CENTIVE
Conference on Electrical Engineering, Informatics
, Industrial Technology, and Creative Media
ISBN : 978-602-53004-0-0

Integration of YOLOv11 and Convolutional Neural Network in a Deep Learning Approach for Coffee Bean Defect Detection and Classification

Rizal Adi Saputra[1], Fildzah Khalishah Ghassani[2*],
Fid Aksara[3], Isnawaty[4], La Ode Muhammad Golok Jaya[5],
Ishak Kadir[6], Rafi Iyad Madani Chaidir[7]
[1,2,3,4,5,6,7]Universitas Halu Oleo
[1,2,3]Kampus Bumi Tridharma Anduonohu, Jalan H.E.A. Mokodompit,
Kecamatan Mandonga, Kendari, Sulawesi Tenggara, Indonesia
[*]fkhalishah20@gmail.com

Abstract — The coffee industry is a strategic commodity that significantly contributes to the global and national economy. Coffee bean quality strongly influences flavor and market value, while defective beans—such as broken, moldy, or quaker beans—can reduce overall quality. Manual sorting methods, still widely used by farmers and small-scale producers, are time-consuming, inefficient, and prone to human error. This study proposes an automated deep learning–based system for detecting and classifying defective coffee beans by integrating YOLO for object detection and EfficientNetV2 as the classifier. A dataset of 5,636 coffee bean images from multiple sources was used. The system was evaluated through black box testing to ensure the functionality of the web interface and performance testing using a confusion matrix. Results show that YOLOv11 achieved an mAP@0.5 of 98.83%, while EfficientNetV2 obtained a test accuracy of 93.81%. The proposed system demonstrates strong potential to improve coffee sorting by providing a faster, more accurate, and efficient alternative to manual methods.

Keywords – You Only Look Once (YOLO), Convolutional Neural Network (CNN), Detection, Classification, Coffee Bean Defects

I. INTRODUCTION

The coffee industry in Indonesia continues to show rapid growth, not only at the national level but also across local regions. This development is marked by the increasing number of small and medium enterprises (SMEs) engaged in coffee cultivation, processing, and sales. These SMEs include farmers, processors, coffee shop owners, and coffee-based creative industries, particularly in Southeast Sulawesi, where regions such as Bombana, Konawe, and East Kolaka are emerging as new production centers [1].

In coffee processing, bean quality is a critical factor that determines flavor, aroma, and market competitiveness. Defective beans, such as broken, moldy, or underdeveloped (quaker) beans, can significantly reduce product quality, often resulting in undesirable taste and aroma profiles. For SMEs, poor quality caused by defective beans directly affects customer satisfaction, reduces market value, and weakens competitiveness.

One major challenge faced by coffee processors is the post-harvest sorting process, which separates marketable beans from defective ones [2]. In practice, sorting is still largely performed manually, relying heavily on human accuracy and consistency. This approach is prone to error due to fatigue and subjectivity, while also being time-consuming and labor-intensive, especially when production volumes increase.

To address these challenges, automated detection and classification technologies offer a promising solution. Deep learning approaches have been widely applied in prior studies using architectures such as ResNet-34, VGG-16 [3], AlexNet [4], Multi-Scale Feature Extraction [5], and YOLO [6]. Each method presents distinct advantages and limitations. YOLO (You Only Look Once), for instance, is known for its speed and efficiency in real-time object detection, making it highly suitable for coffee bean sorting [7]. However, earlier versions such as YOLOv3 and YOLOv4 struggled with small or overlapping objects, issues that have been improved in newer versions like YOLOv8 and beyond through architectural refinement and advanced multi-scale feature extraction [7], [8].

Meanwhile, Convolutional Neural Networks (CNNs) have demonstrated strong performance in image classification tasks due to their ability to extract complex visual features such as texture and shape [9]. Earlier models such as ResNet, VGG, and AlexNet achieved considerable success but suffered from drawbacks such as overfitting in shallow architectures or high computational demands in deeper networks. Recent advancements, including EfficientNetV2 [10] and MobileNetV3 [11], provide a balance by offering high accuracy with reduced computational costs [12].

Based on these findings, this research focuses on implementing a deep learning system that integrates YOLO for defect detection and CNN for defect classification of coffee beans. The proposed approach is expected to improve production quality, accelerate the sorting process, minimize human error, and support SME competitiveness. Furthermore, this study aligns with the United Nations Sustainable Development Goals (SDGs), particularly Goal 9 (Industry, Innovation, and Infrastructure) and Goal 12 (Responsible Consumption and Production).

II. RESEARCH METHOD

A. Research Workflow

The overall research workflow is presented in (Fig. 1).

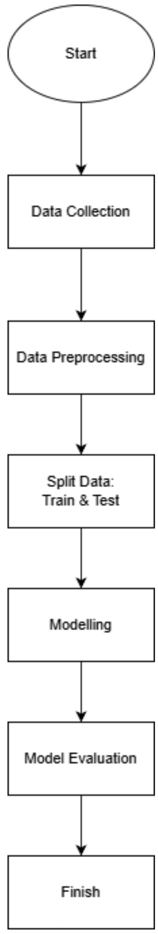


Fig. 1. Research Workflow.

a) Data Collection
The dataset was compiled from multiple sources, including coffee farmers in Southeast Sulawesi (Buton, Konawe, Bombana), coffee shops in Kendari (Mujur Coffee and Roaster, Tree Coffee), tidand a publicly available dataset from GitHub. In total, 5,636 images were collected, consisting of both normal and defective coffee beans.

b) Data Preprocessing
Data preprocessing is a crucial stage to prepare the dataset for deep learning models and to enhance data quality and variability. Common steps include normalization, resizing, labeling, and image augmentation. Proper preprocessing ensures that models can learn effectively and generalize well during training and testing.

In this research, all coffee bean images were resized to 640 × 640 pixels for YOLOv11 and 224 × 224 pixels for EfficientNetV2. Labeling was performed in YOLO format by assigning bounding boxes to each bean along with its class (normal/defective). To increase variability and reduce overfitting, augmentation techniques such as rotation, flipping, and brightness adjustment were applied. Furthermore, YOLO-detected beans were cropped based on bounding boxes and then used as inputs for classification with EfficientNetV2.

c) Data Splitting
Generally, dataset splitting is performed to ensure that models are trained effectively and evaluated on unseen data. The dataset is commonly divided into three subsets: training (for model learning), validation (for hyperparameter tuning), and testing (for final performance evaluation).

In this study, the dataset was divided into 70% for training, 15% for validation, and 15% for testing. The splitting was conducted using a stratified approach to preserve the balance between normal and defective coffee bean classes across all subsets.

d) Modelling
The modeling stage integrates object detection and image classification to identify coffee bean defects. YOLOv11 was implemented for object detection, enabling the localization of both normal and defective beans within images. Each

detected region of interest was cropped and used as input for the classification stage.

For classification, four CNN architectures were evaluated: EfficientNetV2, ResNet, DenseNet, and MobileNetV3. Each model was trained on the cropped datasets to compare their performance in recognizing defective versus normal beans. The evaluation included accuracy, loss, and other performance metrics. The best-performing CNN model was subsequently selected and integrated with YOLOv11 to form the final end-to-end detection and classification system.

e) Model Evaluation
The evaluation phase was conducted to measure both detection and classification performance. For object detection using YOLOv11, the metrics included mean Average Precision (mAP@0.5), Precision, Recall, and F1-score to assess bounding box accuracy and defect localization. For classification models (EfficientNetV2, ResNet, DenseNet, and MobileNetV3), evaluation was carried out using a confusion matrix, with performance metrics such as accuracy, precision, recall, and F1-score.

In addition, black-box testing was employed to validate the functionality and usability of the web-based system, ensuring that model integration with the user interface operated as intended.

B. Modeling Workflow

The modeling workflow is presented in (Fig. 2

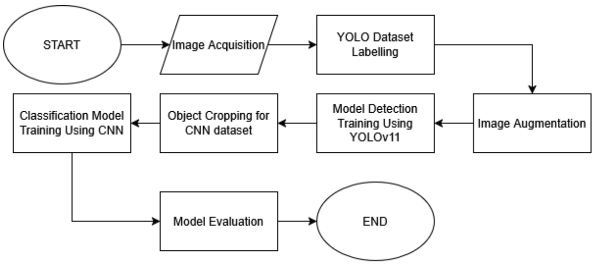


Fig. 2. Modeling Workflow.

The following is a detailed explanation of each step involved:

1) **Start:**This marks the beginning of the system process. This stage serves as a starting point without any data processing occurring at this initial phase.
2) **Image Acquisition:** Coffee bean images are obtained either through a camera or from a pre-collected dataset. These images serve as the primary input for further processing. Image quality is crucial as it directly impacts the accuracy of detection and classification. When images are captured using a camera, careful attention must be given to proper lighting and focus to ensure optimal image clarity.
3) **Dataset Labeling:** Each coffee bean within the images is labeled according to its defect category, such as normal, moldy, broken, or quaker. These labels act as the ground truth used during the training of detection and classification models.
4) **Image Augmentation:** To enhance data variability and mitigate the risk of overfitting, augmentation techniques are applied to the dataset. These techniques include rotation, horizontal/vertical flipping, adjustments in brightness and contrast, zooming, blurring, and adding noise. The aim is to make the model more generalizable and robust to varied real-world image conditions.
5) **Detection Model Training Using YOLOv11:** At this stage, the YOLOv11 model is trained to detect the presence and location of coffee beans within the images and classify them as either normal or defective. The model learns to identify coffee beans based on the labeled training data.
6) **Object Cropping for CNN Dataset:** Following object detection by YOLOv11, the coordinates of the predicted bounding boxes are used to crop the images. These cropped images then serve as inputs to the CNN model. Cropping allows the system to focus on individual coffee beans, thereby improving classification accuracy.
7) **Classification Model Training Using CNN:** The CNN model is trained to classify the detected coffee beans into their respective defect categories. This step aims to evaluate and compare the classification performance across the processed images.
8) **Model Evaluation:** After training, the models are evaluated using metrics such as accuracy, precision, recall, sensitivity, and F1-score. This evaluation is essential to ensure that the model can classify coffee bean defects accurately and reliably.

C. System Workflow

The system workflow is illustrated in (Fig. 3)

The following is a detailed explanation of each step involved:

1) **Start:**This marks the beginning of the system process.
2) **Image Input:** The user uploads or selects the coffee bean image to be analyzed.
3) **Convert Image to Base64:** The image is converted into a Base64 string format so that it can be transmitted from the HTML page to the backend Flask through JavaScript.
4) **Extract and Decode Base64:** In the backend Flask, the Base64 image is received and decoded

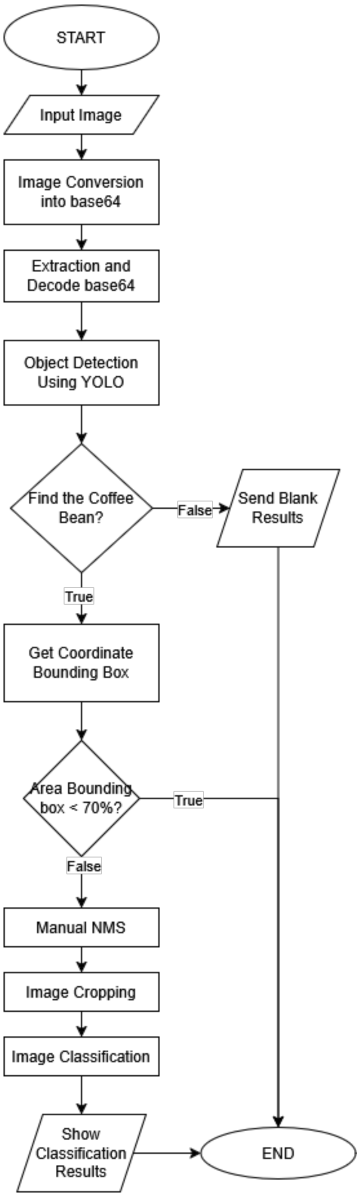


Fig. 3. System Workflow.

back into its original image format.

5) **Object Detection Using YOLO:** The YOLOv11 model is used to detect defective coffee beans in the image. YOLO provides results in the form of bounding box coordinates (x1, y1, x2, y2), confidence score, and class index of the detected objects.
6) **Is a Coffee Bean Detected?:** At this point, the system checks whether any objects have been detected by YOLO:
   - If no object is detected (no coffee beans found):
     - The system returns an empty result, indicating that no coffee beans were detected.
     - The process proceeds directly to the finishing stage.
   - If objects are detected:
     - The system proceeds to the next step by extracting the bounding box coordinates.
7) **Extract Bounding Box Coordinates:** The coordinates (x1, y1, x2, y2) from YOLO's detection results are extracted for further processing and used for cropping the image.
8) **Is the Bounding Box Area $<$ 0.7%?:** The system checks whether the bounding box area is too small (less than 0.7% of the total image area). This check helps eliminate noise from very small detected objects (false positives).
   - Yes: The detected object is considered noise, and the process terminates without further processing.
   - No: The process continues to the next step.
9) **Manual Non-Maximum Suppression (NMS):** Manual Non-Maximum Suppression is applied to eliminate duplicate bounding boxes that overlap. NMS ensures that only the best bounding box is kept if there are multiple detections of the same object.
10) **Crop Image:** Based on the bounding box from NMS, the image is cropped to create a new image containing a single coffee bean. Each cropped image is sent to the CNN for classification.
11) **Classify Image:** The cropped image is classified using the EfficientNetV2S CNN model. The result includes the probability for each class (e.g., normal, cracked, quaker, moldy), and the class with the highest probability is selected as the classification result.
12) **Display Classification Results:** The label and confidence of the classification result are displayed on the user's web page.
13) **Finish:** The process ends, and all results are shown to the user.

III. RESULT

This section presents the experimental results obtained from the implementation of the proposed coffee bean defect detection and classification system. The evaluation is divided into two main parts: object detection using YOLOv11 and classification using several Convolutional Neural Network (CNN) architectures, namely EfficientNetV2, ResNet, DenseNet, and MobileNetV3. Performance metrics such as accuracy, precision, recall, F1-score, and mean Average Precision (mAP) are reported to provide a comprehensive assessment. Furthermore, the integration of the best-performing models is demonstrated to show the effectiveness of the end-to-end system.

A. Object Detection

The detection model was developed using YOLOv11 to identify both normal and defective coffee beans. (Fig. 4) presents the training and validation curves for box loss, classification loss,

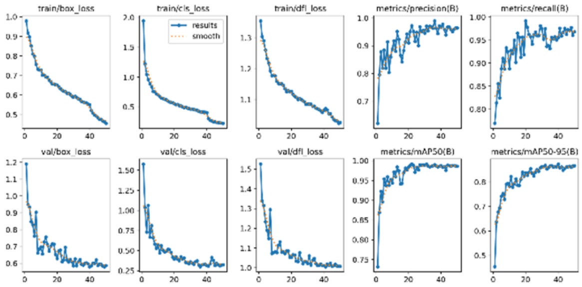


Fig. 4. Training Metrics and Loss Curve.

distribution focal loss (DFL), and evaluation metrics including precision, recall, and mean Average Precision (mAP).

During training, the box loss, classification loss, and DFL loss decreased steadily across the epochs. For instance, the training box loss decreased from around 1.0 in the first epoch to below 0.5 after 50 epochs. Similarly, the validation losses followed the same downward trend, indicating that the model not only learned effectively but also generalized well on unseen data. The closeness between training and validation loss curves also suggests that the model avoided significant overfitting, which is crucial for robust performance in real-world scenarios.

The evaluation metrics further highlight the effectiveness of YOLOv11. The precision increased rapidly during the initial epochs and stabilized above 0.95 by the end of training. This demonstrates that the model was able to minimize false positives, ensuring that most of the predicted bounding boxes corresponded to actual coffee beans. The recall also showed a strong performance, reaching values around 0.96, which indicates that the model was capable of correctly detecting nearly all beans present in the images, minimizing false negatives.

In terms of detection accuracy, YOLOv11 achieved a mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) of approximately 0.98, while the mAP@0.5–0.95 reached around 0.89. The high mAP@0.5 indicates excellent detection accuracy under relatively lenient overlap thresholds, whereas the slightly lower but still strong mAP@0.5–0.95 reflects robust performance across stricter overlap criteria. This balance demonstrates that the model can consistently detect coffee beans under varying levels of localization precision. The model performance is presented in (Table 2.

Overall, the results confirm that YOLOv11 is highly effective in detecting normal and defective coffee beans. The combination of low loss values, high precision, high recall, and strong mAP scores demonstrates that the model provides accurate and reliable bounding box predictions. These outputs serve as the foundation for the subsequent classification stage, where each

Table 1. Model Performance

Model Performance		
Metric	**Validation Result**	**Explanation**
Precision	0.95	The model minimized false positives, ensuring accurate detections.
Recall	0.96	The model minimized false negatives, detecting nearly all beans.
mAP@0.5	0.98	Achieved very high detection accuracy at IoU = 0.5.
mAP@0.5–0.95	0.89	Maintained strong performance under stricter IoU thresholds.

detected bean is cropped and further classified into specific defect categories.

B. Classification

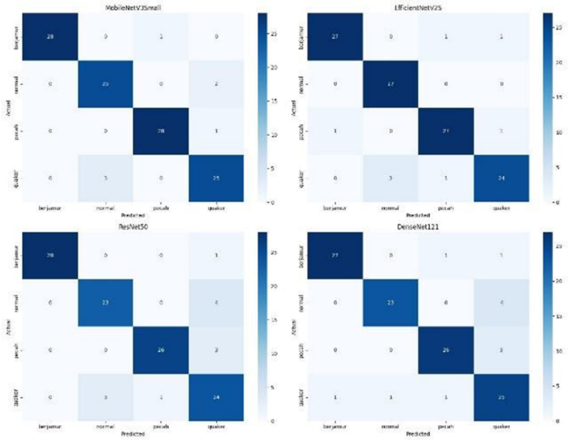


Fig. 5. Comparison of Confusion Matrices Across CNN Architectures.

Figure 5 presents the confusion matrices of the four convolutional neural network (CNN) architectures—MobileNetV3Small, EfficientNetV2S, ResNet50, and DenseNet121—evaluated on four classification categories: moldy beans, broken beans, underdeveloped beans (quaker), and normal beans.

For **MobileNetV3Small**, the model correctly classified 28 out of 29 moldy bean samples, 25 out of 27 normal beans, and 28 out of 29 broken beans. The quaker class was also well recognized, with 25 correct predictions, although three samples were misclassified as normal. These results indicate that MobileNetV3Small demonstrates strong capability in detecting defective beans, with only minor confusion occurring between the visually similar normal and quaker classes.

EfficientNetV2S achieved the most consistent and accurate performance across all categories. The model correctly classified 27 moldy beans, all 27 normal beans, and 27 out of 29 broken beans. In the quaker class, 24 samples were classified correctly, with only three misclassifications. This architecture exhibited the lowest level of confusion in distinguishing between normal and defective beans, highlighting its effective

Table 2. Performance Comparison of CNN Architectures

Metric	Model Performance Comparison			
	MobileNetV3Small	EfficientNetV2S	ResNet50	DenseNet121
Test Accuracy	0.93	0.95	0.91	0.92
Test Loss	0.21	0.18	0.25	0.23
Precision	0.92	0.94	0.90	0.91
Recall	0.91	0.95	0.89	0.90
F1-Score	0.915	0.945	0.895	0.905

balance between high accuracy and good generalization.

For **ResNet50**, the model correctly predicted 28 moldy beans, 23 normal beans, and 26 broken beans. However, four normal beans were misclassified as quaker, and three quaker beans were incorrectly predicted as broken. These findings suggest that although ResNet50 was able to extract strong discriminative features for defective categories, it showed reduced robustness in differentiating normal beans from quaker beans.

Similarly, **DenseNet121** correctly identified 27 moldy beans, 23 normal beans, and 26 broken beans. In the quaker category, 25 samples were correctly classified; however, several misclassifications occurred, including one quaker sample predicted as moldy and another predicted as broken. Comparable to ResNet50, DenseNet121 encountered difficulties in separating normal and quaker beans, indicating overlapping visual characteristics between these two classes.

To complement the qualitative insights obtained from the confusion matrices, Table 2 presents a quantitative comparison of the models in terms of test loss, test accuracy, precision, recall, and F1-score. These metrics provide an overall performance evaluation and enable a clearer comparison among the evaluated CNN architectures.

As shown in Table 2, EfficientNetV2S delivered the most consistent and reliable performance across all metrics, leading to its selection as the classification backbone in the proposed YOLO–CNN system for defective coffee bean recognition.

C. Integration

At this stage, the coffee bean defect detection and classification system is implemented within a web interface using the Flask framework as the backend. The integration process is carried out through several key steps that connect deep learning models with the user interface in real-time.

The first step in the integration process is the receipt of the image uploaded by the user through the web interface. The image is received in base64 format via a POST method and is then decoded for further processing. After decoding, the image is converted into an array format using OpenCV, and the color format is changed from RGB to BGR, in accordance with the input format required by the YOLO model.

Once the image is prepared, the next step is object detection using the previously trained YOLO model. YOLO is responsible for detecting the location of coffee beans in the input image. The YOLO model, loaded from the *best.pt* file, identifies the bounding boxes that represent the areas containing coffee beans. During this detection process, a confidence threshold of 0.3 and an Intersection over Union (IoU) threshold of 0.6 are applied to ensure that only valid detections are processed further. If no objects are detected, the system sends an empty response back to the user interface.

After detecting objects, the next step is to avoid duplicate detections of the same coffee beans. The *is_too_close* function is employed to check if two bounding boxes are too close to each other. This function calculates the Euclidean distance between the centers of the two bounding boxes, and if the distance is smaller than a predefined threshold (e.g., 20 pixels), the two boxes are considered to represent the same object. As a result, only one bounding box is processed for further analysis.

Once duplicate checks are completed, the image regions corresponding to the bounding boxes are cropped and prepared for the next step of classification. Each cropped image resulting from the YOLO detection is resized to 224×224 pixels, in line with the input requirements of the EfficientNetV2 CNN model. The cropped image is processed using *preprocess_input* from the CNN model and then passed through the model for class prediction and confidence assessment. The CNN model classifies each coffee bean into one of four categories: moldy, cracked, quaker, or normal. The classification result includes the predicted class label and the confidence level of the classification.

After classification, the results, including the class label, confidence level, and bounding box information, are returned to the user interface in JSON format. The system then displays the image with annotated bounding boxes and classification labels on the web page, allowing users to visually review the detection and classification results.

Finally, the system re-encodes the annotated image back into base64 format and sends it back to the user interface, which then displays the image along with the classification results on the screen. In this way, the predict endpoint in the Flask backend serves as the bridge between the user interface and the two deep learning models (YOLO for object detection and EfficientNetV2 for classification), enabling the entire process of detection and classification to run automatically and seamlessly integrated. The output of the system can be observed in (Fig. 6)

D. System Testing and User Evaluation

In the model evaluation stage for detection and classification, testing is conducted to measure the accuracy of the developed system. The system testing in this study uses a confusion matrix approach and black-box testing methods. Based on the image, the evaluation

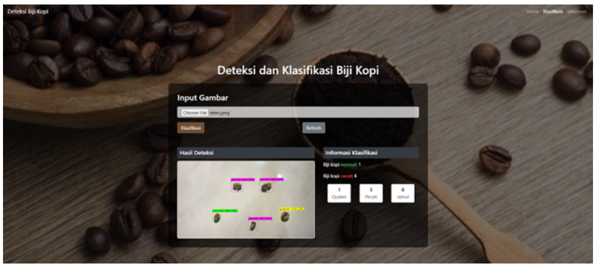
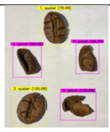


Fig. 6. Output System.

Table 3. Comparison Between Actual Classes and CNN Predictions

No	Output Image	Actual Class	CNN Predictions	Information
1		Quaker: 3 Normal: 1 Broken: 1	Quaker: 3 Normal: 1 Broken: 1	Appropriate Class
2		Normal: 10	Normal: 10	Appropriate Class
3		Normal: 10	Normal: 10	Appropriate Class
4		Broken: 3 Quaker: 2	Broken: 3 Quaker: 2	Appropriate Class
Jumlah Biji Kopi		**30**	Normal: 21 Broken: 4 Quaker: 5	

results show that the system accurately detects and classifies the coffee beans into their respective categories. The ”Actual Class” and ”CNN Predictions” columns match for all the tested samples, indicating that the CNN model has appropriately classified the coffee beans as Normal, Pecah, or Quaker. In total, the system classified 21 Normal beans, 4 Pecah beans, and 5 Quaker beans out of 30 samples, demonstrating that the system is effective in identifying and categorizing the defects of the coffee beans.

Based on (Table. 3), the evaluation results show that the system accurately detects and classifies the coffee beans into their respective categories. The ”Actual Class” and ”CNN Predictions” columns match for all the tested samples, indicating that the CNN model has appropriately classified the coffee beans as Normal, Broken, or Quaker. In total, the system classified 21 Normal beans, 4 Broken beans, and 5 Quaker beans out of 30 samples, demonstrating that the system is effective in identifying and categorizing the defects of the coffee beans.

E. Confusion Matrix Testing

Testing using a confusion matrix is conducted to evaluate the performance of the classification model based on the prepared test data. The confusion matrix provides a detailed overview of the number of correct and incorrect predictions for each class, as well as showing how often the model makes errors in distinguishing between classes. In this study, the confusion matrix is used to calculate important evaluation metrics such as accuracy, precision, recall, and F1-score for each coffee bean class: moldy, cracked, quaker, and normal. By using the confusion matrix, researchers can identify the model’s weaknesses in recognizing specific classes, which will serve as the basis for future improvements.

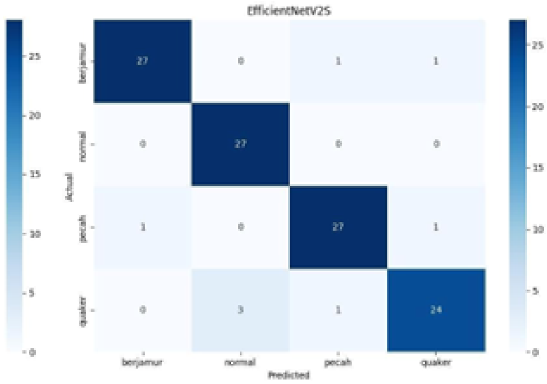


Fig. 7. Confusion Matrix of the EfficientNetV2S Model.

Based on the confusion matrix shown in (Fig.7), the EfficientNetV2 model demonstrates excellent classification performance across the four coffee bean defect classes: moldy, normal, cracked, and quaker. The model successfully classifies most samples correctly, as reflected by the high accuracy and F1-score values. The normal class is the most accurately classified, with a recall of 100% and an F1-score of 94.73%, indicating that all normal coffee bean images were correctly identified by the model without any classification errors. The moldy and cracked classes also show very good performance, with F1-scores above 93%, although there were a few misclassifications within each class. Meanwhile, the quaker class exhibits the lowest performance among the classes, with a recall of 85.71% and an F1-score of 88.88%.

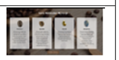
F. Black Box Testing

Testing of the system’s functionality in this study was conducted using a black-box testing approach, applied to the coffee bean defect detection and classification system. The purpose of this testing was to evaluate whether the system can perform its functions according to user requirements and the specified specifications. The testing was carried out without examining the program code structure, instead focusing solely on evaluating the system’s responses. The results of the black-box testing can be observed in (Table 4)

IV. DISCUSSION

The detection results using the YOLOv11 model indicate that the system is highly effective in identifying coffee bean defects in digital images. The model successfully recognized various types of defects, including moldy, broken, quaker, and normal beans with high precision. Based on the training results, the

Table 4. Black Box Testing

No	Test Description	Action	Expected System Response	Results
1	Main Menu Test	Open the system	Redirect to the dashboard page	
2	Classification Menu Test	Press the Classification button	The system displays detection and classification results for the uploaded image	
3	File Uploaded Menu Test	Press the Choose File button	The system displays the file uploaded form	
4	Classification Button Test	Press Classification button	The system displays the output image of the classification result.	
5	Refresh Button Test	Press the Refresh Button	The system displays only the file upload form without any image.	
6	Information Menu Test	Press the Information button	The system displays information about the classes in the coffee bean defect detection.	
7	Home Menu Test	Press the Home button	Redirect to the main menu page.	

mean Average Precision (mAP@0.5) reached 98.83%, demonstrating the model's excellent object detection capability. This finding aligns with previous studies reporting that YOLO, particularly YOLOv4 and YOLOv5, is highly efficient for object detection across various domains, including agriculture and product processing [13]. The success of YOLOv11 in this study is likely due to its end-to-end single-stage processing, which enables rapid object detection without compromising accuracy.

These results are consistent with earlier research applying YOLO for object detection in coffee bean processing [14]. However, this study provides additional contributions by using a larger dataset of 5,636 images, covering a wider variation of coffee bean conditions. Previous research highlighted that YOLOv3 and YOLOv4 struggled with detecting small or overlapping objects [15]. In contrast, YOLOv11 demonstrated significant improvement in detecting small objects, such as quaker beans, which often overlap with normal beans. This indicates that the latest YOLOv11 architecture addresses some of the limitations observed in earlier versions.

From a practical standpoint, the use of YOLOv11 for defective coffee bean detection offers significant benefits to small and medium enterprises (SMEs) in the coffee industry. With high precision, this system enables automatic sorting, reducing dependence on manual labor and improving efficiency. For instance, SMEs that previously relied on human workers for defect sorting can now save time and operational costs. This automation also enhances the consistency of coffee bean quality, potentially increasing market competitiveness. This aligns with previous findings that deep learning-based automation can improve quality and efficiency in agriculture and product processing [16].

Despite the promising results, several limitations should be acknowledged. One limitation is the quality of the images in the dataset. Images taken under poor lighting or with low-contrast backgrounds may affect the model's accuracy in detecting defects. This was evident in some samples where shadows or visual noise caused misdetections, particularly for quaker and broken beans. Additionally, while EfficientNetV2 performed well for classification, it still experienced some difficulties in distinguishing between normal and quaker beans, which are visually similar. These limitations could be mitigated by collecting a more diverse and representative dataset reflecting real-world conditions [17].

Future research could focus on developing a real-time detection system using portable devices such as smartphone cameras or webcams, making the system more practical for field use. Improving image acquisition quality with better cameras and lighting conditions could further enhance detection accuracy. Expanding the dataset with more samples under varying conditions of lighting and background would help the model generalize better to real-world scenarios. Moreover, exploring transfer learning approaches could boost classification performance by leveraging pre-trained models on larger or more diverse datasets.

Overall, the findings of this study demonstrate that the combination of YOLOv11 for detection and EfficientNetV2 for classification is highly effective for detecting and classifying defective coffee beans. This integrated system offers an efficient solution for automating coffee bean sorting, with significant implications for improving product quality and operational efficiency in the coffee industry. Nevertheless, challenges related to lighting conditions and image variability remain, which can be addressed through larger and more diverse datasets and improved image acquisition methods.

V. CONCLUSSION

Based on the results and discussion presented in this study, it can be concluded that:

1) The You Only Look Once (YOLOv11) method proves to be effective for the automatic detection of defective coffee beans in digital images. The model delivers accurate and fast detection of various types of coffee bean defects, including moldy, cracked, quaker, and normal beans.
2) The application of the YOLOv11 model on a dataset of 5,636 processed images, trained with specific configurations, demonstrates excellent detection performance, achieving a mean Average Precision (mAP@0.5) of 98.83%. This

result indicates a high level of precision in detecting defective coffee beans.

3) The use of several Convolutional Neural Network (CNN) architectures for defect classification shows that the EfficientNetV2 model performs the best with a test accuracy of 93.81%, followed by MobileNetV3Small, ResNet50, and DenseNet121. Evaluation using the confusion matrix resulted in the highest F1-score of 94.73% for the normal class.
4) Overall, these results confirm that the combination of YOLOv11 for detection and CNN (EfficientNetV2) for classification provides an effective and efficient solution for building an automatic coffee bean defect sorting system. In addition, functional testing using a black-box approach on the web interface demonstrates that all system features operate as expected.

REFERENCES

[1] Badan Pusat Statistik, *Produksi Perkebunan Menurut Kabupaten/Kota dan Jenis Tanaman (ton), 2022 dan 2023*. Jakarta, Indonesia: BPS, 2023.

[2] A. E. N. Ramadhan, W. Setiawan, dan D. C. Khrisne, "Rancang bangun deteksi objek dengan metode filter warna HSV pada sistem klasifikasi kualitas biji kopi berbasis NVIDIA Jetson Nano," *Jurnal Teknik Industri Terintegrasi*, vol. 6, no. 4, pp. 1500–1509, 2023, doi: 10.31004/jutin.v6i4.21406.

[3] Y. Hafifah, K. Muchtar, A. Ahmadiar, dan S. Esabella, "Perbandingan kinerja deep learning dalam pendeteksian kerusakan biji kopi," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 6, p. 1928, 2022, doi: 10.30865/jurikom.v9i6.5151.

[4] S. J. Chang and C. Y. Huang, "Deep learning model for the inspection of coffee bean defects," *Applied Sciences*, vol. 11, no. 17, 2021, doi: 10.3390/app11178226.

[5] S. J. Chang and K. H. Liu, "Multiscale defect extraction neural network for green coffee bean defects detection," *IEEE Access*, vol. 12, pp. 15856–15866, 2024, doi: 10.1109/ACCESS.2024.3356596.

[6] H. D. Thai, H. J. Ko, and J. H. Huh, "Coffee bean defects automatic classification realtime application adopting deep learning," *IEEE Access*, early access, 2024, doi: 10.1109/ACCESS.2024.3452552.

[7] Z. Huang, L. Li, G. C. Krizek, and L. Sun, "Research on traffic sign detection based on improved YOLOv8," *Journal of Computer and Communications*, vol. 11, no. 7, pp. 226–232, 2023, doi: 10.4236/jcc.2023.117014.

[8] Y. Yanto, F. Aziz, dan I. Irmawati, "YOLOv8 peningkatan algoritma untuk deteksi pemakaian masker wajah," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 3, pp. 1437–1444, 2023, doi: 10.36040/jati.v7i3.7047.

[9] D. S. Wita dan D. Y. Liliana, "Klasifikasi identitas dengan citra telapak tangan menggunakan convolutional neural network," *Jurnal Rekayasa Teknologi Informasi*, vol. 6, no. 1, p. 1, 2022, doi: 10.30872/jurti.v6i1.7100.

[10] M. A. Mutasodirin and F. M. Falakh, "Efficient weather classification using DenseNet and EfficientNet," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 9, no. 2, pp. 173–179, 2024, doi: 10.30591/jpit.v9i2.7539.

[11] G. A. Rakhmat and F. Rizkiawarman, "Implementasi arsitektur MobileNetV3 (studi kasus klasifikasi jamur beracun)," 2023.

[12] D. P. Sidik, F. Utaminingrum, dan L. Muflikhah, "Penggunaan variasi model pada arsitektur EfficientNetV2 untuk prediksi sel kanker serviks," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 5, pp. 2116–2121, 2023.

[13] B. Zhang, J. Li, Y. Bai, Q. Jiang, B. Yan, and Z. Wang, "An improved microaneurysm detection model based on SwinIR and YOLOv8," *Bioengineering*, vol. 10, no. 12, pp. 1–16, 2023, doi: 10.3390/bioengineering10121405.

[14] A. I. Pradana and J. Maulindar, "Intelligent traffic sign detection using YOLOv9," pp. 352–360, 2024.

[15] A. Sharma, V. Kumar, and L. Longchamps, "Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species," *Smart Agricultural Technology*, vol. 9, p. 100648, 2024, doi: 10.1016/j.atech.2024.100648.

[16] K. Sivakoti, "Vehicle detection and classification for toll collection using YOLOv11 and ensemble OCR," pp. 1–13, 2024.

[17] O. A. Abioye, A. E. Evwiekpaefe, and A. J. Olalekan, "Performance evaluation of EfficientNetV2 models on the classification of histopathological benign breast cancer images," *Scientific Journal of the University of Zakho*, vol. 12, no. 2, pp. 208–214, 2024, doi: 10.25271/sjuoz.2024.12.2.1261.