

Multi-Class Real-Time Color Classification of Coffee Beans via Fine-Tuned EfficientNetB0 and Post-Training Quantization

Siti Yuliyanti*¹, Syamsul Maarip²

^{1,2}Departement of Informatics, Faculty of Enginnering, Siliwangi University, Indonesia

Email: sitiyuliyanti@email.ac.id

Received : Oct 28, 2025; Revised : Dec 2, 2025; Accepted : Dec 2, 2025; Published : Feb 15, 2026

Abstract

The first problem faced in coffee bean classification is that the manual grading or sorting process still relies heavily on human labor, making it subjective, time-consuming, and prone to errors. Secondly, existing deep learning-based systems often require substantial computing resources, rendering them inefficient for industrial-scale implementation or on limited hardware. The research objective is to develop an efficient, lightweight, and accurate automatic classification model to recognize coffee bean color and support the automation of quality control processes in the coffee post-harvest chain. This study develops an automated system for coffee bean classification based on four color classes: light, medium, green, and dark, utilizing the lightweight EfficientNet model with fine-tuning of smaller versions of EfficientNet (B0–B3). The research stages consist of dataset acquisition, pre-processing, modeling and fine-tuning, as well as model evaluation on the detection system on low-end devices. The main innovation of this research is the efficiency and speed of real-time classification of coffee bean color images using a lightweight CNN model optimized through quantization, which supports field applications with hardware limitations without sacrificing accuracy. Fine-tuning EfficientNetB0 by unfreezing the last 30 layers achieved 97.17% training accuracy and 99.25% validation accuracy with consistent loss reduction, supported by Test-Time Augmentation (TTA) which improves prediction stability to >80% confidence against variations in field image quality. Deployment to TensorFlow Lite (TFLite) with 8-bit quantization resulted in a lighter model that maintained 99.50% accuracy and accelerated inference by up to 6x compared to the original H5 model, and excelled at multi-object detection without sacrificing classification confidence.

Keywords : *Coffee beans, Device low-end, EfficientNetB0, Lightweight, Quantization*

This work is an open access article licensed under a Creative Commons Attribution 4.0 International License.



1. INTRODUCTION

Coffee bean quality assessment during the post-harvest stage is heavily influenced by bean color, as it reflects the ripeness and roasting level, which impact flavor and selling price[1], [2]. Quantitative color measurements have been utilized in laboratory studies to correlate color patterns with the degree of roasting and product quality[3], [4]. In recent years, computer vision and deep learning techniques have demonstrated high performance in coffee bean quality classification and inspection tasks, including defect detection [5], [6]distinguishing between ripe and unripe beans, and determining roasting levels. Various pre-trained CNN architectures (e.g., DenseNet, MobileNet, Inception, ResNet, and EfficientNet) have been compared and proven capable of achieving high accuracy on diverse coffee datasets[7], [8], [9]. However, many studies focus on binary classification (good vs. defective) or specific defect detection, while multi-class classification based on color variations (e.g., light, medium, green, dark) which is crucial for commercial grading, has not received consistent attention[10], [11].

With the need to implement these solutions in industrial or device-constrained environments (e.g., real-time sorting systems, embedded cameras in small factories), there is a growing demand for lightweight yet accurate models[12], [13]. Lightweight approaches using compressed architectures,

efficient architectures (e.g., smaller versions of EfficientNet), or techniques like depthwise separable convolution and knowledge distillation have successfully balanced the accuracy and computational efficiency trade-offs in coffee bean inspection tasks[14], [15]. Previous studies have reported lightweight architectures that achieve competitive results while being easier to deploy on resource-constrained devices[16], [17].

EfficientNet is a family of CNN architectures that simultaneously scales the network (depth, width, resolution) to achieve a good accuracy/computation ratio; several studies on coffee and roasting datasets have reported high performance of EfficientNet variants, making this architecture promising for coffee bean color recognition tasks in multi-class scenarios[18], [19], [20]. However, the literature specifically evaluating the lightweight EfficientNet for classification of four color classes (light, medium, green, dark) with attention to color preprocessing, augmentation for varying lighting, and efficiency metrics (latency, model size, FLOPs) is still relatively limited[12], [21], [22].

Alternative quantization methods that can improve real-time accuracy on the EfficientNetB0 model include Quantization-Aware Training (QAT), which simulates the effects of quantization during training with fake quantization nodes, allowing the model to adapt to low precision, such as INT8 or 4-bit, without significant loss in fast inference for coffee bean classification[23], [24], [25]. This approach is superior to Post-Training Quantization (PTQ) because it maintains accuracy up to 68.39% on ResNet-18, which is similar to EfficientNet on ImageNet using only 10% of the training data, and can be applied to EfficientNetB0 for low-latency edge devices[22], [26].

The main objective of this research is to optimize a real-time color-based coffee classification model using the Lightweight EfficientNet model and quantization techniques to reduce model size and accelerate the inference process without significantly sacrificing accuracy. This will provide an efficient and practical automated solution for the coffee bean classification process, which can improve quality and productivity in the coffee industry.

2. METHOD

The research method in Figure 1 shows the stages starting from collecting the dataset, exploring the dataset, configuring the training model using augmentation and normalization data, base model EfficientNetB0, transfer learning vs fine-tuning EfficientNetB0, deployment on low-end devices with quantization, and finally evaluation model consisting of an H5 model as the training model for coffee bean color detection. and a TFLite model for deployment on mobile devices.

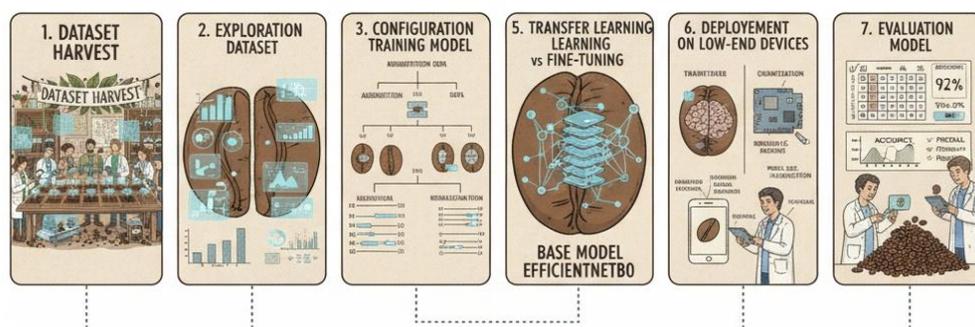


Figure 1. Research Method

2.1. Dataset

The Coffee Bean Dataset comprises a variety of data regarding coffee beans sourced globally, illustrated in Figure 2. This dataset encompasses details related to the beans' origin, classification, and taste characteristics, as well as insights into their cultivation conditions and preparation methods. The

images are collected and stored in PNG format without manual intervention. For this research, the dataset consists of 1,600 images of coffee beans, divided into 1,200 for training and 400 for testing and validation purposes.



Figure 2. Sample dataset coffee

2.2. Exploration dataset

Data exploration in this study is a crucial initial step to understand, summarize, and visualize the main characteristics of the image dataset used, thereby understanding the structure and content of the dataset that will be utilized in image classification. This research data exploration involves data division and the allocation of data for training, testing, and validation purposes, with a total of 4 classes: 'Dark', 'Green', 'Light', and 'Medium'. Thus, ensuring data balance, which is a good thing, to avoid bias in model training.

2.3. Configuration Training Model

Model training configuration: this study consists of a set of settings and parameters that determine how the model will be trained. The initial configuration of the model is seed = 42, which aims to ensure Reproducibility, which is very important for debugging and comparing experimental results. Next, the image will be resized to 224 x 224 pixels to standardize the size [27], [28], and then use DataGenerator to automate the calling of training data, test data, and validation. batch size configuration = 16 is the number of images processed in one forward pass during training or evaluation, where small batches require less RAM / GPU but training can be slower while large batches require faster but require large memory.

2.4. Transfer learning and Fine-Tune in EfficientNetB0

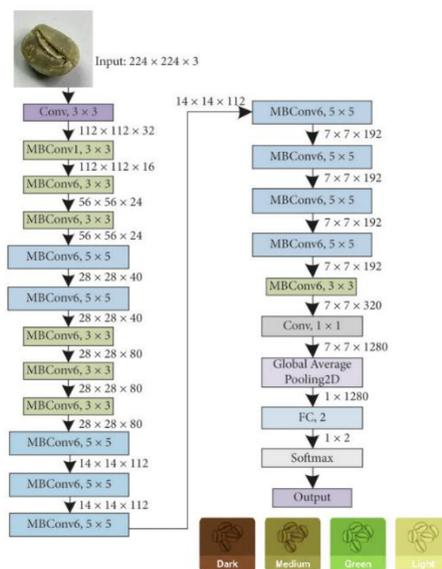


Figure 3. EfficientNet-B0 architecture

EfficientNet-B0 is the foundation upon which the entire EfficientNet family is built and is the smallest and most efficient model within the EfficientNet family[29], [30]. Discovered through Neural Architectural Search (NAS), EfficientNet-B0 is the baseline model, and its architecture is shown in Figure 3. The main components of the architecture are the MBConv (Cellular Inverse Bottleneck Convolution) block and the squeeze and excitation optimization[16], [31].

Figure 4. Shows an illustration of coffee bean quality detection based on dark, light, medium, green color classes, where the final results of the model detection produce comparative results of class prediction analysis, confidence difference, speedup on the H5 and TFlite models.

The transfer learning stage in the EfficientNetB0 baseline model aims to utilize pretrained weights trained on a large dataset to accelerate and improve training performance on a new target dataset for coffee bean classification[32], [33]. Load Pretrained Base Model without Top Layer: The EfficientNetB0 model is used as a feature extractor with pretrained weights, typically with the `include_top=False` parameter to omit the final classification layer specific to the ImageNet dataset. This model already understands common and complex basic visual features[2], [34], [35]. Freeze Base Model: The base model weights are initially frozen (not retrained) to maintain their basic features and prevent them from being corrupted by the potentially smaller target data, accelerating training and avoiding initial overfitting. Then, a Custom Classifier is added on top of the base model, adding fully connected layers (e.g., a dense layer with the appropriate number of neurons for the target class) to map the basic features to class predictions specific to the new dataset.

EfficientNet employs a method known as compound coefficient to enhance models in a straightforward yet impactful way. Rather than arbitrarily adjusting width, depth, or resolution, compound scaling consistently modifies each aspect using a specific set of scaling factors. By utilizing this scaling approach along with AutoML, the creators of EfficientNet produced seven models of different sizes, which exceeded the top accuracy of many convolutional neural networks while also being significantly more efficient.

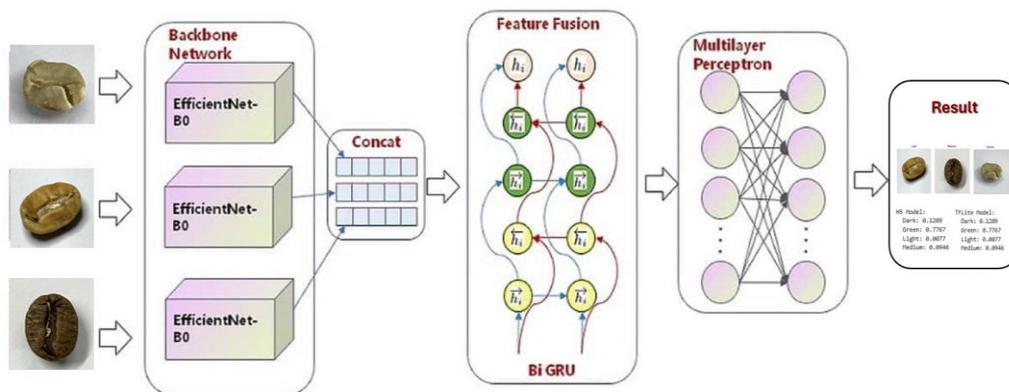


Figure 4. Illustration of coffee bean quality detection

This research coffee bean prediction is based on color, with EfficientNet as the main model, TTA (Test-Time Augmentation) to improve accuracy[36], with output: coffee color class (light, medium, green, dark) + confidence + probability table of all classes based on parameter configuration. This modeling performs TTA (Test-Time Augmentation), which is a technique to improve prediction accuracy by creating several augmented versions of the input image and then averaging the predictions. So there is modeling based on the initial dataset, which becomes the model baseline, and modeling using the augmented dataset.

2.5. Deploy on low-end devices with quantization

The technical implementation of the Lightweight EfficientNetB0 model with quantization on low-end devices in this study began with creating a baseline model (`base_model = EfficientNetB0`

(include_top=False, weights='imagenet', input_shape = (*img_size, 3), pooling='avg') using the EfficientNetB0 architecture without a top classification layer. It used pretrained weights from ImageNet and adjusted the input image size with the parameter img_size = 224 x 224. Global average pooling was applied to convert the convolution feature output into a 1-dimensional vector[22].

Then, set the Trainable Layer (Fine-tuning). If fine_tune=True, all layers except the last unfreeze_last are frozen (trainable = False), meaning this layer is not retrained, and the last unfreeze_last layer is allowed to be trained so the model can adapt to the new dataset. If fine_tune=False, the entire base model is frozen; only the top layer will be trained. Next, create an input layer to define the input layer so that the input size matches the image data. Layers are constructed on top of the base model by adding BatchNormalization and Dropout (0.5) for regularization and stabilization of the training. These layers function as classifiers specific to the coffee dataset, improving the ability to map features to classes. The final Dense layer, with a size of num_classes (e.g., 4 classes), uses an output layer with softmax activation for multi-class classification to fully model the input and output. Finally, the return model and base model function returns the complete model and the base model for monitoring or other operations.[7]

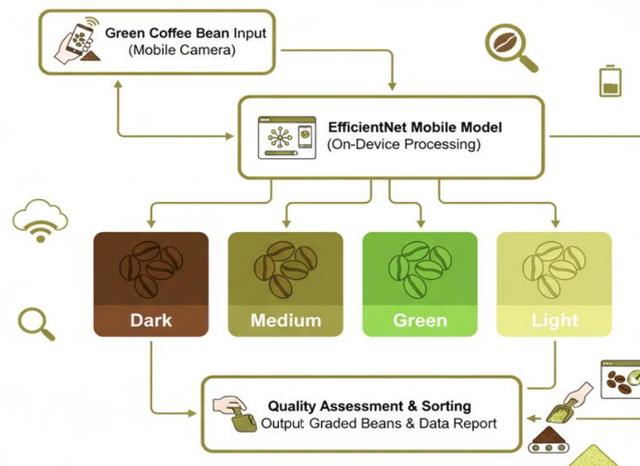


Figure 5. Illustration of coffee bean classification based on 4 color classes on a low-end device

2.6. Evaluation model

The final results of the model detection produce comparative results of class prediction analysis, confidence difference, and speedup on the H5 and TFLite models. The model performance evaluation in this study compares whether the H5 model (the native Keras/TensorFlow format) and TFLite (the converted model for deployment on edge/mobile devices) can produce consistent class predictions on the same test data, to ensure that the conversion process does not alter the model's predictive performance/accuracy. The use of confidence values indicates the model's level of confidence in a particular class prediction. This is to determine quantization in TFLite; however, confidence scores may shift slightly compared to the H5 model. Comparative analysis is necessary to maintain model integrity and systematically understand any changes. The confidence value in a classification model prediction is generally derived from the softmax function on the output layer of the neural network, which generates the probability of each class. The most common formula for confidence is equation 1.

$$\text{confidence} = \max(\text{softmax}(z)) \tag{1}$$

where z is the vector of logit scores resulting from the last layer, and the softmax function defines the probability of each class with equation 2.

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2)$$

The confidence value for prediction is the probability of the class with the highest value from the softmax result. Speedup comparisons are conducted to measure the contribution of TFLite model optimization or compression in the context of inference on edge/mobile devices[28], [37], [38]. TFLite is generally optimized for faster performance and a smaller RAM footprint. Comparing inference times between H5 and TFLite demonstrates the practical advantages of model conversion to TFLite, especially for real-time applications on limited devices.

3. RESULT

In this section, the results of the research show that the coffee bean image classification model was developed using EfficientNetB0 as the backbone (feature extractor) and several additional layers. The initial backbone lacks the top classification layer (pure features only), and the output is given batch normalization and dropout (50%) for regularization. The dense layers are: Dense 512 units + LeakyReLU + BatchNorm + Dropout 40% and Dense 128 units + LeakyReLU + BatchNorm + Dropout 30%.

3.1. Transfer Learning in Baseline Model EfficientNetB0

The base model training phase with TensorFlow Keras is used to pre-train the model by freezing (untraining) the base part of the model and only training its top layers. This function compiles the model using the Adam optimizer and categorical cross-entropy loss with label smoothing, which helps improve training stability and reduce overfitting[39], [40]. The model is then trained using training and validation data with epochs 1 to 20, learning rate = 0.00005, smoothing = 0.1, with the training results stored in a history object for further analysis. This approach is commonly used to adapt a pretrained model before proceeding to the full training phase to allow the model to gradually adapt to new data. Fine-tuning on the EfficientNetB0 transfer learning model in this study was carried out through the unfreeze_for_finetune stage using the last 30 layers that were unfrozen, namely looping from the beginning to the last 5 parts of the train model was frozen and the loop on the last 30 layers was trained with a smaller learning rate of 0.00005, aiming for the model to update the weights slowly so as not to lose the knowledge that has been learned from the pretrained model.

Table 1. Base Model EfficientNetB0 (epoch = 20, learning rate =0.00005, smoothing =0.1)

Epoch	Accuracy	Training Loss	Accuracy Validation	Validation Loss
1	83.63	0.7805	85.00	0.6775
5	87.99	0.6239	96.25	0.4634
10	92.38	0.5534	96.50	0.4624
15	94.86	0.5039	98.00	0.4308
20	94.57	0.5204	98.25	0.4191

The accuracy results of the EfficientNet B0 base model are shown in Table 1, where there is an increase in accuracy and a decrease in loss values on both data sets as the number of epochs increases, indicating that the model is increasingly able to recognize data patterns well and has improved performance from epoch 1 to epoch 20. At epoch 20, validation accuracy reached 98.25% and validation loss decreased to 0.4191, indicating that the model has undergone effective training without any signs of significant overfitting.

3.2. Fine-tuning Model EfficientNetB0

Fine-tuning results for training showed that the training accuracy initially started at around 88.6% and gradually increased to over 97% in the final epochs. The loss values generally decreased, indicating the model was improving at learning the training data. Meanwhile, testing and validation results showed accuracy starting at around 97.75%, with validation loss values being quite low and relatively stable, indicating the model's good generalization.

The graph in the image shows two phases: Phase 1, initial training from epochs 0 to 14, and Phase 2, after fine-tuning from epochs 14 to 35. Phase 1 shows that the training accuracy (blue) and validation accuracy (orange) both show an increasing trend. This means the model is learning well from the data in the accuracy graph, while the loss graph shows that the training loss (blue) and validation loss (orange) both show a decreasing trend. This confirms that the model is getting better at making predictions and its error rate is decreasing. This green line indicates the start of the fine-tuning stage, where the layers before the green line are frozen and during fine-tuning are thawed to participate in training with the aim of fine-tuning the model parameters more finely to be more specific to the dataset used.

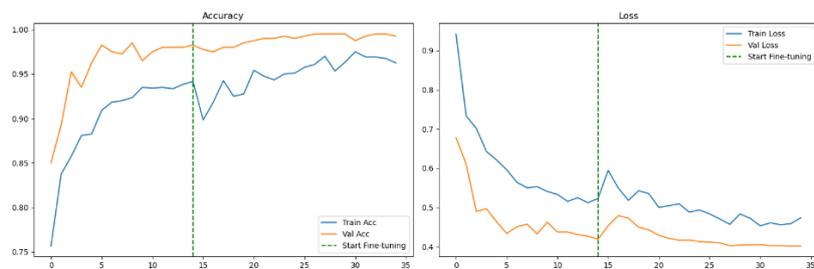


Figure 6. Comparison accuracy with train loss and validation loss in fine-tuning EfficientNetB0

After fine-tuning began, the accuracy graph shows that both training (blue) and validation (orange) accuracy continued to improve, even at a faster rate for a while. Validation accuracy (orange) peaked at around 0.98-0.99 (98-99%), which is an excellent result. Meanwhile, the loss graph shows a drastic decrease in both training and validation losses right after fine-tuning began. This indicates that the fine-tuning strategy is very effective in reducing model error. Validation loss (orange) reaches its lowest point during this phase, indicating peak model performance. The results of fine tuning EfficientNETB0 with epoch = 20, learning rate = 0.00005, smoothing = 0.1, with unfreeze_last = 30 are shown in Table 2.

It can be seen that the training accuracy increased from 88.63% in the first epoch to 97.17% in the 20th epoch, while the loss value decreased from 0.6198 to 0.4626, indicating that the model is getting better at recognizing training data patterns. Validation accuracy also showed a consistent increase from 97.75% to 99.25% with validation loss decreasing from 0.4534 to 0.4021, indicating that this fine-tuning result effectively improved the model's performance without significant overfitting. Thus, unlocking and retraining the last 30 layers helped the model achieve higher accuracy on both the training and validation sets.

Table 2. Result Fine tuning EfficienetNETB0 (epoch = 20, learning rate =0.00005, smoothing =0.1 using unfreeze_last=30)

Epoch	Accuracy	Loss	Accuracy Validation	Validation Loss
1	88.63	0.6198	97.75	0.4534
5	92.51	0.5416	98.50	0.4431
10	93.77	0.5053	99.00	0.413
15	93.96	0.4952	99.50	0.4042
20	97.17	0.4626	99.25	0.4021

3.3. Test-Time Augmentation (TTA) to improve the robustness and accuracy of coffee image classification models.

The purpose of implementing the Test-Time Augmentation (TTA) concept to improve the robustness and accuracy of coffee image classification model predictions is to obtain more stable and noise-resistant predictions by predicting several augmented versions of the input image. The stage begins with the ImageDataGenerator initialized with various augmentations (rotation, shift, zoom, brightness adjustment, horizontal flip) where the original prediction results are stored, and then the next prediction with the augmented version. The reason for using Test-Time Augmentation (TTA) for deployment in real situations so that predictions become more resistant to interference, as well as for evaluating models on images of varying quality in the field so that the output results are more stable. For example, the coffee bean dataset is loaded in Figure 7 then the model produces a "Green" class prediction with the highest confidence value of 0.802 compared to other classes Dark = 0.109843, Medium 0.070624 and Light = 0.017206.

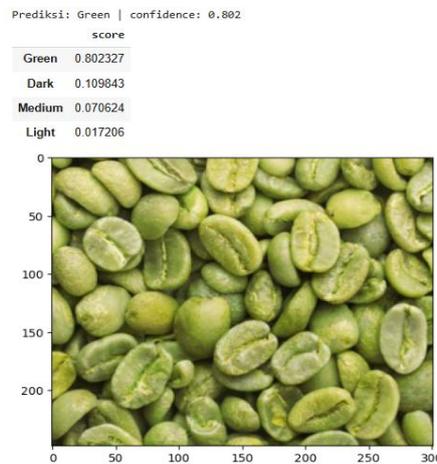


Figure 7. Visualization results of predictions with confidence values and class

A confidence score of 0.802 indicates that, of all the probabilities calculated by the model after the softmax function, the value for "Green" is the largest, so the model classifies the image as belonging to that category. A confidence score above 80%, or 0.802, indicates that the model is quite accurate and its predictions are reliable for practical applications.

3.4. Deployment on low-end devices with quantization

This stage begins by converting the H5 model to TensorFlow Lite (TFLite) to make the model lighter and optimized for mobile or embedded devices. Quantization is then performed using a representative dataset that emulates the input (100 random images) as an optimization reference and sets the target operation and input-output type to 8-bit integer (int8) types, so that the model becomes smaller and more efficient without much loss of accuracy.

4. DISCUSSIONS

The model evaluation results show that from a total of 400 test images divided into 4 classes: Dark, Green, Light, and Medium, both the H5 model and the TFLite version achieved a very high accuracy of 99.50%. This indicates that both are equally reliable in classifying the four classes without significant performance differences, as seen from the zero accuracy difference value. Furthermore, the inference time for the TFLite model is significantly faster than the H5 model, with an average speed of 0.0323 seconds per image compared to 0.1999 seconds for H5, resulting in a 6.19-fold speedup in execution, dan shown in Table 3.

Table 3. Comparison results of model performance with a single object

EfficientNetB0	Accuracy	Inference Time	Confidence	Inference Time
Model H5	99.50%	0.1999 second	0.8091	0.3281
Model TFLite	99.50%	0.0323 second	0.8091	0.0479
Speedup TFLite		6.19x		6.85x

The classification report, which includes accuracy using inference time, confidence using inference time, and visualization with a bar chart shown in Figure 8, provides a detailed overview of the model's ability to recognize each class very well, with near-perfect scores across all evaluation parameters. This performance demonstrates the success of the model optimization process in the efficient TFLite format, providing a speed advantage without sacrificing accuracy, which is very beneficial for applications on resource-constrained devices such as smartphones or embedded systems.

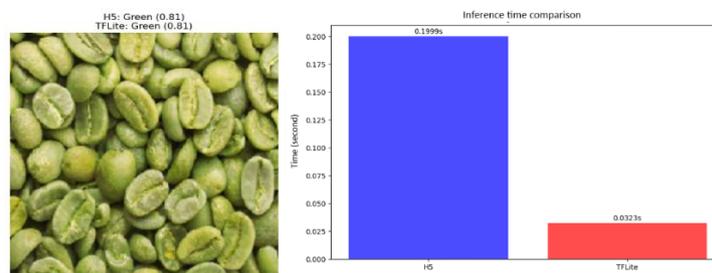


Figure 8. Comparison visualization of H5 and TFLite models based on their inference time

Multi-object detection has advantages over single-object detection because it can recognize and classify multiple objects simultaneously in a single image, as well as accurately determine the location of each object. While single-object detection is simpler and faster because it focuses on only one object in the image, multi-object detection offers flexibility and more comprehensive information, although it requires more computational resources and a more complex inference process. Therefore, multi-object detection is better suited for applications that require simultaneous detection of multiple objects with accurate object localization. Because this study used coffee beans as an object, which can contain multiple coffee beans during image detection, in addition to single-image detection, multi-image detection was also performed, as described in Table 4 for the model performance analysis. While the confidence value of alignment in each class is depicted in the graph in Figure 9.

Table 4. Comparative analysis of model performance with multi-image and the highest confidence value

EfficientNetB0	Image	Class	Confidence	Inference Time (s)	Speedup
Model H5	1	Green	0.8091	0.4198	1
	2	Green	0.7767	0.2154	1
	3	Green	0.3493	0.4645	1
	4	Light	0.5557	0.3648	1
Model TFLite	1	Green	0.8091	0.0672	6.25x
	2	Green	0.7767	0.0337	6.39x
	3	Green	0.3493	0.067	6.94x
	4	Light	0.5557	0.0772	4.73x
Average Speedup TFLite					5.98x

Table 4 summarizes the results of the comparative analysis of the performance of the EfficientNetB0 model in two different formats, namely the H5 Model and the TFLite Model on 4 test images (multi-image) with the classes appearing Green and Light from the 4 classes predicted in the model (Green, Dark, Medium, Light). The TFLite model shows a much faster inference time on all images compared to the H5 model, for example for image 1, the inference time of the H5 Model is 0.4198 seconds while the TFLite Model is only 0.0672 seconds, resulting in a speedup of approximately 6.25 times. The confidence values between the two models are very similar, indicating that the conversion process to TFLite does not significantly reduce the model's confidence. The average speedup of the TFLite Model compared to the H5 Model is 5.98 times, indicating high efficiency in TFLite for faster inference without losing confidence accuracy.

Thus, this study is able to outperform previous studies as shown, most models with CNN architecture, even with Efficiency B0, have not been deployed on low-end devices [41]. The model in this study is able to produce performance with an average speed of H5 model conversion on the TFLite model increasing 5.98 times with the highest confidence value of 0.8091 or 80.91% superior to previous studies which only had a model of 75% when tested[42].

5. CONCLUSION

This research demonstrates that a coffee bean image classification model developed using EfficientNetB0 as a backbone with several additional layers is capable of recognizing data patterns very well. In the initial training phase using transfer learning, the model achieved validation accuracy of up to 98.25% at the 20th epoch, demonstrating a significant performance improvement without overfitting.

Fine-tuning EfficientNetB0 by unfreezing the last 30 layers achieved 97.17% training accuracy and 99.25% validation accuracy with consistent loss reduction, supported by Test-Time Augmentation (TTA) which improves prediction stability to >80% confidence against variations in field image quality. Deployment to TensorFlow Lite (TFLite) with 8-bit quantization resulted in a lighter model that maintained 99.50% accuracy and accelerated inference by up to 6x compared to the original H5 model, and excelled at multi-object detection without sacrificing classification confidence.

Overall, the model development strategy using EfficientNetB0, careful fine-tuning, TTA, and quantization for deployment successfully optimized accuracy, stability, and inference speed, making it suitable for applications on constrained devices. Key contributions include optimizing TFLite through effective quantization, enabling high performance on resource-constrained devices without sacrificing significant accuracy, thus opening up practical ML deployment opportunities in edge computing. These results emphasize the importance of post-training quantization for balancing speed and accuracy in real-time applications, increasing by 5.98 times with the highest confidence value of 0.8091 or 80.91% superior to previous research that only had a model accuracy of 75%. Future research developments include comparing CNN architectures and combining them in a multimodal model using fusion for image and text data integration.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest between the authors or with the research object in this paper.

ACKNOWLEDGEMENT

We would like to express our gratitude to all parties involved directly or indirectly in the completion of this research.

REFERENCES

- [1] U. P. Ganesha, “Klasifikasi Kualitas Biji Kopi Robusta Dengan Metode Naive Bayes,” vol. 10, pp. 280–289, 2023.
- [2] S. Wahyudi, “Aplikasi Deteksi Kualitas Biji Kopi Menggunakan Metode Histogram Equalization Berbasis Android,” vol. 2, no. 1, pp. 50–56, 2018.
- [3] K. Ciptady, M. Harahap, J. Jonvin, Y. Ndruru, and I. Ibadurrahman, “Prediksi Kualitas Kopi Dengan Algoritma Random Forest Melalui Pendekatan Data Science,” *Data Sci. Indones.*, vol. 2, no. 1, 2022, doi: 10.47709/dsi.v2i1.1708.
- [4] D. Herdhiansyah, S. Sudarmi, S. Sakir, and A. Asriani, “Analisis Faktor Prioritas Pengembangan Komoditas Perkebunan Unggulan Dengan Metode Ahp (Analytical Hierarchy Process,” *J. Tek. Pertan. Lampung (Journal Agric. Eng.*, vol. 10, no. 2, p. 239, 2021, doi: 10.23960/jtep-l.v10i2.239-251.
- [5] A. Agung Istri Ratnadewi, R. Anggraini, A. Yulvia, and A. A. Ngurah Nara Kusuma, “Extraction and Purification of Xylooligosaccharides from Coffee Peel Using Endo- β -1,4-D-Xylanase XynBTN63D and Adsorption for Prebiotics,” *Coffee Sci.*, vol. 20, pp. 1–11, 2025, doi: 10.25186/v20i.2354.
- [6] V. Carolina Achicanoy Mejia, L. Alberto Sandoval Mejia, and S. de los Angeles Morales Vanegas, “Profiling coffee consumers: an instrument based on values and attitudes for Spanish-speaking countries in Latin America with a sustainability perspective,” *Coffee Sci.*, vol. 20, 2025, doi: 10.25186/v20i.2346.
- [7] R. Rusinek *et al.*, “How to Identify Roast Defects in Coffee Beans Based on the Volatile Compound Profile,” *Molecules*, vol. 27, no. 23, pp. 1–13, 2022, doi: 10.3390/molecules27238530.
- [8] N. L. H. Hien, A. L. Kor, M. C. Ang, E. Rondeau, and J. P. Georges, *Image Filtering Techniques for Object Recognition in Autonomous Vehicles*, vol. 30, no. 1. 2024. doi: 10.3897/jucs.102428.
- [9] A. Ardiansyah, J. Triloka, K. kunci-Pengolahan Citra, D. Kesegaran Buah, and K. Akurasi, “Evaluasi Kinerja Model YOLOv8... 357.”
- [10] S. Widyaningtyas, M. Arwani, Sucipto, and Y. Hendrawan, “Improving the accuracy of green bean palm civet coffee purity classification using wrapper feature selection,” *Coffee Sci.*, vol. 20, 2025, doi: 10.25186/v20i.2277.
- [11] S. C. Vanegas-Ayala, D. D. Leal-Lara, and J. Barón-Velandia, “Roasted coffee beans characterization through optoelectronic color sensing,” *Coffee Sci.*, vol. 18, 2023, doi: 10.25186/v18i.2156.
- [12] M. A. B. Bhuiyan, H. M. Abdullah, S. E. Arman, S. Saminur Rahman, and K. Al Mahmud, “BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases,” *Smart Agric. Technol.*, vol. 4, Aug. 2023, doi: 10.1016/j.atech.2023.100214.
- [13] T. Jiang *et al.*, “Improved YOLOv8 Model for Lightweight Pigeon Egg Detection,” *Animals*, vol. 14, no. 8, pp. 1–18, 2024, doi: 10.3390/ani14081226.
- [14] M. C. da Silva Andrea, J. P. F. de Oliveira Nascimento, F. C. M. Mota, and R. de Souza Oliveira, “Predictive framework of plant height in commercial cotton fields using a remote sensing and machine learning approach,” *Smart Agric. Technol.*, vol. 4, Aug. 2023, doi: 10.1016/j.atech.2022.100154.
- [15] M. N. Yusri, I. P. Ramadhani, and A. B. Aswar, “Citra Digital Dan Jaringan Syaraf Tiruan,” vol. 02, no. May, pp. 36–43, 2021.
- [16] A. Astaraja, B. S. Syamsudin, M. Diaz, and M. Dhafin, “Optimizing Coffee Ripeness Classification Using Yolov5 for Automated Detection and Sorting,” vol. 4, 2025.
- [17] X. Huang, Z. Liu, S. Y. Liu, and T. K. T. Cheng, “Efficient and Robust Quantization-aware Training via Adaptive Coreset Selection,” *Trans. Mach. Learn. Res.*, vol. 2024, pp. 1–23, 2024.
- [18] J. A. Suarez-Peña, H. F. Lobaton-García, J. I. Rodríguez-Molano, and W. C. Rodriguez-Vazquez, *Machine Learning for Cup Coffee Quality Prediction from Green and Roasted Coffee Beans Features*, vol. 1274 CCIS, no. October. Springer International Publishing, 2020. doi: 10.1007/978-3-030-61834-6_5.

- [19] A. Melke Woldegebriel, "Coffee (*Coffea arabica* L.) Cup Quality Varies with Growing Environments in Ethiopia: Climate and Soil Perspectives," *Coffee Sci.*, vol. 20, 2025, doi: 10.25186/v20i.2295.
- [20] W. de M. Moura *et al.*, "Identification of morphoagronomic traits correlated with the N use efficiency in coffee," *Coffee Sci.*, vol. 20, 2025, doi: 10.25186/v20i.2303.
- [21] E. Aghdamifar, V. Rasooli Sharabiani, E. Taghinezhad, A. Rezvanivand Fanaei, and M. Szymanek, "Non-destructive method for identification and classification of varieties and quality of coffee beans based on soft computing models using VIS/NIR spectroscopy," *Eur. Food Res. Technol.*, vol. 249, no. 6, pp. 1599–1612, 2023, doi: 10.1007/s00217-023-04240-x.
- [22] R. Arablouei, L. Wang, C. Phillips, L. Currie, J. Yates, and G. Bishop-Hurley, "In-situ animal behavior classification using knowledge distillation and fixed-point quantization," *Smart Agric. Technol.*, vol. 4, Aug. 2023, doi: 10.1016/j.atech.2022.100159.
- [23] S. Ashkboos, B. Verhoef, T. Hoefler, E. Eleftheriou, and M. Dazzi, "EfQAT: An Efficient Framework for Quantization-Aware Training," 2024, [Online]. Available: <http://arxiv.org/abs/2411.11038>
- [24] E. Yvinec, A. Dapogny, and K. Bailly, "NUPES : Non-Uniform Post-Training Quantization via Power Exponent Search," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–11, 2025, doi: 10.1109/TPAMI.2025.3593987.
- [25] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile, "AIBERTO: Italian BERT language understanding model for NLP challenging tasks based on tweets," *CEUR Workshop Proc.*, vol. 2481, 2019.
- [26] J. Lang, Z. Guo, and S. Huang, "A Comprehensive Study on Quantization Techniques for Large Language Models," *2024 4th Int. Conf. Artif. Intell. Robot. Commun. ICAIRC 2024*, pp. 224–231, 2024, doi: 10.1109/ICAIRC64177.2024.10899941.
- [27] V. No, D. Gusmaliza, and S. Aminah, "Edumatic : Jurnal Pendidikan Informatika Sistem Identifikasi Kualitas Biji Kopi Robusta berbasis Image Processing dengan Support Vector Machine," vol. 8, no. 2, pp. 744–753, 2024, doi: 10.29408/edumatic.v8i2.28008.
- [28] T. Firmansyah, R. Kurniawan, and A. T. Hidayat, "Klasifikasi Tingkat Kematangan Roasting Biji Kopi Berbasis Deep Learning dengan Arsitektur MobileNet," vol. 6, no. 2, pp. 1432–1442, 2025, doi: 10.47065/josh.v6i2.6811.
- [29] K. Przybył *et al.*, "Application of Machine Learning to Assess the Quality of Food Products—Case Study: Coffee Bean," *Appl. Sci.*, vol. 13, no. 19, 2023, doi: 10.3390/app131910786.
- [30] J. Y. Kim, "Coffee Beans Quality Prediction Using Machine Learning," *SSRN Electron. J.*, pp. 1–7, 2022, doi: 10.2139/ssrn.4024785.
- [31] I. Alfiantama, M. I. Kresnawan, and A. P. Handoko, "Klasifikasi Tingkat Roasting Biji Kopi Dengan Metode CNN," *Pros. Semin. Nas. Teknol. Dan Sains Tahun 2024*, vol. 3, pp. 285–290, 2024.
- [32] T. Agus Siswoyo, E. Novianti, and A. Isnainun, "Changes in the antioxidant activities and angiotensin-I converting enzyme inhibitory protein during the germination of green coffee beans," *Coffee Sci.*, vol. 20, pp. 1–9, 2025, doi: 10.25186/v20i.2332.
- [33] Q. D. Utama *et al.*, "Effect of Ethyl Acetate Concentration on Decaffeination of Robusta Coffee From Lombok, Indonesia," *Coffee Sci.*, vol. 20, no. 2009, 2025, doi: 10.25186/v20i.2306.
- [34] Kamil Fadli, "Pengolahan Citra Digital Menggunakan Metode Yolo Untuk Mendeteksi Kualitas Dari Biji Kopi Berbasis Android," *J. Artif. Intel. dan Sist. Penunjang Keputusan*, vol. 1, no. 1, pp. 120–125, 2023, [Online]. Available: <https://jurnalmahasiswa.com/index.php/aidanspk>
- [35] C. H. de Freitas, R. D. Coelho, J. de O. Costa, and P. C. Sentelhas, "Smart Coffee: Machine Learning Techniques for Estimating Arabica Coffee Yield," *AgriEngineering*, vol. 6, no. 4, pp. 4925–4942, 2024, doi: 10.3390/agriengineering6040281.
- [36] A. A. Soomro *et al.*, "Data augmentation using SMOTE technique: Application for prediction of burst pressure of hydrocarbons pipeline using supervised machine learning models," *Results Eng.*, vol. 24, no. September, 2024, doi: 10.1016/j.rineng.2024.103233.
- [37] Y. Yu *et al.*, "Design and Operation Parameters of Vibrating Harvester for *Coffea arabica* L.," *Agric.*, vol. 13, no. 3, 2023, doi: 10.3390/agriculture13030700.
- [38] dkk Dzulfian Syafrian, *No 主観的健康感を中心とした在宅高齢者における健康関連指標*

- に関する共分散構造分析Title, vol. 11, no. 1. 2025. [Online]. Available: http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SYSTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI
- [39] A. Z. Putra, C. Chalvin, A. Nurhadi, A. E. Tambun, and S. Defha, “Coffee Quality Prediction with Light Gradient Boosting Machine Algorithm Through Data Science Approach,” *Sinkron*, vol. 8, no. 1, pp. 563–573, 2023, doi: 10.33395/sinkron.v8i1.12169.
- [40] F. H. Alkallas, A. M. Mostafa, E. A. Rashed, A. B. G. Trabelsi, M. A. I. Essawy, and R. A. Rezk, “Authentication of Roasted Coffee Beans via LIBS: Statistical Principal Component Analysis,” *Coatings*, vol. 13, no. 10, 2023, doi: 10.3390/coatings13101790.
- [41] Y. Ji, J. Xu, and B. Yan, “Coffee Green Bean Defect Detection Method Based on an Improved YOLOv8 Model,” *J. Food Process. Preserv.*, vol. 2024, no. 1, 2024, doi: 10.1155/2024/2864052.
- [42] F. F. L. Dos Santos, J. T. F. Rosas, R. N. Martins, G. de M. Araújo, L. de A. Viana, and J. de P. Gonçalves, “Quality assessment of coffee beans through computer vision and machine learning algorithms,” *Coffee Sci.*, vol. 15, no. 1, pp. 1–9, 2020, doi: 10.25186/v15i1.1752.