Z. Boroujeni, “Software Development Effort Estimation Using Boosting Algorithms and Automatic Tuning of Hyperparameters With Optuna,” *Journal of Software Evolution and Process*, vol. 36, no. 9, 2024, doi: 10.1002/smr.2665.

[16] H. Almarzooq and U. b. Waheed, “Automating Hyperparameter Optimization in Geophysics With Optuna: A Comparative Study,” *Geophys Prospect*, vol. 72, no. 5, pp. 1778–1788, 2024, doi: 10.1111/1365-2478.13484.

[17] M. Goldblum, L. Fowl, S. Feizi, and T. Goldstein, “Adversarially Robust Distillation,” *Proceedings of the Aaai Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3996–4003, 2020, doi: 10.1609/aaai.v34i04.5816.

[18] Y. Tian, D. Krishnan, and P. Isola, “Contrastive Representation Distillation,” 2019, doi: 10.48550/arxiv.1910.10699.

[19] F. H. Garabaghi, S. Benzer, and R. Benzer, “Sequential GP-UCB Bayesian Optimization for Deep Neural Network Fine-Tuning in Dissolved Oxygen Prediction,” 2024, doi: 10.21203/rs.3.rs-3930680/v1.