Data and Metadata. 2025; 4:840
doi: 10.56294/dm2025840

ORIGINAL

Binary classification of defects in multiple coffee beans using lightweight convolutional neural networks for embedded systems

Clasificación binaria de defectos en múltiples granos de café mediante redes neuronales convolucionales ligeras para sistemas integrados

Anderson Chamorro-Pinchao[1], Marco Pusdá-Chulde[1], Diego Trejo-España[1], Victor Caranqui-Sánchez[1], Iván García-Santillán[1]

[1]Universidad Técnica del Norte, Facultad de Ingeniería en Ciencias Aplicadas. Ibarra, Ecuador.

Cite as: Chamorro-Pinchao A, Pusdá-Chulde M, Trejo-España D, Caranqui-Sánchez V, García-Santillán I. Binary classification of defects in multiple coffee beans using lightweight convolutional neural networks for embedded systems. Data and Metadata. 2025; 4:840. https://doi.org/10.56294/dm2025840

Submitted: 14-07-2025 **Revised:** 11-09-2025 **Accepted:** 03-12-2025 **Published:** 04-12-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding Author: Iván García-Santillán

ABSTRACT

Coffee and its derivatives are part of the world's gastronomy and are also one of the main export products in Imbabura, Ecuador. Coffee producers and marketers must ensure the quality of their coffee beans, as a single seed can present several morphological defects. However, the selection of quality beans is a manual, tedious, and time-consuming process, prone to misclassification due to human limitations such as fatigue and varying subjective classification criteria. This work aims to detect and classify multiple coffee beans automatically present in an image (both good and bad) using modern convolutional neural networks, specifically YOLOv11 (Nano and Small variants), which are selected for their computational efficiency and good performance in real-time detection tasks. A custom dataset with a total of 4090 RGB images with multiple coffee beans randomly located within each image was collected and manually labeled into two classes (good and bad) using the CVAT tool. Image preprocessing included background variation (white, gray, and black) to improve dataset variability and model robustness. Models were trained on the Kaggle platform with an NVIDIA Tesla P100 GPU (16 GB VRAM), using the PyTorch framework and the Ultralytics library. Results showed an accuracy of 0,880 for YOLO v11-Nano and 0,871 for YOLO v11-Small, along with inference times of 7,89 ms (126,7 fps) and 10,08 ms (99,20 fps), respectively. These results are comparable to those reported in the literature, but now considering the challenge of superimposing multiple coffee beans randomly located in each image with a variable background. These results suggest the potential for applying the model to embedded systems in practical settings for small and medium-sized producers and associations, thereby contributing to technological innovation in the Ecuadorian coffee sector.

Keywords: Coffee Beans Classification; Coffee Defects; CNN; YOLO; Lightweight Network.

RESUMEN

El café y sus derivados forman parte de la gastronomía mundial y son uno de los principales productos de exportación de Imbabura, Ecuador. Los productores y comercializadores de café deben garantizar la calidad de sus granos, ya que una sola semilla puede presentar varios defectos morfológicos. Sin embargo, la selección de granos de calidad es un proceso manual, tedioso y lento, propenso a errores de clasificación debido a limitaciones humanas como la fatiga y la variación de los criterios subjetivos de clasificación. Este trabajo busca detectar y clasificar automáticamente múltiples granos de café presentes en una imagen (tanto buenos como malos) mediante redes neuronales convolucionales modernas, específicamente YOLOv11 (variantes Nano y Small), seleccionadas por su eficiencia computacional y buen rendimiento en tareas de detección en tiempo real. Se recopiló un conjunto de datos personalizado con un total de 4090 imágenes RGB con múltiples granos de café ubicados aleatoriamente dentro de cada imagen y se etiquetó manualmente en dos clases (buenos y malos) utilizando la herramienta CVAT. El preprocesamiento de imágenes incluyó la variación del fondo (blanco, gris y negro) para mejorar la variabilidad del conjunto de datos y la robustez del modelo. Los modelos se entrenaron en la plataforma Kaggle con una GPU NVIDIA Tesla P100 (16 GB de VRAM), utilizando el framework PyTorch y la biblioteca Ultralytics. Los resultados mostraron una precisión de 0,880 para YOLO v11-Nano y 0,871 para YOLO v11-Small, junto con tiempos de inferencia de 7,89 ms (126,7 fps) y 10,08 ms (99,20 fps), respectivamente. Estos resultados son comparables a los reportados en la literatura, pero considerando el desafío de superponer múltiples granos de café ubicados aleatoriamente en cada imagen con un fondo variable. Estos resultados sugieren el potencial de aplicar el modelo a sistemas embebidos en entornos prácticos para pequeños y medianos productores y asociaciones, contribuyendo así a la innovación tecnológica en el sector cafetalero ecuatoriano.

Palabras clave: Clasificación de Granos de Café; Defectos del Café; CNN; YOLO; Red Ligera.

INTRODUCTION

Coffee is one of the most popular beverages worldwide, and mass consumption has driven intense interest in improving its quality and utilizing its by-products within the circular economy.[1] In Ecuador, coffee is a symbolically significant crop with a profound impact on rural communities. Coffee production is one of the country's primary agricultural activities, a vital sector that generates employment and wealth, and contributes to environmental protection. Coffee has been among the ten crops with the largest harvested area, having been produced in more than 19 provinces for the past 15 years,[2] with expectations of a sustained increase in production.[3] In the country, the focus is on improving both the productivity and quality of coffee, facing significant challenges such as morphological defects in the beans, which make their proper selection and classification difficult.[4] In Imbabura (Ecuador), where this study was conducted, several small and medium-sized farms cultivate coffee under suitable climatic conditions. However, they face financial and personnel constraints that complicate the bean selection process, negatively impacting product quality and market value.

High-quality or specialty coffees play a prominent role in the coffee market, where they often fetch high prices. However, their commercialization requires specific certifications, which vary depending on the final destination of the coffee beans.[5] These certifications assess key aspects of the crop, including climate, soil, fertilizer, post-harvest process, and the quality of the beans produced (free from defects).

In developing countries, manual sorting of coffee beans is a common technique among coffee growers; however, it presents several limitations that affect the quality and efficiency of the process. This technique relies heavily on the subjective judgment of the operator (collector), which can lead to inconsistencies and errors in classifying coffee beans. Furthermore, it is a slow and laborious procedure, which increases production costs and reduces the capacity to adequately process large volumes. Staff fatigue and lack of specialized training can significantly reduce the accuracy of morphological defect detection,[6] while external factors such as lighting conditions and environment negatively influence the uniformity of grain analysis and selection.

The application of advanced object detection and tracking techniques in precision agriculture is crucial for enhancing the efficiency and sustainability of the sector, enabling informed decision-making and optimizing resource utilization.[7] Convolutional neural networks are widely used in fields such as computer vision, applied in different industries, such as precision agriculture[8,9,10] security,[11] livestock,[12] medicine,[13,14] education.[15] The ability to process complex images with multiple objects, even in uncontrolled visual environments, significantly reduces human error and speeds up image analysis.

Thus, this work proposes the development of a computer vision system based on convolutional neural networks (CNN) for the detection and classification of coffee beans (good/bad). The model uses the lightweight and modern YOLOv11 architecture (Nano and Small), recognized for its balance between accuracy and computational efficiency, with fewer training parameters, low memory consumption (VRAM), and high inference speed, even on devices with limited resources, such as embedded systems. An image dataset was created and manually annotated into two classes: good and bad beans. The dataset is publicly available for comparison and future investigations at the following link, contributing to this field of study. The project was developed in a controlled environment with adequate lighting conditions for capturing RGB images, facilitating the training of the model, and improving its predictive capacity of multiple coffee beans located randomly within an image, overcoming the limitations of our previous work[6] and other existing ones, where they do not consider either the superposition of several coffee beans or the handling of a variable background in the image, which improves the model's performance and adapts it to real conditions. This also represents a significant contribution to technological innovation in the coffee sector. Furthermore, the studies that report the best results were developed in rigorously controlled environments, using high-resolution cameras, specialized RGB lighting, and configurations of perfectly aligned/organized beans in the image, avoiding any contact or overlap between coffee beans, which restricts their practical application.

In our work, open-source tools such as Python, PyTorch, and OpenCV were utilized for implementing the CNN model for coffee bean classification, CVAT for manual annotation of the custom dataset, and Roboflow for data augmentation. To the best of our knowledge, studies of this kind are very limited in Ecuador, particularly for small and medium-sized producers who may not have access to expensive industrial sorting equipment. Therefore, developing this CNN model is necessary to automate the coffee industry.

The rest of the paper is organized as follows: The subsection relating to works presents a brief review of the literature relevant to our proposal. Section Materials and methods presents the methodology and tools for developing the CNN models. Section Results and Discussion show the results obtained using various metrics, performance graphs, and the findings in comparison with other similar research. Finally, present the section conclusions and future work.

Related works

Below is a summary of some relevant works that serve as the basis for this study. In our previous work, Cevallos et al.(6) used convolutional neural networks (CNNs) with the SSD-MobileNetV2 and SSD-ResNet50 architectures to classify coffee beans as good or bad, based on individual images captured in a controlled environment (with a white background). Data augmentation was applied using the Roboflow tool to increase the size of the original dataset from 3276 to 20 970 images. The results showed an accuracy of 91,65 % with MobileNetV2 and 83,07 % with ResNet50. The best resulting model (SSD-MobileNetV2) outperforms the individual's average accuracy of manual classification (78,9 %), with a lightweight size of 26,5 MB and an inference speed of 2,3 fps. However, the model only allows one grain to be analyzed per image, and it is a binary classification (good and bad), which limits its application in real environments where multiple grains need to be processed simultaneously, even with overlap between them.

In Gonzabay-Jiménez(16), a CNN network based on MobileNetV1 was implemented to classify coffee beans according to shape and color. Four thousand images of coffee beans were used, each with blocks of between 25 and 36 beans, randomly distributed according to their state, but with a defined spatial order. The model was able to detect multiple beans per image and classify five categories: four types of morphological defects and the healthy bean. In terms of performance, an accuracy of 86,72 % was achieved during the training phase and 81,25 % in the validation stage. However, one of the main limitations of the study is that it was developed under highly controlled conditions, using specialized microscope cameras and RGB lighting, which may restrict its applicability in real production scenarios.

In Gope et al.(17), different versions of YOLO (v3 to v8) and a custom model (custom-YOLOv8n) were compared for defect detection in green coffee beans, using a dataset of 5,044 images with multiple defect classes. Unlike other approaches, this model performed bean-by-bean detection (one bean per image) and classified five categories: four defects and the healthy bean. Among the evaluated models, custom-YOLOv8n achieved the best performance, with a mean Average Precision (mAP) of 0,995, a precision of 0,987, a recall of 0,990, and an overall accuracy of 0,952. However, despite its high performance, the study identified several important limitations, including the lengthy training time and the reliance on a dataset with limited diversity, which could compromise the model's generalization ability in real-world production environments.

In Arwatchananukul et al.(18) used convolutional neural networks to classify 17 types of defects in green Arabica coffee beans in Thailand, using a dataset augmented from 979 to 6853 images. The MobileNetV3 model stood out for its performance, achieving an accuracy of 90,19 % and an Accuracy of 86,38 %. The model was implemented in a web application for real-time classification. A limitation is the low diversity of the dataset, as it detects only one bean per image, which affects its generalization capacity in real-world conditions.

Manansala et al.(19) developed an automatic defect detection system in coffee beans using Slim-CNN, YOLOv5, and VGG-16. They used a dataset of 3040 images and over 25 000 annotations, classifying five types of physical defects. YOLOv5 obtained the best performance with an accuracy of 98,52 %, followed by VGG-16 and Slim-CNN. The system was implemented on a Raspberry Pi 4 with a camera and screen, demonstrating its applicability in the field. The main limitation was the confusion between similar classes, highlighting the need to improve labeling and expand the dataset.

Nugroho et al.(20) developed a YOLOv5-based Android mobile application to detect defects in robusta coffee beans in real-time, replacing inefficient manual methods. The model was trained with 1860 images distributed across four defect classes and was able to detect multiple coffee beans simultaneously. In terms of performance, the model achieved an accuracy of 95,3 % in detecting black beans; however, it performed significantly worse in detecting moldy beans (62,2 %), which affected the overall accuracy, which stood at 78,3 %. This decrease in performance is attributed to the visual similarity between some defects and healthy beans. The main limitations of the study include the absence of data augmentation techniques during training and the model's difficulty in distinguishing subtle morphological defects.

https://doi.org/10.56294/dm2025840 ISSN: 2953-4917

Prabu et al.[21] proposed an intelligent system that combines Super-Resolution GAN (SR-GAN) and CNNs (not defined), to improve the detection of defects in coffee beans, increasing the sharpness of the images before classifying them. The approach enabled the identification of visual defects that are often overlooked in low-quality images, detected more than one bean per image, and classified them into four categories, achieving an accuracy of 82,4 %. However, its performance depends on the initial images being of acceptable quality and has not been validated in real-world environments, which limits its practical generalization.

Thai et al.[22] developed a real-time application for automatic defect classification in coffee beans using YOLOv8 and the OpenCV library. A dataset including nine defect types and data augmentation techniques was used. The system achieved an overall accuracy of 38 %, being most effective for obvious defects. Although the model is promising for industrial applications, its accuracy varies depending on the defect type, with the most challenging being the broken type and the most effective being the heavy type.

Based on the literature, there is a need to continue proposing more accurate and robust models, especially those addressing multiple coffee beans in the same image, which are randomly distributed and overlapping, as would be the natural configuration in a real coffee bean classification system. This is precisely what this study aims to address.

METHOD

In this work, the PyTorch object detection API was used to experiment with the YOLOv11 model in its two variants, Nano and Small. A custom dataset of 4,090 RGB images containing multiple coffee beans was collected, which were manually labeled into good and bad classes with the help of an expert using the online annotation application CVAT.[23]

The analysis and development process is based on our previous work[6] and consists of five sequential phases: (i) Characterization of coffee bean defects, (ii) Image acquisition, (iii) Image preprocessing and labelling, (iv) Model training, and (v) Model validation. Each phase is detailed below.

Characterization of coffee bean defects

The various defects present in coffee beans are classified as defective (bad) and healthy (good) beans with the help of an expert. A single seed may present several morphological defects, as shown in figure 1.

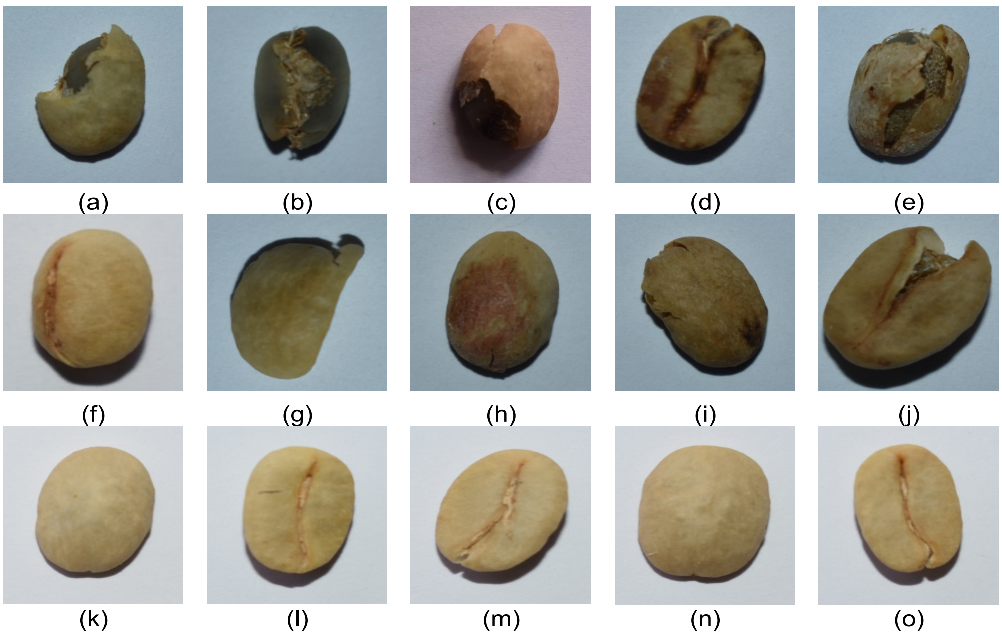


Figure 1. Examples of Coffee bean defects: (a) cuts, (b) broken, (c) pulp or husk attached, (d) humidity damage, e) insect damage, (f) Peaberry malformation, (g) empty bean, (h, i) fungi damage, (j) sudden temperature change, (k-o) healthy beans without flaws[6]

Image acquisition

A total of 4090 new RGB images containing multiple healthy and defective grains were collected, with a distribution of 56,48 % and 43,52 %, respectively. The images were taken with a Canon SX50 HS digital camera, with a 12,1 MP CMOS sensor, 50x optical zoom and stabilization, mounted on a tripod at a fixed distance of 14 cm from the object, against a white background (figure 2). For this study, the camera's built-in lens was used, with a focal length of 55 mm and an aperture of f/5,6. These values were determined through trial and error to obtain sharp images that clearly show the defects present in the coffee beans. The images have different resolutions to account for variability in data acquisition. The image capture conditions feature natural lighting, with no predetermined orientation of the coffee beans or specific order or alignment in their arrangement. Furthermore, the number of beans per image varies, without following a fixed pattern, which introduces a more realistic and practical scenario in data acquisition, unlike other existing studies.(16,21)

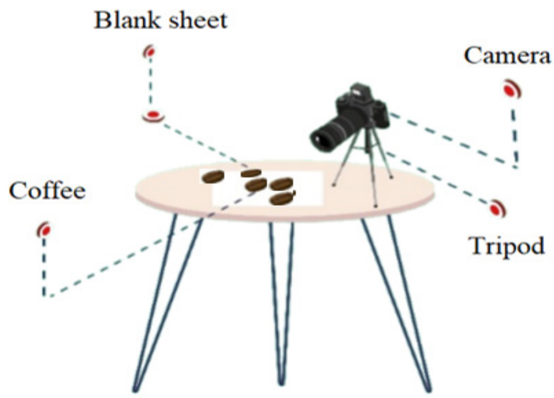


Figure 2. Imaging multiple coffee beans in a controlled lighting environment(6)

Image preprocessing and labelling

The acquired images were cropped using the GIMP software at a resolution of 902×688 pixels to standardize the image size, which contains the region of interest—the coffee beans—and to eliminate foreign objects from the frame. The images were then labeled using the online service CVAT.(23,24) After validation by a coffee quality expert, the images were classified using two labels, "good" and "bad" (binary classification). After completing the image labeling, they were exported in the Pascal VOC format, which generates both images and associated XML files. These files contain detailed information about the images, including name, dimensions, and the bounding box coordinates of the multiple coffee beans present, along with their respective labels (good, bad). This format was chosen due to its compatibility with YOLOv11 and the tools used in the preprocessing stage.

To enhance the diversity and representativeness of the dataset, 2,954 RGB images from our previous study(6), were randomly added, each containing a single coffee bean. These images added variability in visual and contextual features, enriching the original and augmented datasets. Thanks to this integration, the final dataset consisted of 7044 images, covering a broader spectrum of morphological variations in coffee beans, which enhances the model's generalization capabilities.

For the data augmentation process, the Roboflow platform(25) was used, which allowed for a significant increase in the size of the training dataset and improved the model's generalization capacity by introducing variations in the images. Specifically, for each original image, three augmented images were generated by applying random transformations, such as: random crops (crop) with a zoom of up to 20 %, inclined cuts (shear) with angles of up to ±15° to simulate different perspectives, conversion to grayscale in 15 % of the images to reduce color dependence, and tone adjustments (hue) in a range of -50° to +50°, to emulate various lighting conditions. These augmentation techniques were applied exclusively to the training set, while the validation and test sets remained unchanged. Two versions of the dataset were created to train the YOLOv11 model and improve its robustness:

Figure 3. Example images from dataset v1, containing coffee beans with a white background

Dataset version 2: to build the second version of the dataset, we started from the original set of 7,044 images, of which 686 were randomly selected to manually modify their white background using Adobe

https://doi.org/10.56294/dm2025840 ISSN: 2953-4917

Photoshop[26], to simulate more varied and realistic scenarios (black and gray background). These images were manually relabeled based on information previously generated during the first labeling process, which allowed for consistency in the annotations. As a result, this version consisted of a total of 7730 images. Overall, the second version includes a total of 18,548 images generated through data augmentation, of which 15 % have black and gray backgrounds, while the rest maintain a white background. This variation in backgrounds sought to increase the representativeness of the set and evaluate the model's ability to adapt to different lighting and contrast conditions (figure 4). The split of this dataset was 80/10/10 for training, validation, and testing.

Figure 4. Example images from dataset v2, containing coffee beans with variable black (left) and gray (right) backgrounds

Table 1 summarizes the two versions of the dataset created for training the model.

Table 1. Comparison of the two versions of the coffee bean dataset

	Version 1	Version 2
Number of images	16916	18548
Background	white	white, gray, black

Model Training

The PyTorch object detection API was chosen for its support of various models, including YOLOv11, and its easy integration with the Ultralytics library, which facilitates visual analysis of results.

YOLOv11 is an enhanced version of the YOLO (You Only Look Once) family, focusing on maintaining high inference speed while increasing accuracy. Its main improvement over previous versions lies in the use of an optimized architecture with a lightweight backbone, efficient blocks, and a refined prediction head for multi-scale detection. It also incorporates more robust training techniques (such as advanced data augmentation and dynamic anchor assignment), achieving a better balance between accuracy and performance on low-power or real-time devices.[27] Ultralytics has launched a complete family of YOLOv11 models with different sizes and usage modes, summarized below:

- YOLO11n (nano) – Ultra light.
- YOLO11s (small) – Fast and efficient.
- YOLO11m (medium) – Good speed/precision balance.
- YOLO11l (large) – High precision.
- YOLO11x (extra-large) – Maximum precision.

In this study, it was decided to experiment with the two most efficient versions (nano and small), with a view to their implementation in edge devices.

- YOLOv11-Nano: it is a light and fast version, ideal for environments with limited hardware, offering good accuracy to detect defects in coffee beans, which surpasses previous versions in performance.[28]
- YOLOv11-Small: it offers better accuracy than the Nano version, while maintaining efficiency in speed and size. It is suitable for computers with greater computing power.

Several key hyperparameters of YOLOv11 (considered for lightness, speed, and lower latency) were carefully tuned and configured using trial-and-error methodology, which allowed for better use of available resources and greater efficiency in model learning. They are described next:

- Data: the path to the YAML file was specified, which contains essential information about the classes and distribution of the images.
- Batch size: it was set to 16, aiming for a good balance between training speed and GPU memory usage.
- Epochs: fifty epochs were configured, allowing the model enough time to adjust its weights and minimize the loss. Additionally, the early stopping option (patience=10) was used to stop training if the model's performance did not improve.

- Imgsz: images were resized to 640×640×3 pixels to fit the YOLOv11 model, maximizing resolution without compromising computational performance.
- Plots: the option to save training graphs was enabled, allowing the model's progress to be monitored in real time using visual metrics such as accuracy and loss.
- Optimizer: the AdamW optimizer, widely recognized for its effectiveness in computer vision tasks, was used.
- Workers: finally, four worker threads were configured for data loading, which significantly accelerated the training process by reducing bottlenecks in reading the dataset.

Table 2 summarizes the hyperparameters used in the training of the YOLOv11 model (nano and small).

Table 2. Summary of hyperparameters used in training the YOLOv11 model

Hyperparameter	Value
Dataset format	YOLO
Framework	PyTorch + Ultralytics
Optimizer	AdamW
Epochs	50
Batch size	16
Callback	Early Stopping (patience = 10)
Image size	640×640×3
Evaluation metrics	Accuracy, precision, recall, mAP, model size, training time

Four specific models were trained, each with different configurations, to compare their performance under various conditions: YOLOv11-Nano with datasets 1 and 2 (white and variable backgrounds) and YOLOv11-Small with datasets 1 and 2 (white and variable backgrounds).

Model validation

The evaluation metrics for coffee bean classification were mainly: Accuracy, Precision, Recall, F1-score[29,30], as well as model size (MB), training time (hours), inference speed (FPS), and the precision-confidence curve. They are detailed in the next section.

RESULTS

Quantitative metrics

Below are the confusion matrices of the four models trained for coffee bean classification: YOLOv11-Nano with white background images (table 3), YOLOv11-Nano with variable background (table 4), YOLOv11-Small with white background (table 5), and YOLOv11-Small with variable background (table 6). We will assign the positive class (1) to the examples we wish to identify, in this case, to the good beans, while the negative class (0) to the bad beans.

Table 3. YOLOv11-Nano confusion matrix with white background in the test set (dataset v1)

Predicted/Actual	Bad (0)	Good (1)
Bad (0)	1149	174
Good (1)	203	1631

Table 4. YOLOv11-Nano confusion matrix with background variables in the test set (v2)

Predicted/Actual	Bad (0)	Good (1)
Bad (0)	1016	203
Good (1)	168	1346

From the confusion matrices, the accuracy, precision, recall, and F1-Score metrics were calculated using equations (1-4), respectively, where TP represents true positives, FP stands for false positives, TN means true negatives, and FN represents false negatives. The results of these metrics are shown in table 7 for the YOLOv11 model versions (Nano and Small) using datasets 1 and 2 with white backgrounds and variable backgrounds, respectively.

Table 5. YOLOv11-Small confusion matrix with white background in the test set (dataset v1)		
Predicted/Actual	**Bad (0)**	**Good (1)**
Bad (0)	1115	171
Good (1)	237	1634

Table 6. YOLOv11-Small confusion matrix with background variables in the test set (dataset v2)		
Predicted/Actual	**Bad (0)**	**Good (1)**
Bad (0)	1010	182
Good (1)	176	1367

Accuracy = (TP+TN)/(TP + FP+TN+FN) (1)

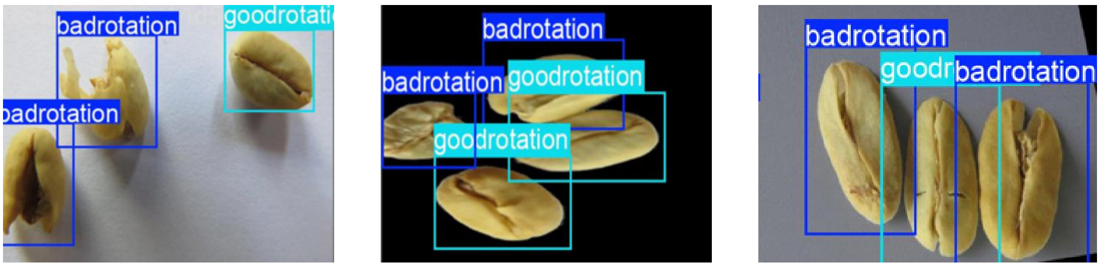
Precision = TP/(TP + FP) (2)

Recall = TP/(TP + FN) (3)

F1-score = (2·Recall·Precision)/(Recall+Precision) (4)

Table 7. Summary of the classification metrics calculated for the four models trained based on YOLOv11

Model	**Dataset**	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Model size (MB)**	**Training time (hours)**
YOLOv11-Nano	1 - White background	86,40	0,87	0,89	0,94	10,6	5:55
	2 - Variable background	88,03	0,85	0,91	0,95		6:10
YOLOv11-Small	1 - White background	86,90	0,85	0,90	0,95	38	7:45
	2 - Variable background	87,16	0,86	0,90	0,94		8:10



(a) (b) (c)

Figure 5. Example images of multiple coffee bean detection (bounding boxes) on a white background (a) and variable backgrounds: black (b) and gray (c).

Regarding the datasets used, the results showed higher accuracy in both versions of YOLOv11 (Nano and Small) when trained on Dataset 2, which included variable backgrounds in the images (white, black, and gray). The Nano version achieved 1,89 % greater accuracy, while the small version achieved only 0,30 %.

Regarding the two versions of YOLOv11, the Nano variant (10,6 MB) performed better, with 88,03 % accuracy, compared to the small version (38 MB), which achieved 87,16 % accuracy on dataset 2 (variable background). This represents a 1 % improvement in accuracy, achieved with fewer training parameters, suitable for embedded systems.

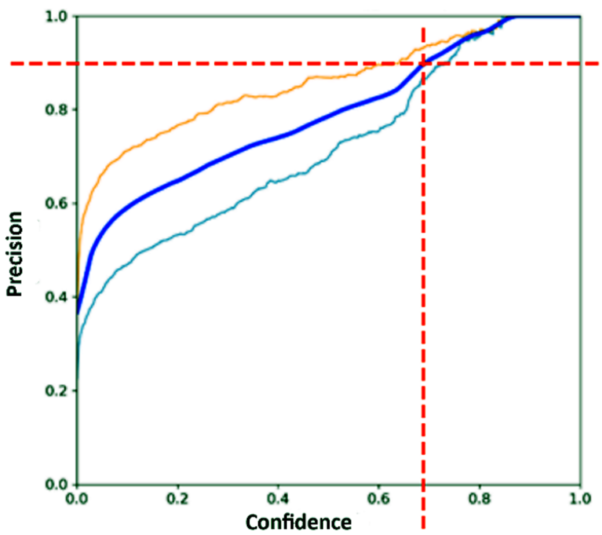
Some examples of detecting multiple coffee beans (bounding boxes) on various white and variable backgrounds using the best YOLOv11-Nano model are shown in figure 5.

Model performance graphs

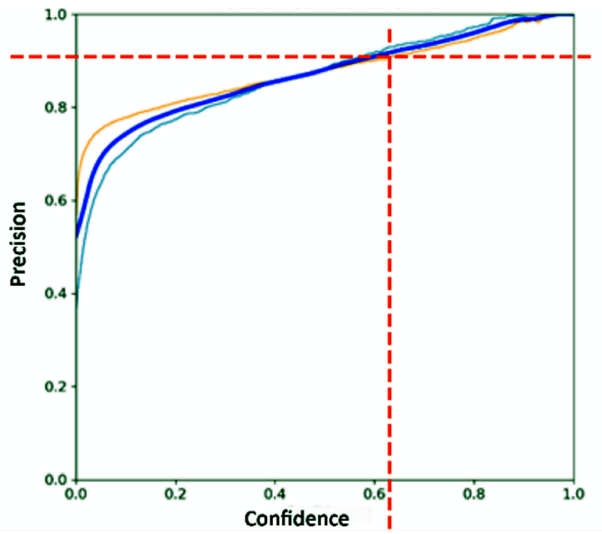
In figure 6 (a, b), the accuracy-confidence curves of the YOLOv11 nano and small models are shown on a variable background (dataset 2) during training, reaching the best accuracy values of 0,85 and 0,86, respectively.

The precision-confidence curve shows how the model's accuracy varies as the confidence threshold for accepting a prediction as valid is adjusted. This curve helps optimize classification reliability by selecting the optimal cutoff point based on model confidence. The cutoff points, where the precision and confidence lines intersect, indicate the minimum confidence threshold required for the model to maintain the desired accuracy. In the Nano version, the model should only accept predictions with at least 70 % confidence to ensure that 85 % of them are correct. In the Small version, the model should only accept predictions with at least 65 % confidence to ensure that 86 % of them are correct. This helps minimize FP errors (a bad bean classified as good) and FN errors (a good bean classified as bad), which affect the quality of the final product. These values (precision and confidence) show that the models have a high capacity to correctly identify defective beans, keeping the number of false positives low. This demonstrates good system reliability even under visual variations, such as different backgrounds.

The goal here is to determine the optimal confidence level that maintains high accuracy while minimizing critical errors in the automated classification process. If the confidence level is too low, the model may produce many false positives, thereby decreasing coffee quality. Conversely, if it is too high, it may increase the number of false negatives, wasting good beans (a valuable raw material).



(a)



(b)

Figure 6. Precision-confidence curves for YOLOv11 with a variable background (dataset 2) during training. Orange represents good kernels, light blue represents poor kernels, and blue represents the average. In (a), the Nano version, showing the cutoff threshold at precision=0,85 and confidence=0,7. In (b), the small version has cutoff thresholds at precision = 0,86 and confidence = 0,65

The Kaggle platform was chosen for model training because it offers an optimized environment for machine learning and computer vision projects. Kaggle provides free execution resources with an NVIDIA Tesla P100 GPU (16 GB VRAM), which was sufficient for training. According to Table 7, the YOLOv11-Nano version required approximately 6 hours, while YOLOv11-Small took about 8 hours. The models' sizes (MB) and inference speeds (fps) were 10,6 MB and 126,7 fps for the Nano variant and 38 MB and 99,20 fps for the Small variant.

DISCUSSION

In this study, we propose a lightweight YOLOv11-based model for coffee bean detection and classification, addressing two important limitations identified in our previous work[6], which are the detection of multiple coffee beans and the variable background (white, black, grey) of the image. It allows us to improve the model and adapt it to more realistic conditions. Table 8 compares the results obtained by the YOLOv11 nano and small models on a variable background (Dataset 2) with other similar works for coffee bean classification.

https://doi.org/10.56294/dm2025840 ISSN: 2953-4917

Table 8. Comparison of results (accuracy) with different existing works for coffee bean classification

Reference	Year	Model	Multiple detection	Number of classes	Accuracy (%)
Cevallos et al.[6]	2024	SSD-MobileNetV2	No (1 bean)	Binary (2)	0,9165
		SSD-ResNet50	No (1 bean)	Binary (2)	0,8307
Arwatchananukul et al.[18]	2024	MobileNetV3	No (1 bean)	Multiclass (17)	0,9019
Gope et al.[17]	2024	custom-YOLOv8n	No (1 bean)	Multiclass (5)	0,952
Gonzabay[16]	2023	MobileNetV1	Yes (> 1 beans)	Multiclass (5)	0,8125
Thai et al.[22]	2024	YOLOv8	Yes (> 1 beans)	Multiclass (9)	0,38
Manansala et al.[19]	2024	Slim-CNN	Yes (> 1 beans)	Multiclass (5)	0,936
		YOLOv5	Yes (> 1 beans)	Multiclass (5)	0,9852
		VGG-16	Yes (> 1 beans)	Multiclass (5)	0,968
Nugroho et al.[20]	2025	YOLOv5	Yes (> 1 beans)	Multiclass (4)	0,783
Prabu et al.[21]		SR-GAN and CNN	Yes (> 1 beans)	Multiclass (4)	0,824
Our work	2025	YOLOv11-Nano (variable background)	Yes (> 1 beans)	Binary (2)	0,880
		YOLOv11-Small (variable background)	Yes (> 1 beans)	Binary (2)	0,871

Regarding the works that consider a single coffee bean in the image, models such as SSD-MobileNetV2, SSD-ResNet50, MobileNetV3, and a customized version of YOLOv8n have been used. The best result is YOLOv8n with an accuracy of 95,2 %,[17] followed by SSD-MobileNetV2 with 91,65 %,[6] and MobileNetV3 with 90,19 %.[18] The worst performance was SSD-ResNet50 with 83,07 %.[6] These differences can be attributed to the type of architecture, the number of classes, and the quality of the dataset used. However, these approaches have a significant limitation: they only consider one coffee bean per image, which reduces their applicability in real-life contexts where multiple, randomly distributed beans are usually grouped/overlapping.

In contrast, regarding the works that consider multiple coffee beans in the image, the best result is obtained by YOLOv5[19] with an accuracy of 98,52 %, followed by VGG-16 (96,8 %) and Slim-CNN (93,6 %). Our model, YOLOv11-Nano, achieves 88 %, showing good performance in more realistic environments with variable background, outperforming several works, including MobileNetV1[16] with 81,25 %, YOLOv8[22] with only 38 %, YOLOv5[20] with 78,3 % and SR-GAN[21] with 82,4 %.

A common strategy in industrial systems, due to its ease of implementation and its usefulness in rapid automated selection decisions, is to prioritize models for binary classification, that is, determining whether a grain is good or defective. A model like the one proposed here (YOLOv11-Nano) has achieved an accuracy rate of 88 %, demonstrating that this approach is very suitable, even for real-time processing (126,7 fps). In contrast, some studies opt to categorize different types of morphological defects in a specific manner. This offers a more detailed and useful analysis for exhaustive quality control, but implies greater technical demands and controlled environments. In general, models that handle multiple classes in a complex real-life scenario tend to reduce their accuracy.

According to reports by [6], most subjects are efficient in classifying coffee beans during the morning hours, with a mean average accuracy of 91,1 %. However, they cannot maintain this level of precision during the afternoon hours, with a result of 66,7 %, resulting in a daily average of 78,9 %. Comparing the accuracy of manual selection by operators (78,9 %) with that of the automatic selection by the best model obtained (YOLOv11-Nano) at 88 %, we can see that the model performs better on average and is adequate for embedded systems.

Regarding the study's limitations, the dataset's images are limited to a variety of coffee beans, as they were captured in a controlled lighting environment with uniform illumination, which could affect the generalizability of the model results. Thus, the classifier is trained on a specific type of coffee bean (Arabica variety), and the model is tailored to that species and a particular camera/lighting configuration. Furthermore, this study experiments only with the YOLOv11 architecture, leaving aside other hybrid architectures (CNN + Transformers) that could benefit from attention mechanisms. Addressing these constraints may enable models to achieve higher accuracy in practical applications within real-world contexts.

CONCLUSIONS

This work presents the classification of morphological defects in coffee beans in a controlled environment using deep learning with Python and PyTorch. Two lightweight CNN models are trained to classify coffee beans as good or bad, showing promising results and achieving accuracies of 88 % for YOLOv11-Nano and 81,10 % for YOLOv11-Small. The best resulting model (Nano variant) outperforms the individual's average accuracy in manual classification (78,9 %), with a lightweight size of 10,6 MB and an inference speed of 126,7 FPS on a NVIDIA Tesla P100 GPU (16 GB VRAM), making it suitable for embedded systems. The models are trained on a custom image dataset containing 4090 manually labelled images using the CVAT tool. The images contain multiple coffee beans, overlapping and randomly located in each image, considering different backgrounds (white, black, and gray). Then, the dataset is expanded to 16916 in version 1 and to 18548 in version 2 using data augmentation techniques, thereby increasing the variety of training examples to enhance the model's performance in real-world conditions.

In future work, it is recommended to continue expanding the dataset with images captured under various conditions to enhance the model's generalization capacity. Training the model on a higher-performance hardware setup or utilizing cloud services would facilitate conducting more experiments to achieve higher accuracies, suitable for practical use in real-world scenarios. Exploring newer, hybrid (CNN + Transformers) or more specialized model architectures could have provided additional insights. To explore a multiclass classification of specific morphological defects. Finally, it is suggested to implement the model on low-cost embedded hardware using the trained lightweight version of YOLOv11-Nano.

BIBLIOGRAPHIC REFERENCES

1. Durán-Aranguren DD, Robledo S, Gomez-Restrepo E, Arboleda Valencia JW, Tarazona NA, Robledo DD;, et al. Scientometric Overview of Coffee By-Products and Their Applications. Molecules 2021, Vol 26, Page 7605. 2021;26(24):7605. https://doi.org/10.3390/MOLECULES26247605.

2. León-Serrano LA, Matailo-Pinta AM, Romero-Ramón AA, Portalanza-Chavarría CA. Ecuador: producción de banano, café y cacao por zonas y su impacto económico 2013-2016. Revista Científica UISRAEL. 2020;7(3):103-21. https://doi.org/10.35290/RCUI.V7N3.2020.324.

3. USDA. Ecuador: Coffee Annual | USDA Foreign Agricultural Service. 2023. https://www.fas.usda.gov/data/ecuador-coffee-annual-6

4. MAGAP. MAGAP ejecuta "Proyecto de Reactivación de la Caficultura Ecuatoriana" – Ministerio de Agricultura y Ganadería. 2020. https://www.agricultura.gob.ec/magap-ejecuta-proyecto-de-reactivacion-de-la-caficultura-ecuatoriana/

5. Otero Lastra GI. Análisis de la capacitación y asistencia técnica a los productores de café, caso de estudio el noroccidente de Quito 2014-2018. PUCE-Quito; 2019. https://repositorio.puce.edu.ec/handle/123456789/34090

6. Cevallos M, Sandoval-Pillajo L, Caranqui-Sánchez V, Ortega-Bustamante C, Pusdá-Chulde M, García-Santillán I. Morphological Defects Classification in Coffee Beans Based on Convolutional Neural Networks. In: Valencia-García R, Borodulina T, Del Cioppo-Morstadt J, Moran-Castro CE, Vera-Lucio N, editors. Technologies and Innovation. Cham: Springer Nature Switzerland; 2024. p. 3-15. https://doi.org/10.1007/978-3-031-75702-0_1.

7. Ariza-Sentís M, Vélez S, Martínez-Peña R, Baja H, Valente J. Object detection and tracking in Precision Farming: a systematic review. Comput Electron Agric. 2024;219:108757. https://doi.org/10.1016/J.COMPAG.2024.108757.

8. Pusdá-Chulde MR, Salazar-Fierro FA, Sandoval-Pillajo L, Herrera-Granda EP, García-Santillán ID, De Giusti A. Image Analysis Based on Heterogeneous Architectures for Precision Agriculture: A Systematic Literature Review. In 2020. p. 51–70. https://doi.org/10.1007/978-3-030-33614-1_4.

9. Vinueza K, Sandoval-Pillajo L, Giret-Boggino A, Trejo-España D, Pusdá-Chulde M, García-Santillán I. Automatic weed quantification in potato crops based on a modified convolutional neural network using drone images. Data and Metadata. 2025 Feb 13;4(194). https://doi.org/10.56294/dm2025194.

10. Sandoval-Pillajo L, García-Santillán I, Pusdá-Chulde M, Giret A. Weed Detection Based on Deep Learning from UAV Imagery: A Review. Smart Agricultural Technology. 2025. https://doi.org/10.1016/j.atech.2025.101147.

11. Chacua B, Garcia I, Rosero P, Suarez L, Ramirez I, Simbana Z, et al. People Identification through Facial Recognition using Deep Learning. 2019 IEEE Latin American Conference on Computational Intelligence, LA-CCI 2019. 2019 Nov 1. https://doi.org/10.1109/LA-CCI47412.2019.9037043.

12. Montenegro S, Pusdá-Chulde M, Caranqui-Sánchez V, Herrera-Tapia J, Ortega-Bustamante C, García-Santillán I. Android Mobile Application for Cattle Body Condition Score Using Convolutional Neural Networks. In: Narváez FR, Urgilés F, Bastos-Filho TF, Salgado-Guerrero JP, editors. Smart Technologies, Systems and Applications. Cham: Springer Nature Switzerland; 2023. p. 91–105. https://doi.org/10.1007/978-3-031-32213-6_7.

13. Ulloa F, Sandoval-Pillajo L, Landeta-López P, Granda-Peñafiel N, Pusdá-Chulde M, García-Santillán I. Identification of Diabetic Retinopathy from Retinography Images Using a Convolutional Neural Network. 2025;121–36. https://doi.org/10.1007/978-3-031-75702-0_10.

14. Salazar-Fierro F, Cumbal C, Trejo-España D, León-Fernández C, Pusdá-Chulde M, García-Santillán I. Detection of Scoliosis in X-Ray Images Using a Convolutional Neural Network. 2025 [cited 2025 Jan 22];167–83. https://doi.org/10.1007/978-3-031-75702-0_13.

15. Guaichico E, Pusdá-Chulde M, Ortega-Bustamante M, Granda P, García-Santillán I. Mobile app for real-time academic attendance registration based on MobileFaceNet Convolutional neural network. Data and Metadata. 2025 Feb 13;4:193. https://doi.org/10.56294/dm2025193.

16. Gonzabay Jiménez EF. Procesamiento de imágenes para la clasificación de granos de café por forma y color. 2023 [cited 2025 Mar 20]; http://repositorio.utmachala.edu.ec/handle/48000/21622

17. Gope HL, Fukai H, Ruhad FM, Barman S. Comparative analysis of YOLO models for green coffee bean detection and defect classification. Sci Rep. 2024;14(1). https://doi.org/10.1038/s41598-024-78598-7.

18. Arwatchananukul S, Xu D, Charoenkwan P, Aung Moon S, Saengrayap R. Implementing a deep learning model for defect classification in Thai Arabica green coffee beans. Smart Agricultural Technology. 2024;9. https://doi.org/10.1016/j.atech.2024.100680.

19. Manansala J, Paglinawan C. Classification of Coffea Liberica Quality Using Convolution Neural Networks (Slim-CNN, YOLOv5, and VGG-16). 2024. 1–6 p. https://doi.org/10.1109/ICCCNT61001.2024.10723931.

20. Nugroho ED, Verdiana M, Algifari MH, Afriansyah A, Firmansyah HB, Rizkita AK, et al. Development of YOLO-Based Mobile Application for Detection of Defect Types in Robusta Coffee Beans. Journal of Applied Informatics and Computing. 2025 Jan 16;9(1):153–60. https://doi.org/10.30871/jaic.v9i1.8886.

21. Prabu S, Thanigaivelu PS, Poonkodi S, Anand M. Enhanced Coffee Bean Defect Detection Using Super-Resolution GANs and Deep Learning. 2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking, CICTN 2025. 2025 Jan 1;827–32. https://doi.org/10.1109/CICTN64563.2025.10932332.

22. Thai HD, Ko HJ, Huh JH. Coffee Bean Defects Automatic Classification Realtime Application Adopting Deep Learning. IEEE Access. 2024 Jan 1;12:126503–17. https://doi.org/10.1109/ACCESS.2024.3452552.

23. Intel. CVAT: Computer Vision Annotation Tool. 2024. https://www.cvat.ai/

24. Lima PM, Reis S. Data Labeling tools for Computer Vision: a Review. 2022 Apr 1; https://run.unl.pt/handle/10362/135873

25. Roboflow. 2025. Roboflow: Annotate, Train, Deploy. https://roboflow.com/

26. Adobe Inc. Software. 2025. Adobe Photoshop. Version 25.0, Adobe. https://photoshop.adobe.com/discover?promoid=LCDWT9XV&lang=es&mv=other&mv2=tab

27. Khanam R, Hussain M. YOLOv11: An Overview of the Key Architectural Enhancements. 2024. https://arxiv.org/abs/2410.17725

28. Sharma A, Kumar V, Longchamps L. Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. Smart Agricultural Technology. 2024 Dec 1;9:100648. https://doi.org/10.1016/J.ATECH.2024.100648.

29. Aguagallo L, Salazar-Fierro F, García-Santillán J, Posso-Yépez M, Landeta-López P, García-Santillán I. Analysis of Student Performance Applying Data Mining Techniques in a Virtual Learning Environment. International Journal of Emerging Technologies in Learning (iJET). 2023 Jun 7;18(11):175–95. https://doi.org/10.3991/ijet.v18i11.37309.

30. Vila D, Cisneros S, Granda P, Ortega C, Posso-Yépez M, García-Santillán I. Detection of desertion patterns in university students using data mining techniques: A case study. Vol. 895, Communications in Computer and Information Science. 2019. 420–429 p. https://doi.org/10.1007/978-3-030-05532-5_31.

FINANCING
The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST
The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION
Conceptualization: Anderson Chamorro, Marco Pusdá-Chulde, Iván García-Santillán.
Formal analysis: Anderson Chamorro, Marco Pusdá-Chulde, Iván García-Santillán.
Research: Anderson Chamorro, Diego Trejo-España, Iván García-Santillán.
Methodology: Anderson Chamorro, Víctor Caranqui-Sánchez, Iván García-Santillán.
Project management: Iván García-Santillán.
Resources: Marco Pusdá-Chulde, Iván García-Santillán.
Software: Anderson Chamorro, Diego Trejo-España, Víctor Caranqui-Sánchez, Iván García-Santillán.
Supervision: Marco Pusdá-Chulde, Iván García-Santillán.
Validation: Anderson Chamorro, Marco Pusdá-Chulde, Iván García-Santillán.
Display: Anderson Chamorro, Diego Trejo-España, Víctor Caranqui-Sánchez.
Drafting - original draft: Anderson Chamorro, Diego Trejo-España, Víctor Caranqui-Sánchez, Marco Pusdá-Chulde, Iván García-Santillán.
Writing - proofreading and editing: Anderson Chamorro, Diego Trejo-España, Víctor Caranqui-Sánchez, Marco Pusdá-Chulde, Iván García-Santillán.