Detection of Coffee Bean Defects in Speciality Coffee Association Standards using YOLOv12

Hocwin Hebert[1], Derry Alamsyah[2]
[1,2] Multi Data Palembang University, Sumatera Selatan, Indonesia, 30113
e-mail: hocwinhebert_2226250078@mhs.mdp.ac.id[1], derry@mdp.ac.id [2]
Correspondence: hocwinhebert_2226250078@mhs.mdp.ac.id

Abstract: Coffee is a high value plantation commodity with a significant role in the global economy. Coffee consumption, reaching more than two billion cups per day, continues to increase global demand for coffee beans. To ensure quality and consumer acceptance, green coffee bean quality evaluation must follow consistent international standards. However, inspection is still carried out manually, making it time consuming and subjective. This study proposes coffee bean defect detection based on the Specialty Coffee Association (SCA) standard using YOLOv12. YOLOv12 addresses limitations of previous YOLO versions by integrating R-ELAN to improve training efficiency and reduce gradient loss, as well as Flash Attention to enhance focus on important regions in complex images. A total of 225 images were obtained through augmentation from 45 original samples captured using a smartphone camera under controlled indoor conditions, with each image representing 300 grams of Mandheling coffee beans. The dataset was divided into training (80%), validation (10%), and testing (10%). Eight experimental configurations were evaluated using variations in initial learning rate (0.001 and 0.0005), batch size (8 and 16), and epochs (100 and 150). The optimal configuration an initial learning rate of 0.0005, batch size of 16, and 150 epochs achieved a precision of 87%, recall of 85%, and mAP_{50} of 84%. These results indicate that the effectiveness of YOLOv12 in detecting coffee bean defects depends on proper hyperparameter tuning. The model performs well on visually prominent defects such as Cherry Pods, but shows reduced performance on subtle defects including floater, fungus damage, and slight insect damage.

Keywords: Object Detection, YOLOv12, Deep Learning, SCA, Coffee Bean Defects.

1. Introduction

Coffee is a plantation commodity that significantly impacts the global economy, particularly in coffee producing countries [1]. Besides being a plantation commodity, coffee is a beverage enjoyed by many, with over two billion cups consumed worldwide daily [2]. This high level of consumption indirectly drives market demand for coffee beans, with demand continuing to increase year after year [3]. As a high value commodity, evaluating green bean quality is essential for storage stability and consumer acceptance, requiring consistent standards in line with international guidelines [4]. The Specialty Coffee Association (SCA) is an organization that sets standards governing the quality and types of coffee bean defects that affect taste. Based on the SCA, quality is divided into 5 classes, and divides the types of coffee bean defects into 16. Although the SCA provides detailed guidelines on the quality and types of coffee bean defects, in practice, quality evaluation is still closely related to traditional processes. This activity is carried out through manual sorting by trained personnel, which involves identifying and counting defective beans to determine the overall quality of a given sample. This process is time consuming and subjective, as the assessment is based on human perception, leading to inconsistencies [4]. Previously, there was research related to coffee beans, such as that conducted by Hira Lal Gope, who conducted defect detection by applying the YOLO architecture from YOLOv3 to Custom-YOLOv8n. The results obtained were that the Custom-YOLOv8n architecture had the best

performance with precision, recall, F1-Score, and mAP of 97.7%, 99%, 98.3%, and 99.5%, respectively [5]. Despite obtaining high accuracy results, the study still had limitations, such as limited coverage of defect types and only performing a single detection per seed. In addition, there is research conducted by Ji, where related research conducted coffee bean defect detection using YOLOv8n modified by combining WIoUv3 and the development of the Atn-C3Ghost module, which integrates Efficient Channel Attention (ECA). In this study only used 4 categories, namely normal beans, shell containing beans, incomplete beans, and discolored beans, and the results of this study obtained an accuracy of 99% in detecting green bean defects in coffee beans [6]. Furthermore, there is a recent research study conducted by Ibadurrahman Syahid trying to apply multiple detection to coffee beans using a dataset that refers to the SCA (Specialty Coffee Association) standard, covering grade 1, grade 2, and grade 3. In this study using the YOLOv8 method and obtained average results for precision, recall, mAP_{50} respectively 27.4%, 12.8%, 13.5% for 5-fold cross validation [7]. However, such studies are still limited in number, so data coverage and accuracy need to be improved.

Based on literature studies that have been conducted on various existing cases and problems, various previous studies have carried out various optimizations of defect detection by utilizing deep learning based image processing technology, but green bean defect detection based on the Specialty Coffee Association (SCA) standard rules has not yet obtained optimal results. These suboptimal results are thought to be due to the use of a grade 1 dataset, which has significantly fewer defect variations than normal field conditions, resulting in decreased YOLO accuracy [8]. In addition, architecturally, YOLOv8, which uses an Efficient Layer Aggregation Network (ELAN), also has several limitations. The ELAN structure tends to cause gradient blocking and lack of direct information flow between layers due to the lack of residual connections, thus hindering the learning process and making it difficult to achieve stable convergence during training [9], [10].

This limitation becomes even more significant in dense and complex detection scenarios, such as coffee bean defect detection, where many small objects with similar visual characteristics must be detected simultaneously in a single image. This condition is reflected in Syahid's research which applied YOLOv8 to SCA based coffee bean defect detection and reported precision, recall, and mAP_{50} values of 27.4%, 12.8%, and 13.5%, respectively [7]. To address this issue, the YOLOv12 architecture integrates R-ELAN and Flash Attention. R-ELAN combines residual learning with feature aggregation to improve training stability and feature representation quality, which are crucial in detecting small objects. In addition, Flash Attention enables the model to better focus on important areas in complex images by suppressing background noise [11], thereby improving defect localization capabilities while maintaining computational efficiency for real-time detection. Therefore, the research problem discussed in this study is how to implement the YOLOv12 architecture on grade 2, 3, and 4 coffee bean datasets to optimize the coffee bean defect detection process. In addition, this research aims to develop a coffee bean defect detection system based on the Specialty Coffee Association (SCA) standard that covers three grade categories (grade 2, 3, 4) by utilizing the capabilities of the YOLOv12 model. Through this approach, the research is expected to be able to improve the accuracy of defect identification, optimize the efficiency of the detection process, and expand the scope of defect types that can be recognized in multiple object detection scenarios.

2. Methods
A. Data Description
In this study, the dataset consisted of coffee bean images obtained from the Narara Coffee online store. The samples were Mandheling coffee beans processed using a semi-washed method, with each sample weighing 300 grams in accordance with SCA assessment standards. The dataset covers three coffee quality grades (grades 2-4) and includes 15 classes of coffee bean defects defined by SCA rules, namely full black, full sour, dried cherry/pod, fungus damage, severe insect damage, partial black, partial sour, parchment/pergamino, floater, immature/unripe, withered, shell, broken/chipped/cut, hull/husk, and slight insect damage [7].

B. Research Methods
In this research, a method design was first carried out to ensure the research was more directed and focused. The stages of the research methodology can be seen in Figure 1.

Figure 1. Research Methodology Flowchart

1. Literature Study
This research begins with a literature review on green coffee bean defect detection based on SCA rules using the YOLOv12 method. The review reveals several important findings. Existing studies demonstrate promising accuracy in detecting simple or limited defect types (4–5 types), while research on multi-type defect detection remains limited and achieves less satisfactory performance. In addition, the adoption of standardized grading systems such as SCA is still limited, and recent studies have not yet utilized newer architectures like YOLOv12, which offer improved feature aggregation and attention mechanisms. Furthermore, most previous studies focus on grade 1 datasets, resulting in limited defect diversity. Therefore, this study addresses a clear research gap by applying YOLOv12 for multi-defect detection based on SCA standards on grade 2–4 datasets with more diverse defect types, providing a theoretical foundation and guiding the proposed methodological framework.

2. Dataset Collection
In this study, the dataset used is the image of coffee beans purchased through the online store from Narara Coffee. The coffee used is semi-washed Mandheling coffee beans. The coffee bean sample used is 300 grams according to SCA standards. The dataset consists of 3 grades of coffee, which are divided into grades 2 to 4, and also consists of 15 classes of coffee bean defects according to SCA rules, namely full black, full sour, dried cherry/pod, fungus damaged, severe insect damage, partial black, partial sour, parchment/pergamino, floater, immature/unripe, withered, shell, broken/chipped/cut, hull/husk, slight insect damage [7]. These types of defects can be seen in Figure 2. This research dataset consists of 45 images of coffee beans that have not gone through any augmentation or other pre-processing processes.

Figure 2. Coffee bean defects based on SCA standards [7]

Table 1. Details of the Dataset Used

Coffee Grade	Amount	Sample Image
2	15	
3	15	
4	15	

The assessment is determined based on the SCA calculation and weighting system. To ensure that the dataset truly contains defects, every 300 grams of coffee used as a sample will undergo a manual search, then annotation is carried out on the defective coffee beans found by drawing bounding boxes and assigning defect classes according to the SCA standards in Roboflow. To validate the dataset, the annotated samples are then validated by a trained SCA coffee validator from Kopi Koloni. The weighting system can be seen in Table 2, while the SCA calculation system is as follows [7]:

a. Specialty Grade (Grade 1) : Maximum 5 full defects per 300g, no category 1 defects, 5% screen size tolerance, at least one distinctive attribute, error free, no quakers, 9-13% moisture content.

b. Premium Grade (Grade 2) : Maximum 8 full defects per 300g, category 1 defects allowed, 5% screen size tolerance, at least one distinctive attribute, error free, a maximum of three quakers, 9-13% moisture content.

c. Exchange Grade (Grade 3) : 9-23 full defects/300g, 50% above screen size 15 (5% tolerance below 14), no cup defects, maximum five quakers, moisture 9-13%.

d. Below Standard Grade (Grade 4) : 24-86 full defects/300g.

e. Off Grade (Grade 5) : More than 86 full defects/300g.

Table 2. Defective Coffee Bean Weight [7]

Category 1 Defects		Category 2 Defects	
Defect	Full Defect Equivalents	Defect	Full Defect Equivalents
Full Black	1	Partial Black	3
Full Sour	1	Partial Sour	3
Dried Cherry/Pod	1	Parchment	5
Fungus Damage	1	Floater	5
Foreign Matter	1	Immature/Unripe	5
Severe Insect Damage	5	Withered	5
		Shell	5
		Broken/Chipped/Cut	5
		Hull/Husk	5
		Slight Insect Damage	10

Based on the table above, the SCA system classifies coffee bean defects into categories 1 and 2 based on their severity and impact on overall coffee quality. Category 1 includes severe defects that directly affect flavor quality, for example, one full black bean counts as one full defect. Conversely, category 2 consists of less severe defects that require multiple occurrences to be considered one full defect according to the SCA weighting system, for example, three partial black beans. This weighting approach ensures a standardized and balanced evaluation of green coffee quality, taking into account both the type and severity of defects [12]. The image capture was performed from a top view to clearly capture object details. Images were acquired using a POCO F5 camera with a white background inside a styrofoam box under controlled lighting to minimize shadows, with a fixed shooting distance of 21 cm. Each 300 gram sample was photographed individually according to its grade. The POCO F5 features a 64 MP wide, 8 MP ultra wide, and 2 MP macro sensor, capable of capturing images at a resolution of 4624×2600 pixels. The dataset acquisition setup is illustrated in Figure 3, and the resulting images were used as the primary data for the coffee bean defect detection process.

Figure 3. Dataset Acquisition Instrument

3. System Design

The system design phase aims to establish and organize the workflow for coffee bean defect detection research in the SCA standard to be developed. The overall system design workflow is depicted in Figure 4.
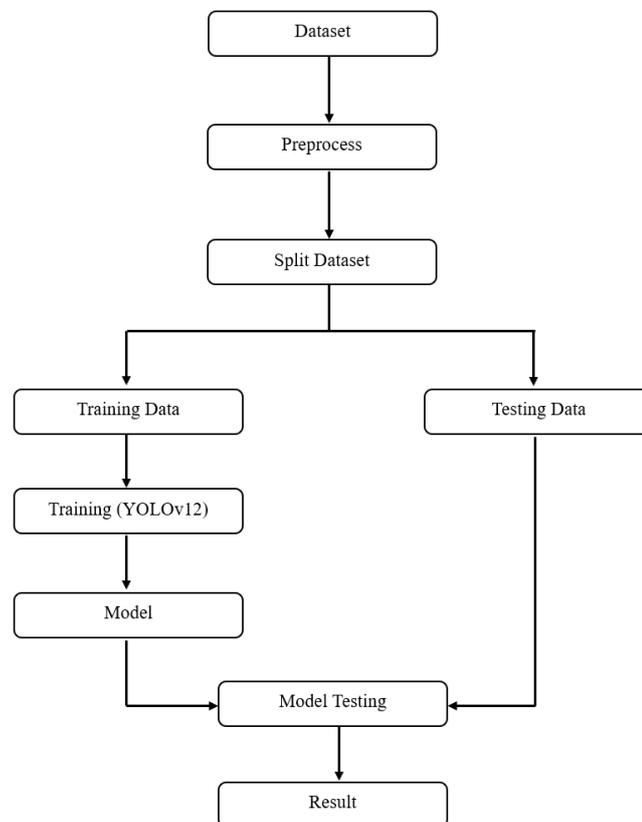
Figure 4. System Design Flowchart

After the dataset collection phase, a model was designed to detect green coffee bean defects using a supervised learning approach. Supervised learning is a type of machine learning where the model learns from labeled data. The data to be trained for this detection model consisted of coffee bean images manually annotated with Roboflow for 15 types of defects. Annotation was performed by providing a bounding box around the defective beans, then the defect was assigned a class according to the defects from the SCA example. During annotation, for overlapping objects, small defects, and visually subtle defects such as slight insect damage and broken, careful inspection was required to ensure accurate region marking. After the annotation process was completed, the image size was resized to 640×640 , and augmentation was performed in the form of rotate, flip, random brightness, and random gamma.

The dataset was initially divided into three subsets is 80% training data, 10% validation data, and 10% testing data. The testing set was separated and not involved in the training or cross-validation process, and was only used for the final performance evaluation. The training and validation data (90%) were used for *k*-Fold Cross Validation with *k* = 5, where in each iteration four parts were used for training and one part for validation. This process was repeated five times and the results were averaged to obtain a more stable and unbiased evaluation before final testing on an unseen dataset.

The model will then be tested in the testing phase on coffee bean images that have not been used in the training phase. Evaluation is performed using the mAP_{50} metric, which takes the average precision (AP) across all classes and calculates it at a predetermined Intersection over Union (IoU) threshold. In the IoU context, success is achieved when the overlap between the ground truth bounding box and the prediction is ≥ 50, while in the mAP_{50} context, success is achieved when it has a high mAP_{50} value. This indicates good detection performance. A model is considered successful when it consistently achieves high precision and recall across all classes, as indicated by an mAP_{50} score close to 1.0 (or 100%). Conversely, a lower mAP_{50} value implies misclassification or localization of defects. Therefore, the mAP_{50} score provides a quantitative benchmark to determine whether the trained YOLOv12 model effectively recognizes and localizes coffee defect types.

4. Implementation

At this stage, the system design is implemented according to the flowchart shown in Figure 4. The process begins with dataset collection, followed by annotation using RoboFlow, resizing, and data augmentation. The augmentation process includes horizontal and vertical flips, rotations at 90° (clockwise, counterclockwise, and reverse), additional rotations at -45° and +45°, as well as random brightness adjustment with a threshold of 0.3 and random gamma correction in the range of 80 - 120 to simulate lighting variations. After augmentation, the dataset is split into 80% training, 10% validation, and 10% testing. The training data is then used to train the YOLOv12 model, followed by *k*-fold cross-validation and hyperparameter tuning involving batch size, learning rate, and epoch to obtain the best and most stable configuration. Finally, the selected model is evaluated on an unseen test dataset using the mAP_{50} metric. The YOLOv12 model is implemented using the Ultralytics library, developed by the official YOLO team and built on the PyTorch framework. This library is widely used in object detection due to its support for training, deployment, flexible hyperparameter configuration, and efficient training and inference performance [13].

5. Testing and Evaluation

In this stage, the previously created model will be tested on the separated test data to evaluate its performance and accuracy in detecting coffee bean defects. After the testing phase, the test results are calculated to determine the success rate. The method used is mAP_{50}. The "50" in the mAP_{50}

metric refers to an IoU threshold of 0.50, meaning the prediction bounding box is considered correct only if the IoU with the ground-truth is ≥ 0.5. This threshold is used because it represents a reasonable level of overlap between the prediction and the ground truth. In the context of coffee bean defect detection, mAP_{50} is an appropriate metric because many defects, such as broken beans, slight insect damage, or other defects that have irregular shapes and small sizes, making it difficult to meet very strict thresholds such as ($mAP_{50\text{-}95}$). mAP_{50} works by taking the average precision (AP) of all classes and calculating it at a predetermined Intersection over Union (IoU) threshold. In calculating it, mAP_{50} requires precision and recall. The equations for precision, recall, AP, and mAP can be seen in Equations (1), (2), (3), and (4) [14].

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$AP = \int_0^1 P\,R\,dR \quad (3)$$

$$mAP50 = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (4)$$

In the evaluation process, predictions are classified into True Positive (TP), False Positive (FP), and False Negative (FN). This is determined based on the overlap of the bounding box between the ground truth and the prediction using the Intersection over Union (IoU) value. TP occurs when the prediction bounding box has sufficient overlap with the ground truth box, especially when the IoU value is ≥ 0.5. FP occurs when the prediction bounding box has an IoU < 0.5, so the overlap with the ground truth is inadequate. Meanwhile, FN occurs when there is ground truth but no prediction. In addition, AP is the average precision, where P in the formula refers to Precision, and R refers to Recall. Then, in AP, an integral will be performed, this is because we want to calculate the area of the P-R curve. Then because of the integral, there is a range, this range starts from 0 to 1. This is because the recall value ranges from 0 to 1 where a value of 0 indicates no ground truth objects were successfully detected and a value of 1 indicates all ground truth objects were correctly detected. Then there is the mAP formula where this is like a regular average formula that adds up the AP per class divided by the number of data (N). Since here there are 15 types of defects, then automatically the AP per class and the number of data (N) is 15.

In addition to testing with mAP_{50}, hyperparameter tuning was performed to identify the optimal model configuration. The tuning process involved varying the batch size (8 and 16), the number of training epochs (100 and 150), and the learning rate, with initial values of 0.001 and 0.0005 and a decay factor of 0.01. Test scenarios for combinations of batch size, epochs, and learning rate are shown in Table 3. This table is designed to demonstrate how varying batch size, epochs, and learning rate can affect model performance and stability. By systematically combining these hyperparameters, this study aims to identify which configurations yield the highest detection accuracy and the most stable convergence during training. Smaller batch sizes and lower initial learning rates are expected to improve gradient precision, while larger batches and higher learning rates can accelerate learning but risk instability. Similarly, comparing 100 and 150 epochs aims to analyze whether additional training time yields significant improvements or simply leads to overfitting. Through this scenario, it is hoped that we can find out which parameter combination is best for the model in detecting coffee bean defects.

Table 3. Model Testing Scenarios

No.	Batch Size	Epoch	Initial Learning Rate	Learning Rate Decay Factor
1.	8	100	0.001	0.01
2.	8	100	0.0005	0.01
3.	8	150	0.001	0.01
4.	8	150	0.0005	0.01
5.	16	100	0.001	0.01
6.	16	100	0.0005	0.01
7.	16	150	0.001	0.01
8.	16	150	0.0005	0.01

3. Results and Discussion

The performance results of the YOLOv12 model applied to coffee bean defect detection in the SCA standard using 640×640 pixel input images. In this study, the experiment explored various hyperparameter combinations. The hyperparameters used were the initial learning rate (0.001 and 0.0005), the final learning rate (0.01), the batch size (8 and 16), and the epochs (100 and 150). These parameters were chosen to observe whether the model requires longer training and how the model updates its weights for better generalization. In this context, the batch size refers to the set of sample data used to update the model weights, while the learning rate determines the magnitude of the weight change step at each iteration. Model performance was assessed using three main evaluation metrics: precision, recall, and mAP_{50}. Each precision, recall, and mAP_{50} value in this table is the average result of the k-Fold Cross Validation process for each hyperparameter scenario.

Based on the table, we can see that increasing the number of epochs from 100 to 150 results in higher precision, recall, and mAP_{50} results, for example, at an initial learning rate of 0.0005, and a batch size of 16, the mAP_{50} increases from 0.69 (at 100 epochs) to 0.84 (at 150 epochs). This shows that with longer training, the model is effective in learning the defect patterns in coffee beans. In addition to epochs, improvements occur when the batch size is increased from 8 to 16. In the table, we can take an example when the epoch is 150, the initial learning rate is 0.0005, the mAP_{50} increases from 0.78 (at a batch size of 8) to 0.84 (at a batch size of 16). This proves that a slightly larger batch size results in more stable gradients and more effective weight updates, so the model is able to generalize better as shown by the increase in mAP_{50}.

Additionally, Table 4 shows the use of initial learning rates of 0.001 and 0.0005, with a learning rate decay factor of 0.01. This learning rate decay factor multiplies the initial learning rate, gradually decreasing the learning rate, making the final learning rate much smaller than the initial value. All initial learning rates are multiplied by 0.01, so 0.001 becomes 0.00001 (0.001 * 0.01) and 0.0005 becomes 0.000005 (0.0005 * 0.01) [15]. In this mechanism, the main principle is to start with a relatively large learning rate, which prevents the model from memorizing noise too quickly and allows the model to focus on the simplest and most dominant patterns. Then, gradually decreasing the rate allows the model to move into a more stable exploitation phase to learn more complex patterns and refine convergence. Based on the table, we can see that using a smaller initial learning rate results in better model performance. For example, in batch 16 of 150 epochs, the mAP_{50} results increased from 0.71 (initial learning rate = 0.001) to 0.84 (initial learning rate = 0.0005). This indicates that a small initial learning rate allows the model to better capture subtle defect patterns in coffee beans and gradually refine its parameters, resulting in better convergence and generalization.

Compared to previous research by Syahid who applied YOLOv8 for SCA based coffee bean defect detection where the model achieved precision, recall, and mAP_{50} values of 27.4%, 12.8%, and 13.5%, respectively [7], the proposed YOLOv12 model shows an improvement on the model by achieving an mAP_{50} of 84%. This difference demonstrates the superiority of YOLOv12

architecture improvements, particularly R-ELAN and Flash Attention, which enhance feature aggregation and focus on relevant regions in dense object scenarios. To evaluate the ability of *k*-Fold which aims to prevent the model from overfitting and biasing, it can be seen in Table 4.

Table 4. *k*-Fold Evaluation (Mean ± Std)

Metric	Mean	Std	Mean ± Std
Precision	0.85	0.0083	0.85 ± 0.0083
Recall	0.80	0.016	0.80 ± 0.016
mAP_{50}	0.78	0.036	0.78 ± 0.036

In this table, the parameters are from batch 8, the initial learning rate is 0.0005, and the epoch is 150. Based on the table, we can see the results of the mean values of the precision, recall, mAP_{50} metrics and standard deviation are 0.85 ± 0.0083, 0.80 ± 0.016, and 0.78 ± 0.036, respectively. The relatively small standard deviation values across all metrics indicate that the model performance is consistent in each fold and is not biased towards certain data divisions. Furthermore, the high mean values of precision, recall, and mAP_{50} indicate that the model is able to effectively learn coffee bean defect patterns on various data subsets. The combination of high means and small standard deviations indicates that the model not only has good accuracy but also stability, so it has good generalization capabilities and does not experience overfitting. This performance consistency provides confidence that the performance improvement obtained is not due to advantageous data sharing, while the small variance value across folds serves as an implicit measure of confidence in the robustness of the model, reflecting stable learning behavior and reliable generalization ability.

Based on the existing parameter combinations, a configuration with a batch size of 16, an initial learning rate of 0.0005, and 150 epochs was selected for further analysis because it provided the best average *k*-Fold performance. Based on this configuration, the following discussion focuses on the mAP_{50} values per class to determine which defect types are easiest and hardest for the model to detect. The results of this test can be seen in Table 5.

Table 5. AP Per Class

Class	AP
Broken	0.45
Cherry Pods	0.89
Floater	0
Full Black	0.45
Full Sour	0.48
Fungus Damage	0.18
Hull	0.48
Immature	0.58
Parchment	0.50
Partial Black	0.33
Partial Sour	0.27
Severe Insect Damage	0.43
Shell	0.47
Slight Insect Damage	0.15
Withered	0.29

Based on Table 5, it can be seen that the model's ability to recognize each type of SCA coffee bean defect is significantly influenced by the visual characteristics of each class. The class with the highest performance is Cherry Pods (0.89) because this defect has clear, contrasting, and easily visible physical characteristics, such as a brownish color with a striking pod shape, allowing the

model to easily capture its visual pattern. In contrast, the classes with the lowest performance, such as Floater (0), Fungus Damage (0.18) and Slight Insect Damage (0.15), are much more difficult to detect. Slight Insect Damage only displays small black dots or very fine bite marks, so they are often hidden by the natural texture of the coffee bean or do not stand out when many beans are displayed in a single image. Similarly, fungus damage has a pattern of thin, irregular spots with a color similar to the grain surface, while floaters have a color and shape very similar to normal grains, making it difficult for the model to distinguish between defective features and visual disturbances. These findings suggest that subtle visual variations and small defect sizes are the main factors causing low mAP_{50} in some categories. In addition to visual complexity, dataset characteristics such as class imbalance and limited sample size certainly contribute to inter class performance variation. Rare and small defect types in the dataset tend to pose additional challenges to model learning. To deepen the analysis on the class with the best AP, an evaluation of the precision–recall curve (P–R Curve) was carried out on the Cherry Pods class as shown in Figure 5.
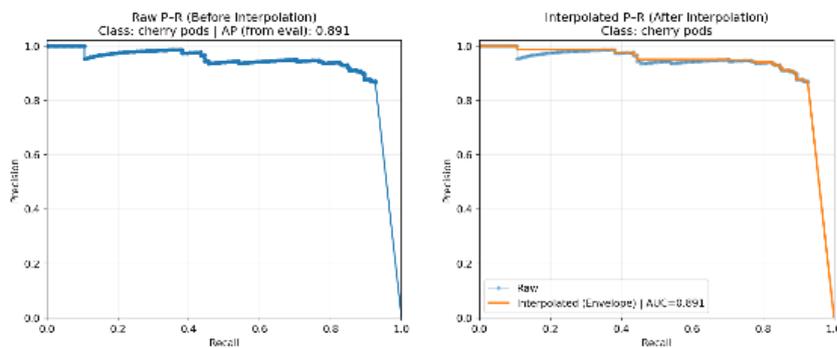

Figure 5. P-R Curve

In the figure, we can see that the Cherry Pods class has a high and relatively stable precision value across most of the recall range. A decrease in precision to 0 when recall approaches 1 is common due to the trade-off between detection coverage and prediction accuracy. A typical Precision Recall graph will have a zigzag pattern, as seen in the left plot. This Precision Recall will be interpolated to prevent the precision value from increasing with increasing recall, resulting in a more stable and representative curve. The interpolation results can be seen in the right plot. The area of the P-R Curve will then be calculated using an integral to obtain the AP class result. The Cherry Pods class obtained an AP of 0.89. This confirms that Cherry Pods defects have a visualization that is very easily recognized by the detection model.

4. Conclusion

Based on the experimental results, the effectiveness of the YOLOv12 method for detecting coffee bean defects according to the SCA standard across 15 defect types was demonstrated. Eight experimental configurations with different hyperparameter settings were evaluated to identify the optimal parameters. The best performance was achieved using a batch size of 16, an initial learning rate of 0.0005, and 150 epochs with a 640×640 image input, resulting in precision, recall, and mAP_{50} values of 87%, 85%, and 84%, respectively. These results highlight the importance of appropriate hyperparameter selection for CNN based defect detection, where the model performs well on defects with clear visual characteristics but faces challenges in detecting subtle defects such as floater, fungus damage, and slight insect damage. From a practical perspective, this approach can support coffee quality inspection by reducing reliance on manual sorting, improving inspection consistency, and accelerating the assessment process according to SCA standards. Such an automated system can help coffee producers and quality inspectors conduct more objective and efficient defect evaluations. Although the results obtained show promising performance, further research needs to focus on improving the robustness of the model through expanding the dataset

to reflect more diverse real world conditions, such as increasing the number of samples for rare and subtle defect types, using images with varying lighting and background conditions, and exploring improvements to the model architecture.

References

[1]   H. H. Nasution, "ANALISIS NILAI TAMBAH PENGOLAHAN KOPI ARABICA DI KOPERASI KOPI MANDAILING JAYA DESA ALAHANKAE KECAMATAN ULU PUNGKUT KABUPATEN MANDAILING NATAL," 2022. Accessed: Nov. 02, 2025. [Online]. Available: https://repositori.uma.ac.id/bitstream/123456789/19204/1/178220100%20-%20Hasmar%20Husein%20Nasution%20Fulltext.pdf

[2]   B. Čižmárová, V. Kraus, and A. Birková, "Caffeinated Beverages—Unveiling Their Impact on Human Health," Feb. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/beverages11010018.

[3]   A. Rahmawati, "DETEKSI DEFECT COFFE PADA CITRA TUNGGAL GREEN BEANS MENGGUNAKAN METODE ENSAMBLE DECISION TREE," 2020. Accessed: Nov. 02, 2025. [Online]. Available: https://publikasi.dinus.ac.id/index.php/technoc/article/view/4529

[4]   E. Oncu, "Deep Learning Framework for Coffee Quality Assessment via YOLOv8n Object Detection of Bean Defects," Sep. 09, 2025. doi: 10.20944/preprints202509.0643.v1.

[5]   H. L. Gope, H. Fukai, F. M. Ruhad, and S. Barman, "Comparative analysis of YOLO models for green coffee bean detection and defect classification," *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-78598-7.

[6]   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.

[7]   I. Syahid and E. Rachmawati, "Deteksi Cacat Biji Kopi Berdasarkan Spesifikasi Specialty Coffee Association dengan YOLOv8," 2025. Accessed: Nov. 02, 2025. [Online]. Available: https://share.google/D85zHdrG0BwDRDlyC

[8]   L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[9]   Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," *ArXiv*, 2025, doi: 10.0.

[10] R. Sapkota *et al.*, "YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series," Jun. 2025, doi: 10.1007/s10462-025-11253-3.

[11] M. A. R. Alif and M. Hussain, "YOLOv12: A Breakdown of the Key Architectural Features," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.14740

[12] Specialty Coffee Association, "Building an Understanding Based on Attributes Towards a Deeinition of Specialty Coffee: An SCA White Paper," 2021. Accessed: Dec. 14, 2025. [Online]. Available: https://sca.coffee/sca-news/just-released-new-sca-white-paper-towards-a-definition-of-specialty-coffee

[13] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Dec. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/make5040083.

[14] F. Lu *et al.*, "Object Detection of UAV Images from Orthographic Perspective Based on Improved YOLOv5s," *Sustainability (Switzerland)*, vol. 15, no. 19, Oct. 2023, doi: 10.3390/su151914564.

[15] J. A. P. Lopes, B. Baptista, N. Lavado, and M. Mendes, "Offline Handwritten Signature Verification Using Deep Neural Networks," *Energies (Basel)*, vol. 15, no. 20, Oct. 2022, doi: 10.3390/en15207611.