Real-time Detection of Defects in Coffee Beans using Object Detection and Multi-Label Classification

[1, *]*Chieh-Ming Yang* (楊哲旻), [1]*Chun-Liang Lin* (林俊良),

[1]*Zhe-Yu Xu* (許哲瑜), *and* [1]*Jen-Yeu Chen* (陳震宇)

[1] Department of Electrical Engineering,
National Dong Hwa University, Hualien, Taiwan,
[*]E-mail: j82887@gmail.com

ABSTRACT

In today's society, where coffee has become a ubiquitous beverage, the demand for coffee beans has experienced rapid growth. Achieving a high-quality cup of coffee relies heavily on the careful selection of premium coffee beans. Traditionally, coffee bean sorting was reliant on human visual inspection. However, the need for machine-based real-time classification of multiple coffee beans has become essential. Common coffee bean defects encompass various issues such as insect-damage, mold, and breakage. This research aims to develop a system that accurately identifies and classifies defects in coffee beans. The study achieves exceptional speed and accuracy by combining YOLOv7 Tiny with a multi-label classifier. The average detection time per image is 14.84 ms, with an average precision of 99.36% and an accuracy of 99.52% respectively. The results demonstrate that the developed coffee bean defect classification system effectively filters out defective beans and can be operated in real-time on a conveyor belt.

Keywords: *Coffee Bean, Object Detection, Multi-label, Real-time.*

1. INTRODUCTION

Coffee green beans can develop defects during various stages, including harvesting, drying, husking, transportation, and storage. Foreign objects and severe color defects can be removed through color sorting machines, but the identification of subtle defective beans often requires manual screening. In the field of computer vision, the technology of deep neural networks has achieved remarkable advancements in various industries due to its exceptional performance and wide applicability. This technology is commonly employed for a range of tasks, including regression, classification, object detection, and image segmentation. A research paper highlighted the application of deep learning in the agricultural sector, emphasizing its potential to enable intelligent, sustainable, and safer methods of production, screening, and sales [1]. Furthermore, Othman's comprehensive review explores the latest advancements in artificial intelligence technology and its seamless integration with diverse sensing devices, enabling the detection of food adulteration and agricultural product defects to ensure the production of high-value products [2].

Based on current literature, real-time computer vision-based detection typically categorizes coffee beans as normal or defective without further classification of the defects in the defective beans [3-16]. However, fine-grained classification results in longer detection times, rendering it impractical for real-time applications and challenging to implement in conveyor belt systems. Therefore, to address this gap, this study explores the combination of multiple methods, including object detection and classifiers. Our aim is to find an approach that enables real-time detection and selection of high-quality coffee beans, while effectively eliminating even the most subtle defective beans. Additionally, we investigate the real-time performance and accuracy of each method to find the optimal balance.

2. MATERIALS AND METHODS

Section 2.1 provides an introduction to the image data used in this study, including its sources and the distribution across different categories. Detailed descriptions of the preprocessing methods employed to enhance subsequent models' performance are presented in Section 2.2. To expand the dataset, Section 2.3 incorporates image augmentation techniques. The deep learning models utilized, including an object detection model, binary classifiers, and multi-label classifiers, are described in Section 2.4. The overall system architecture is illustrated in Figure 1, depicting the system's capability to extract preprocessed coffee bean images from the camera using an object detection model for subsequent defect recognition.

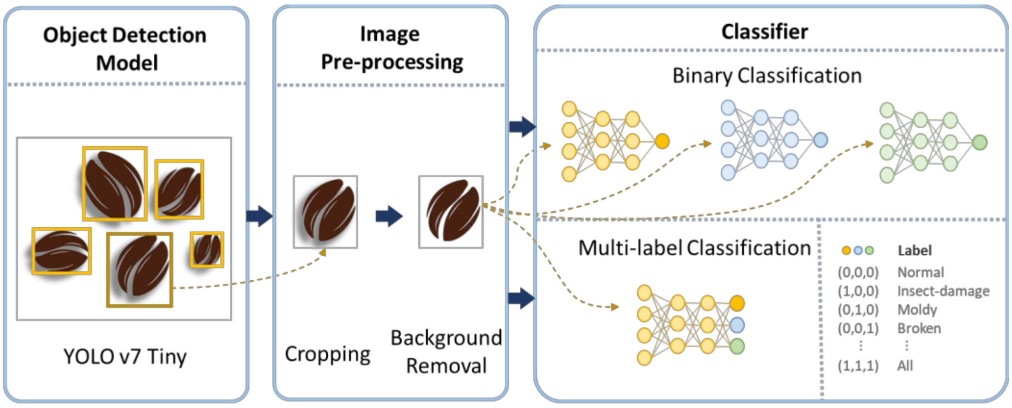


Fig. 1. The flowchart of the proposed method.

2.1. Data Collection

Our data is sourced from a coffee bean supplier, who provided us with the coffee beans. The photographs are taken using the same equipment and under the same controlled environment. The setup involves a two-tier table arrangement, with the first tier consisting of a transparent glass surface for placing the coffee beans and the second tier featuring a white paper. This setup is designed to minimize external factors such as shadows. The environment in which the photographs are taken is depicted in Figure 2. Additionally, in Section 2.2, we employ image preprocessing techniques to significantly mitigate this issue. In total, we collect a total of 741 coffee beans, including 190 normal beans, 115 insect-damage beans, 162 moldy beans, and 274 broken beans.

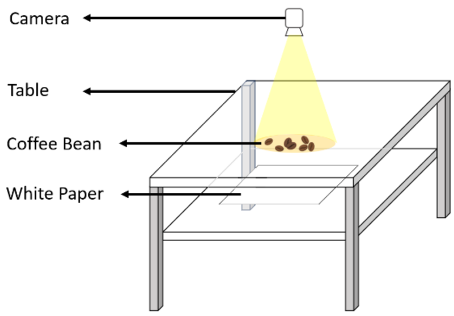


Fig. 2. Controlled environment setup for coffee bean photography.

2.2. Image Pre-processing

First, we use an object detection model to crop the image for the positions of coffee beans. Then, we apply image preprocessing methods. These preprocessing methods aim to improve the performance of the defect classifier. To enhance the features of coffee beans and reduce the influence of the external environment, we employ a background removal method. Each image's color channels are converted from *RGB* to *HSV* color space, and the pixel values of each channel are divided by 255 to ensure they range from 0 to 1. Subsequently, we evaluate the *S* channel values for each pixel. If a value is between 0 and 0.14, the corresponding *RGB* pixel value in the cropped image is changed to (255, 255, 255) as shown in Formula 1, to obtain coffee bean images with a white background, as illustrated in Figure 3.

$$(R', G', B') = \begin{cases} (255,255,255) & 0 \le S \le 0.14 \\ (R, G, B) & otherwise \end{cases} \quad (1)$$

(A) Original (B) Cropped (C) Background removed

Fig. 3. Image preprocessing steps.

2.3. Data Augmentation

During deep learning model training, having a large amount of training data is essential to avoid overfitting. To achieve this objective, this study utilizes data augmentation techniques, including random rotation and horizontal flipping of images. The increased number of images after augmentation is illustrated in Table 1.

Table 1. Number of images after data augmentation.

	Training Set	Test Set	Total
Normal	3,244	741	3,985
Insect-damage	1,928	484	2,412
Moldy	2,678	723	3,401
Broken	2,397	614	3,011
Total	10,247	2,562	12,809

2.4. Deep Learning Model

We employ an object detection model to perform coffee bean bounding box selection, followed by the utilization of a binary and multi-label classifier for defect classification. This two-stage approach enables us to simultaneously achieve bean object detection and defect recognition. In the object detection method, the selected coffee beans are treated as a whole without undergoing further classification. On the other hand, defect recognition is divided into two parts: binary classification and multi-label classification, and compared accordingly. The classifier employs lightweight Convolutional Neural Network (CNN) models, including MobileNetV3Small [17], EfficientNetV2B0 [18], and NASNetMobile [19].

2.4.1. Object Detection
We use the labelme [20] tool to annotate coffee bean images and train an object detection model. This enables us to accurately select all coffee beans in the captured photos. Subsequently, we crop each coffee bean from these images into rectangular pictures. After that, we perform preprocessing and data augmentation on these images to create a dataset suitable for binary and multi-label classification training. In this study, we employ the lightweight Tiny version of the YOLOv7 [21] object detection model to achieve faster detection speed.

2.4.2. Binary Classification
Due to the possibility of coffee beans having multiple defects simultaneously, we train three binary classifiers in a cascaded manner for detection. The images are divided into two categories: non-defective and defective. It is important to note that non-defective images may contain other categories of defects. If all three defect classifiers classify a bean as non-defective, it is classified as normal.

2.4.3. Multi-label Classification
Multi-label tasks are more complex compared to single-label tasks such as binary or multi-class classification. The goal of multi-label tasks is to predict one or more corresponding labels for each sample. In our case, we label normal, insect-damage, moldy, and broken beans as (0, 0, 0), (0, 0, 1), (0, 1, 0), and (1, 0, 0) respectively. Similarly, the same labeling scheme is used for multiple defect situations. For example, coffee beans with both insect-damage and breakage will have the label (1, 0, 1). In total, we have 2^3 label categories.

2.5. Evaluation Metrics

To evaluate the performance of coffee bean defect detection, various metrics are used, including accuracy, specificity, sensitivity, precision, and F_1 score. Accuracy, specificity, sensitivity, and precision are defined in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as shown in equations (2)-(5). The F_1 score is defined as the harmonic mean of precision and recall (sensitivity), calculated using equation (6). The object detection model can be evaluated by incorporating the 101-point interpolation Average Precision (AP) metric, defined as calculated in equation (7), where $\rho(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2}$$

$$Specificity = \frac{TN}{TN + FP}, \tag{3}$$

$$Sensitivity\ (Recall) = \frac{TP}{TP + FN}, \tag{4}$$

$$Precision = \frac{TP}{TP + FP}, \tag{5}$$

$$F_1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{6}$$

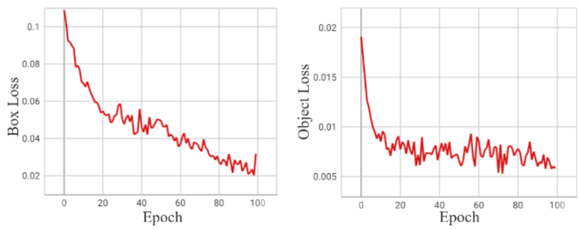
$$AP = \frac{1}{101} \sum_{r \in \{0, 0.01, \ldots, 1\}} \max_{\tilde{r}:\tilde{r} \geq r} \rho(\tilde{r}). \tag{7}$$

2.6. Environment and Hyperparameter

In this study, we use the NVIDIA GeForce RTX 2080 Ti GPU as the computational device to expedite the training process of neural networks. In order to accomplish the objectives of neural network training, we employ Python 3.9 as the programming language and utilize two robust deep learning frameworks, namely PyTorch 2.0.1 and TensorFlow 2.10, to train object detection and classification models. For the YOLOv7 Tiny model, we implement transfer learning by utilizing pre-trained weights obtained from the MS COCO dataset during the model training phase. The purpose of this approach is to facilitate rapid convergence and enhance the performance of the model. Furthermore, the model's hyperparameters include an initial learning rate of 0.01, a learning rate decay factor of 0.1, and a momentum of 0.937. Regarding the classification model, we opt for the Adam [22] optimizer with a learning rate of 0.0001. We train the model for 50 epochs, using a batch size of 16. Similarly, we employ the weights from the ImageNet dataset for transfer learning purposes.

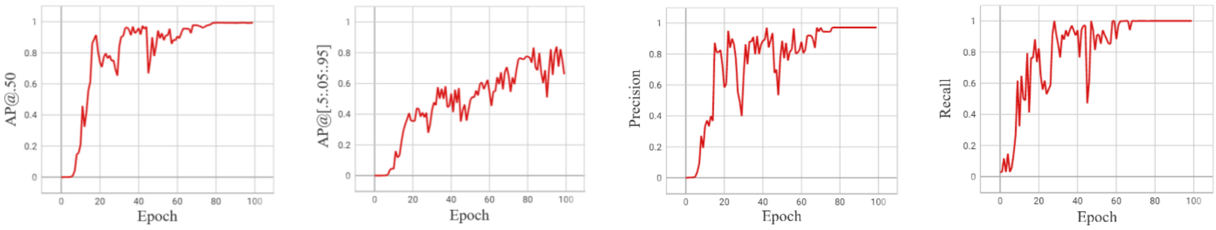
3. RESULTS

In object detection, the learning curve of the YOLOv7 Tiny model is shown in Figure 4, where both the object loss and box loss of the training set converge nicely. This study employs the non-maximum suppression (NMS) method [23] to remove duplicate bounding boxes, with parameters set at an intersection over union (IoU) threshold of 0.7 and a confidence threshold of 0.001. The corresponding performance metrics of the test set are as follows: AP@.50, AP@[.5:.05:.95], precision, and recall achieve 99.36%, 66.00%, 97.14%, and 99.99% respectively, as shown in Figure 5. Additionally, the average inference time of the model is 11.20 ms.



(A) Box Loss (B) Object Loss

Fig. 4. Learning curves



(A) AP@.50 (B) AP@[.5:.05:.95] (C) Precision (D) Recall

Fig. 5. Performance metrics of the test set

Table 2. Performance of binary and multi-label classification.

Task			Accuracy	Specificity	Sensitivity	Precision	F_1 score
Binary Classification	MobileNetV3Small	Normal	99.69%	99.62%	99.87%	99.94%	99.78%
		Insect-damage	100.00%	100.00%	100.00%	100.00%	100.00%
		Moldy	98.59%	99.03%	98.42%	96.11%	97.55%
		Broken	99.80%	100.00%	99.74%	99.19%	99.59%
		Average	99.52%	99.66%	99.51%	98.81%	99.23%
	EfficientNetV2B0	Normal	99.88%	99.84%	100.00%	100.00%	99.92%
		Insect-damage	100.00%	100.00%	100.00%	100.00%	100.00%
		Moldy	99.88%	99.59%	100.00%	100.00%	99.79%
		Broken	100.00%	100.00%	100.00%	100.00%	100.00%
		Average	99.94%	99.86%	100.00%	100.00%	99.93%
	NASNetMobile	Normal	100.00%	100.00%	100.00%	100.00%	100.00%
		Insect-damage	100.00%	100.00%	100.00%	100.00%	100.00%
		Moldy	99.96%	100.00%	99.95%	99.86%	99.93%
		Broken	100.00%	100.00%	100.00%	100.00%	100.00%
		Average	99.99%	100.00%	99.99%	99.97%	99.98%
Multi-label Classification	MobileNetV3Small	Normal	99.53%	100.00%	99.34%	100.00%	99.66%
		Insect-damage	99.92%	99.95%	99.79%	99.79%	99.79%
		Moldy	99.37%	99.40%	99.30%	98.49%	98.89%
		Broken	99.25%	99.48%	98.53%	98.37%	98.45%
		Average	99.52%	99.71%	99.24%	99.16%	99.20%
	EfficientNetV2B0	Normal	99.92%	100.00%	99.89%	100.00%	99.94%
		Insect-damage	100.00%	100.00%	100.00%	100.00%	100.00%
		Moldy	99.68%	99.67%	99.72%	99.17%	99.44%
		Broken	99.76%	99.79%	99.67%	99.35%	99.51%
		Average	99.84%	99.87%	99.82%	99.63%	99.72%
	NASNetMobile	Normal	99.76%	100.00%	99.67%	100.00%	99.83%
		Insect-damage	99.92%	99.95%	99.79%	99.79%	99.79%
		Moldy	99.76%	99.78%	99.72%	99.44%	99.58%
		Broken	99.10%	99.07%	99.18%	97.12%	98.14%
		Average	99.64%	99.70%	99.59%	99.09%	99.34%

According to the results of the test set in Table 2, we observe the performance of binary and multi-label classification. In the binary classification, all models achieve 100% for each metric in detecting insect-damage, demonstrating excellent performance. However, the detection capability for the moldy defect slightly decreases compared to other defects. In the multi-label classification, only EfficientNetV2B0 achieves 100% for each metric in detecting insect-damage, demonstrating outstanding capability. Despite the fact that multi-label classification is more challenging than binary classification, the performance of both is comparable based on the metric. Additionally, Table 3 presents the average detection time for each classifier. Since the binary classifier requires three detections, its detection time is inevitably longer than the multi-label classifier. Among the multi-label classifiers, MobileNetV3Small exhibits the shortest average detection time, only 3.64 ms.

In the fastest detection MobileNetV3Small classifier, we use the Gradient Weighted Class Activation Mapping (grad-cam) [24] method to explain the network's classification basis. The heat map in Figure 6 displays each region of the defects, including insect-damage, moldy, broken, insect-damage and moldy, insect-damage and broken, as well as moldy and broken in coffee beans, totaling 6 types. However, it does not include normal beans and beans with all defects. This is because the heat map of normal coffee beans does not show any significant defect features, and the collected images in this study do not include coffee beans with all defects.

Table 3. Comparing average detection time (ms) for binary and multi-label classifiers.

	Binary			Multi-label
	Insect-damage	Moldy	Broken	-
MobileNet V3Small	3.75	3.75	3.80	3.64
EfficientNet V2B0	10.63	10.58	10.82	9.42
NASNet Mobile	13.15	13.39	13.07	12.61

4. DISCUSSION

We previously attempt to train a coffee bean defect labeling method using only an object detection model. However, this approach yields very poor detection performance. There are two reasons for this problem. One is that defects cannot be accurately represented using a single rectangular bounding box, as their edges exhibit irregular shapes. The other challenge arises when dealing with coffee beans that have multiple defects, as the features of these defects may overlap, causing the NMS

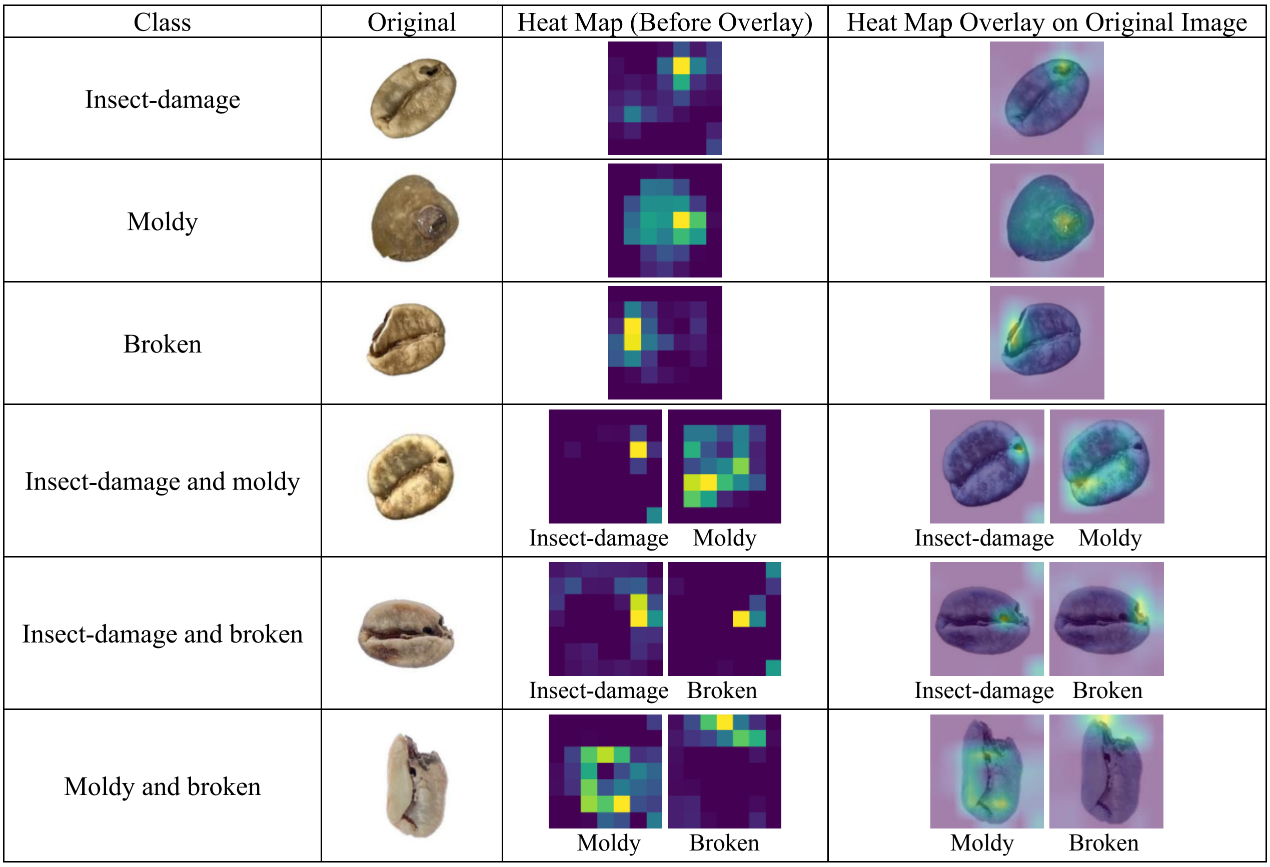


Fig. 6. Visualization of focused areas in coffee beans images using gray-cam heatmap in this study's model.

Table 4. Performance comparison of each model.

Authors (years)	Model / Algorithm	Number of classes	Task	Performance	
Santos et al. (2012) [3]	Partial Least Squares	--	Regression	Relative errors	5%
Pinto et al. (2017) [4]	CNN	6	Classification	Accuracy	72.4% to 98.7%
Arboleda et al. (2018) [5]	Image Processing	2	Classification	Accuracy	100%
Huang et al. (2019) [6,7]	CNN	2	Classification	Accuracy	93.00%
Janandi and Cenggoro (2020) [8]	ResNet-152	3	Classification	Accuracy	73.3%
Arunkumar and Berihun (2020) [9]	CNN	3	Classification	Accuracy	98.38%
Santos et al. (2020) [10]	Deep Neural Network Random Forest Support Vector Machine	5	Classification	Accuracy	94.80% 88.50% 94.70%
Tseng et al. (2021) [11]	ResNet-18 (Knowledge Distillation)	2	Classification	Accuracy	91.00%
Chang and Huang (2021) [12]	Improved AlexNet	2	Classification	Accuracy	100%
Yang et al. (2021) [13]	CNN	2	Classification	Sensitivity Precision F_1 score	97.53% 96.48% 96.54%
Hsia et al. (2022) [14]	LDCNN	2	Classification	Accuracy Sensitivity Precision F_1 score	98.38% 97.89% 98.60% 98.24%
Luis et al. (2022) [15]	YOLOv5	3	Object detection	Accuracy	95.11%
Wang et al. (2022) [16]	Slim-CNN	2	Classification	Accuracy Sensitivity Precision F_1 score	92.66% 92.81% 94.03% 93.42%
Proposed Method	MobileNetV3Small [17]	4	Object detection & Classification	Accuracy Specificity Sensitivity Precision F_1 score	99.52% 99.71% 99.24% 98.16% 99.20%

method to remove overlapping bounding boxes and resulting in the detection of only a single defect. Additionally, we consider the option of bounding the entire coffee bean and using a single object detection model to determine its category. However, the problem of overlapping regions still persists when dealing with coffee beans that have multiple defects. If only using an object detection model to accomplish this task, we would need to train separate models to identify each of the three defect types and then concatenate them. However, object detection typically incurs a significantly longer detection time compared to classifiers. Consequently, we ultimately decided not to pursue this approach.

For object detection, we choose the YOLOv7 Tiny model, which achieves an AP of 99.36%. The average detection time per image is 11.20 ms. Regarding the classifier, it exhibits comparable detection performance, and the MobileNetV3Small multi-label classifier has the shortest average detection time per image, only 3.64 ms. The shortest detection time is achieved by combining object detection with the classifier, totaling approximately 14.84 ms (excluding preprocessing steps). Heatmaps generated by the model provide accurate delineation of defect areas and enable the simultaneous detection of multiple defects.

In Table 4, we have listed the relevant literature, including the methods, the number of classes, the detection tasks, and the performance of each study. Due to the different coffee bean datasets used in each paper and the varying number of detection classes, a direct comparison of the methods' performance is not possible. Therefore, we will indirectly describe the comparison between the literature and our research. Arboleda et al. [5] employed the upper and lower limits of each channel in

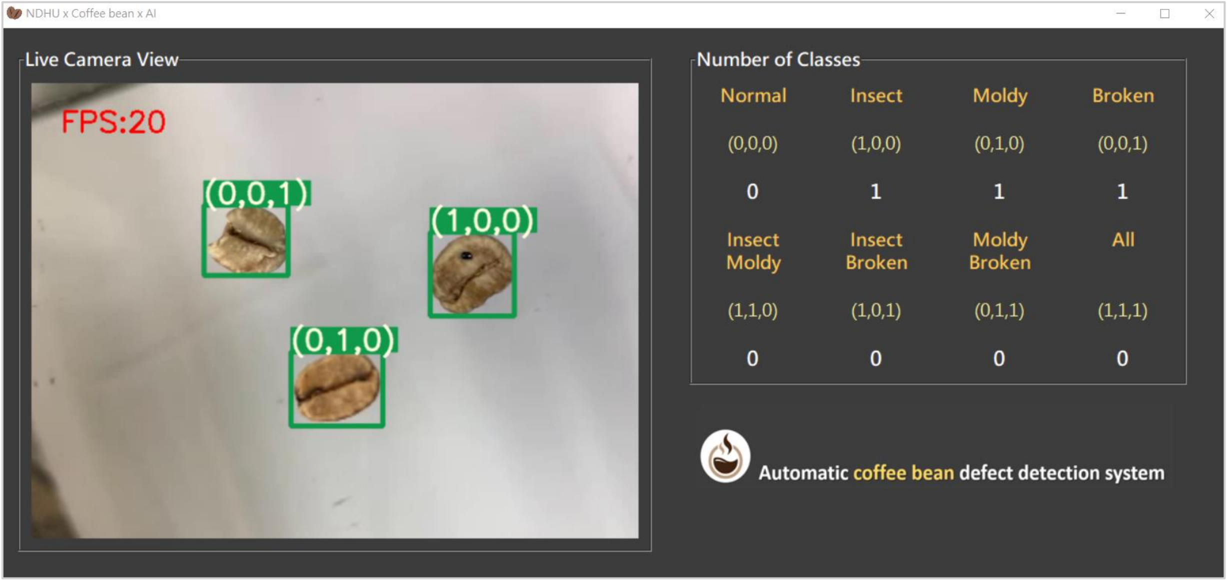


Fig. 7. Coffee bean defect detection user interface.

the RGB color space to determine the presence of defective beans. As the image dataset used in their study consisted of relatively simple coffee bean images with only color defects, the method achieved 100% accuracy. Chang and Huang [12] utilized an improved AlexNet to classify coffee beans into eight categories, achieving an accuracy of 95.1%. When utilizing a simple binary classification method for the task, the accuracy reached 100%. None of the other studies surpassed the accuracy of the method proposed in this research, which achieved an accuracy of 99.52%.

We develop a user interface for the real-time coffee bean defect detection system, as shown in Figure 7. We have designed it to mimic the operation of a conveyor belt, using a moving transparent glass to enable the system to detect coffee beans as they appear within the camera's field of view. The interface is capable of instantaneously identifying various defect categories of the coffee beans in the current frame, and it can also annotate their locations for future bean-picking operations by robotic arms. However, a drawback of this system is that the processing speed slows down when the number of coffee beans in the frame increases because the classifier needs to detect all the coffee beans.

We attempt to accelerate model inference through a common model compression technique known as quantization. However, this technique has not been integrated into the system yet. We conduct simple tests on the MobileNetV3Small multi-label classifier using post-training quantization (PTQ) [25] methods, including dynamic range quantization, float16, and full integer quantization. It is important to note that full integer quantization may lead to faster execution speed on the same hardware device, but it significantly reduces the accuracy of coffee bean defect detection. On the other hand, float16 quantization demonstrates the best performance, achieving accuracies of 97.74, 99.96%, 97.85%, and 99.38% for detecting normal, insect-damage, moldy, and broken beans, respectively, as illustrated in Figure 8. The average detection time per image is 2.86 ms, which is faster than the float16 quantization of 3.64 ms.

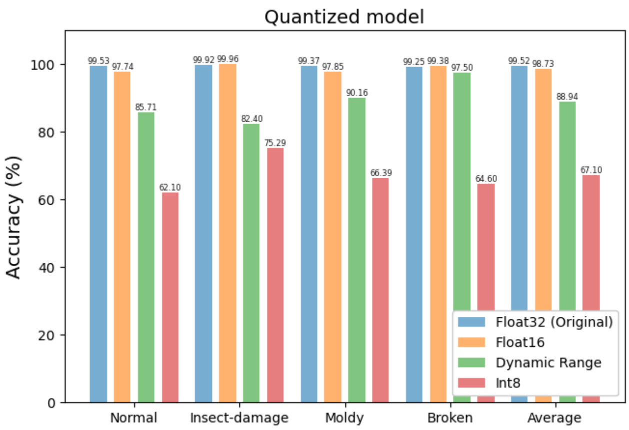


Fig. 8. Quantization methods' performance comparison.

In future work, we plan to delve deeper into exploring quantization techniques, considering post-training quantization and quantization-aware training methods, in order to reduce computational requirements. Focusing on quantizing the classifier has already provided preliminary results. Moving forward, we intend to broaden our testing scope to include quantizing YOLOv7 Tiny Model, enabling a comprehensive evaluation of quantization techniques on the overall model performance. We will evaluate the performance of various quantization methods in terms of accuracy and detection time. Furthermore, we aim to enhance defect detection by incorporating the ability to detect defects of varying degrees. This will enable faster and more effective screening of high-quality coffee beans, contributing to the advancements in precision agriculture.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 111-2622-8-259-001-TE2.

REFERENCES

[1] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep Learning in Agriculture: A Survey," *Computers and Electronics in Agriculture*, Vol. 147, pp. 70-90, 2018.

[2] S. Othman, N. R. Mavani, M. A. Hussain, N. Abd Rahman, and J. M. Ali, "Artificial Intelligence-based Techniques for Adulteration and Defect Detections in Food and Agricultural Industry: A Review," *Journal of Agriculture and Food Research*, Vol. 12, 100590, 2023.

[3] J. R. Santos, M. C Sarraguça, A. O S S Rangel, and J. A Lopes, "Evaluation of Green Coffee Beans Quality Using Near Infrared Spectroscopy: A Quantitative Approach," *Food Chemistry*, Vol. 135, No. 3, pp. 1828-1835, 2012.

[4] C. Pinto, J. Furukawa, H. Fukai, and S. Tamura, "Classification of Green Coffee Bean Images Based on Defect Types Using Convolutional Neural Network (CNN)," Proceedings of the 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), Denpasar, Indonesia, 2017.

[5] E. R. Arboleda, A. C. Fajardo, and R. P. Medina, "An Image Processing Technique for Coffee Black Beans Identification," Proceedings of the 2018 IEEE International Conference on Innovative Research and Development (ICIRD), Bangkok, Thailand, 2018.

[6] N. F. Huang, D. L. Chou, and C. A. Lee, "Real-Time Classification of Green Coffee Beans by Using a Convolutional Neural Network," Proceedings of the 2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC), Singapore, 2019.

[7] N. F. Huang, D. L. Chou, C. A. Lee, F. P. Wu, A. C. Chuang, Y. H. Chen, and Y. C. Tsai, "Smart Agriculture Real-Time Classification of Green Coffee Beans by Using a Convolutional Neural Network," *IET Smart Cities*, Vol. 2, No. 4, pp. 167-172, 2020.

[8] R. Janandi and T. W. Cenggoro, "An Implementation of Convolutional Neural Network for Coffee Beans Quality Classification in a Mobile Information System," Proceedings of the 2020 International Conference on Information Management and Technology (ICIMTech), Bandung, Indonesia, 2020.

[9] J. R. A. Kumar and T. Berihun, "Developing Ethiopian Yirgacheffe Coffee Grading Model Using a Deep Learning Classifier," *International Journal of Innovative Technology and Exploring Engineering*, Vol. 9, No. 4, pp. 3303-3309, 2020.

[10] F. Santos, J. Rosas, R. Martins, G. Araújo, L. Viana, and J. Gonçalves, "Quality Assessment of Coffee Beans Through Computer Vision and Machine Learning Algorithms," *Coffee Science*, Vol. 15, 2020.

[11] P. Wang, H. W. Tseng, T. C. Chen, and C. H. Hsia, "Deep Convolutional Neural Network for Coffee Bean Inspection," *Sensors and Materials*, Vol. 33, No. 7, pp. 2299-2310, 2021.

[12] S. J. Chang and C. Y. Huang, "Deep Learning Model for the Inspection of Coffee Bean Defects," *Applied Sciences*, Vol. 11, No. 17, 2021.

[13] P. Y. Yang, S. Y. Jhong, and C. H. Hsia, "Green Coffee Beans Classification Using Attention-Based Features and Knowledge Transfer," Proceedings of the 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Penghu, Taiwan, 2021.

[14] C. H. Hsia, Y. H. Lee, and C. F. Lai, "An Explainable and Lightweight Deep Convolutional Neural Network for Quality Detection of Green Coffee Beans," *Applied Sciences*, Vol. 12, No. 21, 2022.

[15] V. A. M. Luis, M. V. T. Quiñones, and A. N. Yumang, "Classification of Defects in Robusta Green Coffee Beans Using YOLO," Proceedings of the 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 2022.

[16] Y. F. Wang, C. C. Cheng, and J. K. Tsai, "Implementation of Green Coffee Bean Quality Classification Using Slim-CNN in Edge Computing," Proceedings of the 2022 IEEE 5th International Conference on Knowledge Innovation and Invention (ICKII), Hualien, Taiwan, 2022.

[17] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," Proceedings of the 2019 International Conference on Computer Vision, Seoul, Korea, 2019.

[18] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," Proceedings of the 38th International Conference on Machine Learning, Virtual Only, 2021.

[19] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018.

[20] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman "LabelMe: A Database and Web-based Tool for Image Annotation," *International Journal of Computer Vision (IJCV)*, Vol. 77, No. 1-3, pp. 157-173, 2008.

[21] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 2023.

[22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Proceedings of the 3rd International Conference for Learning Representations, San Diego, USA, 2015.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, pp. 1627-1645, 2009.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017.

[25] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. v. Baalen, and T. Blankevoort, "A White Paper on Neural Network Quantization," arXiv preprint arXiv:2106.08295, 2021.