Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.1, February 2025, pp. 153~160
e-ISSN: 2548-6861

Development of YOLO-Based Mobile Application for Detection of Defect Types in Robusta Coffee Beans

Eko Dwi Nugroho [1]*, Miranti Verdiana [2]*, Muhammad Habib Algifari [3]*, Aidil Afriansyah [4]*, Hafiz Budi Firmansyah [5]*, Alya Khairunnisa Rizkita [6]*, Richard Arya Winarta [7]*, David Gunawan[8]*
* Teknik Informatika, Institut Teknologi Sumatera
eko.nugroho@if.itera.ac.id [1], miranti.verdiana@if.itera.ac.id [2], muhammad.algifari@if.itera.ac.id [3], aidil.afriansyah@if.itera.ac.id [4], hafiz.budi@if.itera.ac.id [5], alya.rizkita@staff.itera.ac.id [6], mailto:richard.121140035@student.itera.ac.id [7], david.121140062@student.itera.ac.id [8]

Article Info

Article history:

Received 2024-11-13
Revised 2024-11-29
Accepted 2025-01-16

Keyword:

Robusta,
YOLO,
Coffee Beans.

ABSTRACT

Improving the quality of Robusta coffee beans is a crucial challenge in the coffee industry to ensure that consumers receive high-quality products. However, the identification of defects in coffee beans is still largely performed manually, making the process error-prone and time-consuming. This study aims to develop a YOLO-based mobile application to detect defects in Robusta coffee beans quickly and accurately. The method employed in this study is YOLO, a deep learning-based object detection algorithm known for its real-time object detection capabilities. The application was tested using a dataset of Robusta coffee beans containing various defects, such as broken, black, and wrinkled beans. The test results indicate that the application achieves high detection accuracy, with the black bean class achieving 95.3% accuracy, while the moldy or bleached bean class records the lowest accuracy at 62.2%. This application is expected to assist farmers and coffee industry stakeholders in improving the quality of Robusta coffee beans and enhancing the efficiency of the sorting process.

I. Introduction

Coffee is one of the plant commodities that has a very high economic value [1], [2], [3]. Indonesia is one of the largest coffee exporters in the world and coffee plays a very important role in the country's foreign exchange sources. The most popular and mainstay types of coffee for export are robusta and arabica coffee [4]. Coffee in Indonesia is cultivated in many scattered areas, resulting in coffee being very diverse in taste and aroma.

Coffee that will be exported to other countries will go through a quality testing stage by coffee farmers in order to maintain and ensure the quality of the coffee that will be sold. One of the coffee quality testing mechanisms commonly used in Indonesia is SNI 01-2907-2008. Each type of arabica and robusta coffee has a different quality level, arabica consists of 6 quality levels and robusta as many as 7 quality levels. Coffee farmers who have a business scale that is not too large carry out quality testing using dry process processing techniques [5], [6]. Dry process processing consists of the stages of selecting fruit, drying fruit by drying it in the sun, peeling coffee, selecting beans, and packaging. Each stage of processing is done manually by farmers without the help of machines like in wet processing.

The manual coffee processing process by farmers, especially with the drying method, takes a relatively long time and can affect the accuracy in selecting quality coffee beans. To facilitate coffee processing, efficient and affordable technology for small-scale farmers can be utilized, so they do not need to buy expensive coffee processing machines. Mobile applications can be a solution to this technology, with its ability to test coffee quality according to applicable standards and easy access for farmers. By utilizing mobile applications, coffee farmers with small business scales can compete in quality and quantity with coffee farmers who usually use machines for coffee processing. Mobile applications for testing coffee bean quality utilize machine learning mechanisms to assess each coffee bean tested.

Therefore, the mobile application used will be integrated with a machine learning model that aims to classify the type of defects in coffee beans.

Previous research that developed a method for classifying types of coffee bean defects was conducted by Ikhsan [7] who applied K-Nearest Neighbor (K-NN) to classify Arabica single-breed coffee, damaged Arabica single-breed coffee, Arabica coffee, and broken Arabica coffee. The results of the classification using K-NN which combined all features of the ratio, perimeter, slenderness, roundness ratio, area, and circumference of the coffee beans obtained a classification accuracy of 63.5%. Further research that classified the physical quality of coffee was conducted by Mardisa [8] who also used the K-NN method. In this study, the classified data scheme was divided into 2, namely the image of inverted and upside-down coffee beans. From each scheme there are 4 classification classes, namely normal, broken, brown, and partially black beans. The results of the study showed that the value of K = 5 was able to obtain an accuracy of 78.625% while the value of K = 3 was able to obtain an accuracy value of 58%.

In addition to using K-NN as a classification technique, research [9] conducted an experiment to classify coffee bean quality using the multilayer perceptron (MLP) technique. The features assessed in this technique are the color features (RGB, HSV, CMYK, LAB, YUV, HCL, and LCH) of coffee beans. MLP was paired with other classification techniques such as Naive Bayes, Decision Tree, and SVM. The best classification results were obtained by MLP on each color feature tested (RGB = 38%, HSV = 57%, CMYK = 63%, LAB = 58%, YUV = 58%, HSI = 42%, HCL = 65%, and LCH = 78%). Furthermore, there is research [10] which has a similar class distribution to research [8] with whole, broken, brown, and black coffee beans which were developed using Linear Discriminant Analysis (LDA) and SVM. The features extracted in this study were color, texture, and shape features. The results of the study prove that the SVM method is able to provide accurate classification of coffee beans with a training accuracy value of 93.56% and a testing accuracy of 80.75%. Research [11] measured the quality of coffee beans using SVM, Deep Neural Network, and Random Forest. The features extracted in this experiment are in the form of coffee bean shape and color features. There are two color features used, namely RGB and HSV. The quality of coffee beans in the dataset used consists of 5 classes, namely whole beans, brown beans, damaged/broken beans, black beans, and husk. The classification results produced in the experiment were able to achieve an accuracy of 88%.

There is also research related to the development of classification techniques carried out by developing classification techniques that are friendly and lightweight for mobile applications. The development of classification techniques that adapt to mobile applications can provide more applicable solutions and can be used directly by farmers as users. In the study [12], a mobile application was developed that utilizes a machine learning model for coffee bean quality classification. The models developed used ResNet-152 and VGG16. The results of each model in testing the quality of coffee beans reached an accuracy of 73.3% for ResNet. However, there are shortcomings in the application of the ResNet-152 and VGG16 models where the parameters used in the model are very large so that they burden the application in processing coffee bean images.

Research [13] developed a classification model with MobileNetV2 to assess the quality of roasted coffee beans. Compared to research [12] using ResNet-152 and VGG16, the MobileNetV2 model has relatively fewer complexity and parameters. The accuracy results obtained for the classification of coffee bean quality in three classes reached 96.44%. There is also research [14] which developed a defect classification application in coffee beans using YOLOv5. The types of defects in the classified coffee beans consisted of broken, black, and normal beans. The experimental results showed an overall accuracy of 95.11%. The accuracy per category was 91.45% for normal coffee beans, 100% for black coffee beans, and 94.23% for broken coffee beans.

Based on previous research, the development of a coffee bean quality classification method has good performance from the most traditional methods such as K-NN, SVM, LDA, DNN, ResNet-152, VGG16, and YOLOv5. Of all the methods compared, YOLO is the method that has the best performance and is the lightest to be applied to a mobile application. Therefore, this study aims to develop a mobile application system integrated with YOLO to identify defects in coffee beans. The objectives of this study can be specified as follows:

(1) Developing a mobile application for coffee bean defect identification.
(2) Integrating YOLO into the application system to assist in identifying coffee bean defects.
(3) Evaluating the capabilities of the mobile application that has been integrated with YOLO in detecting coffee bean defects.

This study will also utilize the dataset that has been collected by robusta coffee farmers in Lampung Province and has been divided into 4 classes. The classes of coffee beans to be tested consist of normal beans, moldy beans, black beans, and hollow beans. The hope for the future of this research is to be able to contribute directly to coffee farmers in Lampung Province in testing the quality of coffee beans with an application that is easy to use and accessible. So that by using the coffee bean defect detection application, it can help small coffee farmers to compete with large-scale coffee farmers who have used machines to select quality coffee beans.

II. METHOD

A. *Research Framework*

This research begins with the identification and definition of the problem, which focuses on understanding the limitations of traditional coffee defect analysis methods. To improve efficiency and accuracy in coffee quality evaluation, this step sets the research context by investigating the feasibility of a machine learning system that can detect defects automatically.

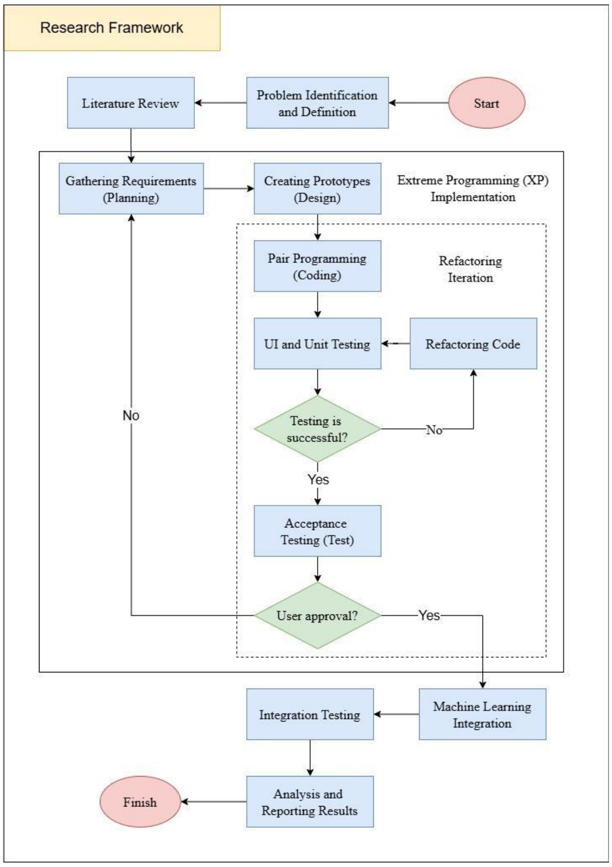

Figure 1. Research Framework

Defining the problem helps to establish clear objectives and sets the stage for focused application development. A literature review is conducted once the problem has been identified. This step involves examining existing research on coffee defect detection, machine vision models such as YOLO, and Android applications that utilize machine learning. This phase helps to build a strong theoretical foundation for the research process and find gaps that can be filled by the newly developed application by gathering information from previous research.

The next stage is part of Extreme Programming (XP) where XP covers all stages of the system development process, including design, programming, testing, and planning. By integrating the XP approach into each development iteration, each development stage can adapt well, be sensitive to user input, and focus on creating reliable and user-centered programs. In the XP approach, there are several stages of development that begin with the gathering requirements stage. Gathering requirements include determining functional and non-functional requirements. Planning at this stage focuses on determining clear goals for the main functionality and usability of the mobile application, while machine learning is integrated as an additional feature for coffee defect detection. Next, there is the creating prototypes stage created during the design process, with a primary focus on the user interface, navigation, and important features. This design prioritizes creating mobile applications that meet the core needs of users. The next stage after getting the appropriate interface design is to start developing the mobile application. At this stage, the programming language used is Kotlin. Along with the development of the mobile application, a machine learning model is also built using YOLO to support the desired detection function. After the application development is complete, unit and interface testing is carried out as the next step to ensure that each component of the application functions according to the specifications that have been set. Acceptance testing is carried out after unit and interface testing to ensure that the features contained in the mobile application meet all the criteria needed based on the needs that have been previously compiled.

The machine learning model that has been developed is integrated with the mobile application that has been completed. Along with this process, the application and model are integrated to ensure that the application is able to run properly to operate the model. The last stage is the analysis of the application's performance in detecting defects in coffee beans. An overview of the entire stage is illustrated in Figure 1.

B. *Mobile Application Architecture*

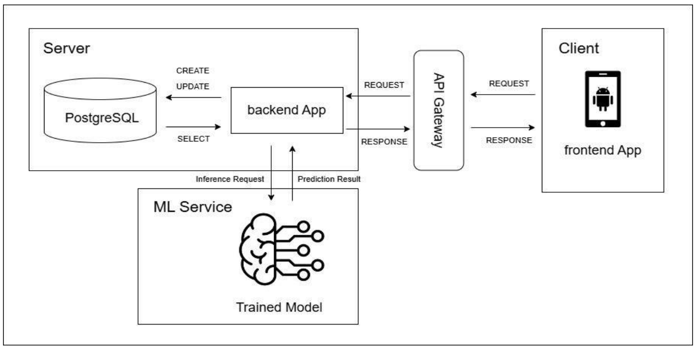

Figure 2. Mobile Application Architecture

The relationship between the components that build a mobile application to detect defects in coffee beans is depicted in the form of a mobile application architecture. The illustration of the mobile application architecture is illustrated in Figure 2. The input variables processed by the application are obtained from the client component that uses the mobile application by taking a picture of the coffee beans. The mobile application will make a request to the API then the server to continue to the image processing stage with a machine learning model and image storage. The results of the detection with the model will be returned to the client side by providing a response in the form of coffee bean defect detection information along with detection accuracy.

C. Sequence Diagram

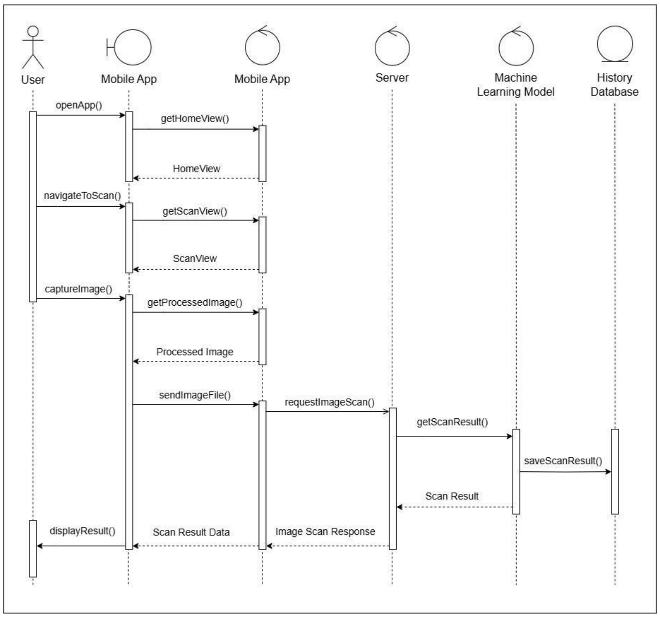

Figure 3. Sequence Diagram

The activities that take place in the coffee plant defect detection application are illustrated in the form of a sequence diagram in Figure 3. The user starts the interaction by opening the mobile application. Once the application is opened, the main page is displayed to the user. From the main page, the user then navigates to the scan page, which is also displayed to the user. On the scan page, the user can take a picture using the application. After the picture is taken, the application processes the image locally before sending it to the server for further processing.

After the image is sent, the server receives a request for image processing and passes it to the machine learning model for analysis. The machine learning model then sends the scan results back to the server. The server stores the scan results into a history database and sends the image processing results back to the mobile application. The mobile application receives the scan results from the server and displays them to the user. This process ends with the display of the scan results in the application that is visible to the user.

D. Application Database Format

The results of defect detection on coffee beans are stored in a PostgreSQL database with data descriptions in the form of image id, image name, image timestamp, and detection results. An overview of the database table that stores the detection results is illustrated in Table 1. The image id functions as the primary key of the stored image, the image name functions as a description of the image, the image timestamp stores the description of the time of capture, and the detection results store the classification results of the stored image.

TABLE I
SCAN HISTORY

Column	Key	Data type	Nullable	Default
scan_id	PK	Integer	FALSE	Sequence
scan_image_uri	-	Text	FALSE	-
scan_timestamp	-	Timestamp	FALSE	-
scan_result	-	JSON	TRUE	-

E. YOLO

You Only Look Once (YOLO) is an object detection method that simplifies detection as a direct regression problem from an image to an image with predicted results in the form of bounding boxes and class probabilities simultaneously. Unlike conventional approaches such as Deformable Parts Models (DPM) and Region-based Convolutional Neural Networks (R-CNN) that require complex multi-component pipelines, YOLO uses a single neural network architecture to perform the entire detection process in one step. Which means YOLO has lower complexity and is lightweight. In YOLO, a convolutional neural network divides the input image into a grid that serves as an object detection unit. Each cell in the grid is responsible for detecting objects whose centers are within the area of that cell, while predicting the bounding box, confidence score, and class probability of the detected object. This integrated process allows YOLO to be optimized end-to-end for overall detection performance, reducing the excessive computational requirements of multi-layered detection setups and enabling real-time image processing [15].

F. Dataset for Training

The development of the YOLO model for defect detection in coffee beans uses a dataset with a total of 1860 coffee bean images. The dataset is divided into three types of data, namely training data, validation data, and test data. The ratio of the division of these three data is 90% for training data, 5% for validation data, and 5% for test data. Sample images from the dataset used in the development of the YOLO model are illustrated in Figure 4. The images used for model training were taken under the same conditions and environment, to ensure consistency in the colour of the coffee beans.

The coffee bean classes in the dataset consist of 4 classes, namely normal beans, moldy beans (bleach), black beans, and hollow beans (berry bore). Normal beans are coffee beans that do not have any physical defects. Moldy beans are coffee beans that have a brighter color due to exposure to mold. Hollow beans or berry bore are coffee beans that are exposed to pests so that there are holes in the beans. Black beans are beans that have a very dark color even though they have not gone through the bean roasting process. Visualization of each class can be seen in Table II. The labelling of the coffee bean class in the image was done manually with the help of a coffee farmer who was tasked with validating the coffee beans according to the supposed class.

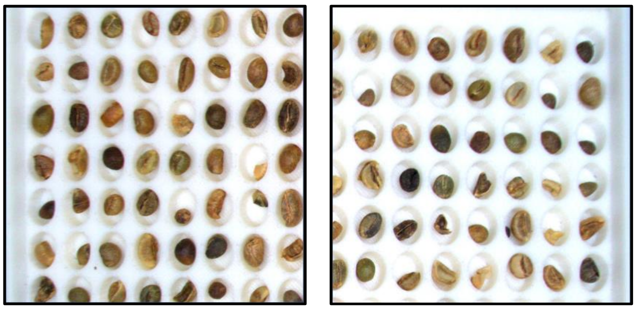

Figure 4. Sample Database

G. Training Preparation

YOLO is trained with a hyperparameter environment using the Adam optimiser with a learning rate of 0.0001. To reduce and prevent overfitting, L2 regularization or weight decay of 0.0005 is also added. Epochs used in this training were 50 with a batch size of 4. The size of the trained image is adjusted to the needs of YOLO which is 640 x 640. In the initial training, an early stopping function is given to find out at what epoch the training starts to stabilise. Therefore, the selection of 50 epochs is the result of the first training with early stopping which illustrates that at epoch 50 the model already has the best performance.

TABLE II
IMAGE CLASS

Class	Bleach	Berry Bore	Black	Normal
Image				

III. Result and Discussions

A. Model Results

Based on the results of training and testing the model built using mAP50, the model has different performance in each class. The class with the highest mAP50 value belongs to the black seed class of 76 for testing and 74 for training. This high mAP50 result is due to the black seed class having a contrasting colour difference. In contrast to the bleached bean class which also has quite a contrasting difference based on a lighter colour compared to normal beans, the bleached bean class is only able to have mAP50 values for testing and training of 39 and 41 respectively. Berry Bore and Normal classes have lower mAP50 values of 46 and 42 respectively, and the Bleach class reaches 39. The lowest mAP50 value is found in the normal class, this may be due to its similarity to healthy coffee beans or other classes that have similar properties. The results of testing and training are presented in Table III.

TABLE III
CLASS PERFORMANCE

Class	**Bleach**	**Berry Bore**	**Black**	**Normal**
Testing (mAP50)	39	46	76	42
Training (mAP50)	41	29	74	36

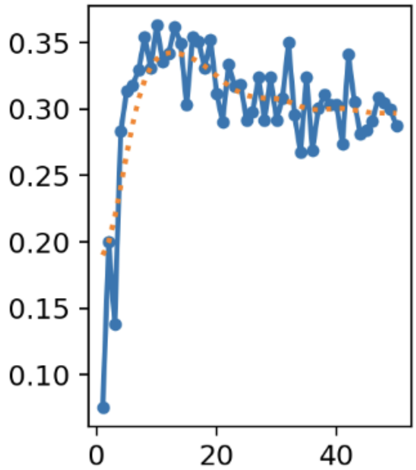

Figure 5. Training Graph (mAP50)

The training graph in Figure 4 shows that the overall mAP50 of the model increases significantly in the first few epochs and then stabilises in the range of 0.30-0.35. This stability indicates that the model has reached convergence, but the mAP50 value can be further improved in underperforming classes. The main challenge is the visual difficulty of some classes and the potential uneven distribution of data. To improve the performance of classes with low mAP50 values, measures such as data augmentation, adjusting anchor boxes, and increasing the amount of data for minority classes can be taken.

The model was built using a dataset divided into 90% for training, 5% for validation, and 5% for testing. To ensure that the model does not occur overfitting, initial training is carried out using the early stopping function so that it can determine at what epoch the model has learned optimally. The training also added L2 regularization of 0.0005 to avoid overfitting. Therefore, the validation technique used in the development of this model is a separate test data validation technique that utilises validation data as a parameter of the model's success in learning.

B. *Application Results*

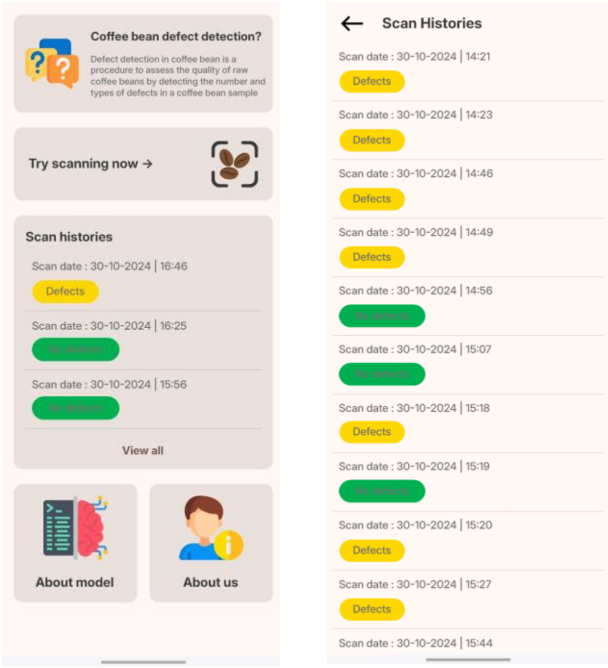

Figure 6. Detection Result Display

The coffee bean defect detection application was developed according to the research stages that have been arranged using the XP development method. The main display of the application can be seen in Figure 6 which contains information related to the coffee bean defect test application, options for performing defect detection, a summary of the coffee bean detection history, and information related to the model and developer. The test history page contains information on coffee bean detection along with a description of the detected defects in the coffee bean image sample. There is also an option to view the detection results in more detail. All detection results are stored in the database in the form of images. So that users can look back at the detection results that have been done before.

In testing the application's ability to detect defects in coffee beans, the shooting conditions are adjusted to the existing dataset conditions. The capture conditions are adjusted in such a way as to capture the dataset, this aims to make the detection results more accurate. The only difference in the test conditions of the application is that the background for picking coffee beans uses a red background. The experiment with a red background aims to reduce the colour of black beans and bleach which is too intense when using a white or black background. Using a background colour that is too contrasting will affect the detection results. The test results have quite different accuracy from the tests conducted on the model itself during the development stage. Detection success during application testing is higher than model testing. The black seed class has a difference in model testing results and application testing of 19.3% higher application testing. The bleach seed class has a difference of 23.2% higher in application testing. The berry bore seed class has a difference of 31%. There is a significant difference between the results of model testing and application testing. This is due to the use of a background for application testing that uses red to reduce colour contrast.

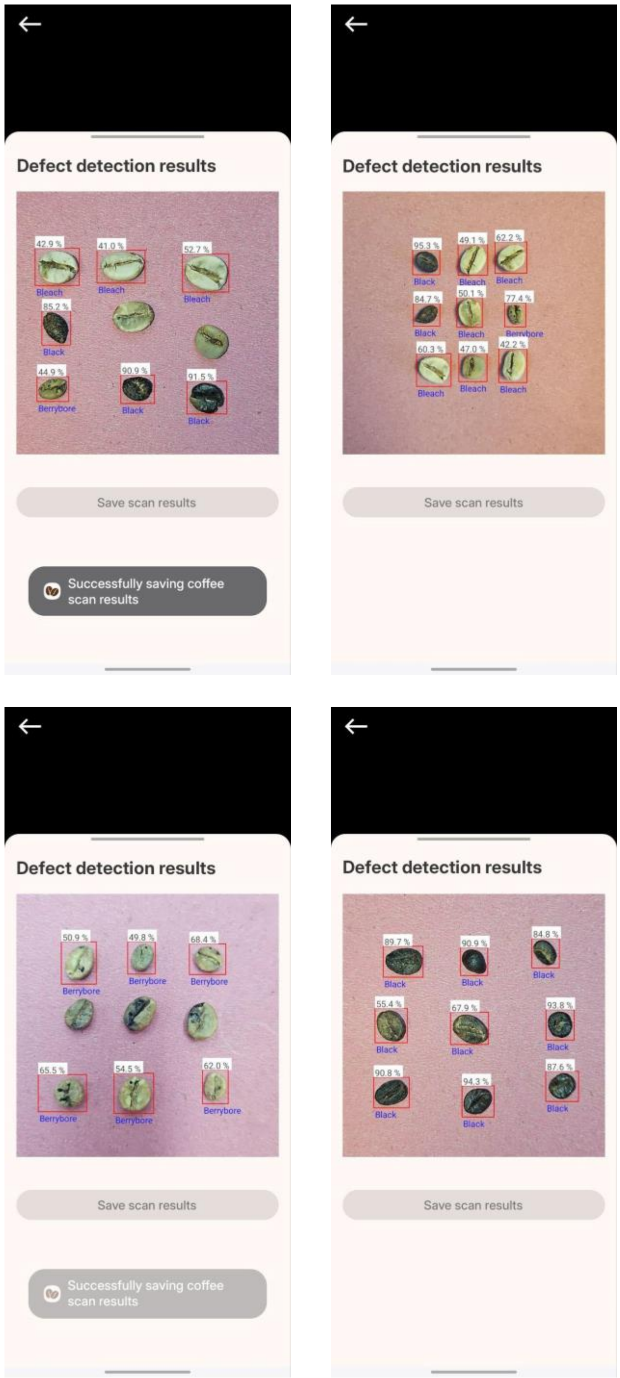

Figure 7. Detection Result Display

Defect detection in coffee beans is done through the “Try scanning now” menu option which produces a display of detection results as in Figure 7. Of all the classes, the black bean class is the class with the highest detection success with an accuracy of 95.3%. Next is the berry bore bean class which has the highest detection accuracy of 77.4%. The moldy or bleached bean class is the most difficult class to distinguish with only achieving an accuracy value of 62.2%. Moldy beans are the most difficult class to distinguish from normal beans. This is because moldy beans still have a similar color to normal beans with a difference in their lighter or paler color. Overall, from the detection results, the coffee bean defect detection application is able to distinguish the coffee bean classes in the sample data well.

C. Functional Testing

TABLE IV
FUNCTIONAL TESTING

Functional	Input	Expected Results	Results
Open App	Selecting the application	The application opens correctly and the main page appears	Success
Navigate to Scan	Selecting detection menu	The application page can switch to the scan page without any problems and the scan page appears correctly	Success
Capture Image	*Capturing picture by using smartphone camera*	The image was successfully taken and appears in the application	Success
Get Processed Image	Picture captured by the camera	The application successfully processed the captured image	Success
Send Image File	Picture captured by the camera	The image was successfully sent to the server and the server received the image process request	Success
Request Image Scan	Using the captured images to process with the YOLO model in the application	The captured image is able to be detected by the model	Success
Restore Scanning Results	Pictures that have been detected by the model	The image detected by the model was successfully saved to the database	Success
Display Scanning Results	Pictures stored in the server	The image of the detection result by the model stored on the server is successfully displayed on the application page	Success

Functional testing of the application is presented in Table IV which consists of 8 main functions. The main functions tested are derived from the objectives of the YOLO-based mobile application development research for defect detection in coffee beans. Overall, the application developed to detect defects in coffee beans has met all the criteria of the 8 main functions of the application. Therefore, this study has succeeded in building a mobile application integrated with a machine learning model to detect defects in coffee beans.

IV. CONCLUSION

This study developed a YOLO-based mobile application for detecting defects in coffee beans. The detection results indicate that among the four defect classes evaluated, the black bean class achieved the highest detection accuracy at 95.3%, outperforming the other classes. In contrast, the moldy or bleached bean class had the lowest accuracy at 62.2%, likely due to its visual similarity to normal beans. The primary distinguishing feature of moldy beans is their brighter and paler color compared to normal beans. The application, designed based on user needs, performed well in its core functions. These include capturing and processing images, sending images to the server, running defect detection using the model, saving detection results to the server, and displaying the results within the application.

The primary contribution of this study is the development of an application that is seamlessly integrated with a machine learning model, facilitating accurate defect detection in coffee beans. This application shows great potential for widespread adoption by small-scale coffee farmers, who typically rely on manual quality assessment methods. By automating defect detection, the application can assist farmers in improving both the quantity and quality of their output, helping them compete with larger farms that utilize advanced technology.

Despite positive results, this study has certain limitations, particularly in its ability to differentiate defects based on color. Future research should focus on developing more advanced models capable of extracting features such as color, shape, texture, and other relevant attributes to enhance defect detection accuracy in coffee beans.

ACKNOWLEDGEMENTS

Authors would like to thank Institut Teknologi Sumatera for providing the research grant (No. 539ba/IT9.2.1/PT.01.03/2024 tanggal 10 June 2024), through “Hibah Penelitian ITERA 2024”.

REFERENCES

[1] Arlius, F., Tjandra, M. A., & Yanti, D. (2017). Analisis Kesesuaian Lahan Untuk Pengembangan Komoditas Kopi Arabika Di Kabupaten Solok. Jurnal Teknologi Pertanian Andalas, 21(1), 70-78.

[2] Rasiska, S., & Khairullah, A. (2017). Efek tiga jenis pohon penaung terhadap keragaman serangga pada pertanaman kopi di Perkebunan Rakyat Manglayang, Kecamatan Cilengkrang, Kabupaten Bandung. Agrikultura, 28(3).

[3] Ardhiarisca, O., & Wijayanti, R. R. (2022). Studi Perbandingan Nilai Ekonomi Kopi Arabika dan Robusta dalam Bisnis Mikro. Jurnal Ilmiah Inovasi, 22(1), 42-50.

[4] Chandra, D., Ismono, R. H., & Kasymir, E. (2013). Prospek perdagangan kopi Robusta Indonesia di pasar internasional. Jurnal Ilmu-Ilmu Agribisnis, 1(1).

[5] Zakaria, A., Aditiawati, P., & Rosmiati, M. (2017). Strategi pengembangan usahatani kopi arabika (kasus pada petani kopi di Desa Suntenjaya Kecamatan Lembang Kabupaten Bandung Barat, Provinsi Jawa Barat). Jurnal sosioteknologi, 16(3), 325-339.

[6] Mawardi, I., Hanif, H., Zaini, Z., & Abidin, Z. (2019). Penerapan teknologi tepat guna pascapanen dalam upaya peningkatan produktifitas petani kopi di Kabupaten Bener Meriah. CARADDE: Jurnal Pengabdian Kepada Masyarakat, 1(2), 205-213.

[7] Ikhsan, D., Utami, E., & Wibowo, F. W. (2020). Metode Klasifikasi Mutu Greenbean Kopi Arabika Lanang Dan Biasa Menggunakan K-Nearest Neighbor Berdasarkan Bentuk. Jurnal Ilmiah SINUS, 18(2), 1-8.

[8] Mardisa, R., Siregar, K., & Nasution, I. S. (2022). Klasifikasi Kualitas Fisik Kopi Beras Arabika Menggunakan Pengolahan Citra Dengan Metode K-Nearest Neighbor (K-NN). Jurnal ilmiah mahasiswa pertanian, 7(2), 514-522.

[9] Ilhamsyah, I., Rahman, A. Y., & Istiadi, I. (2021). Klasifikasi Kualitas Biji Kopi Menggunakan Multilayer Perceptron Berbasis Fitur Warna LCH. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 5(6), 1008-1017.

[10] Putri, D. A., Munawar, A. A., & Nasution, I. S. (2022). Klasifikasi Mutu Fisik Biji Kopi Beras Robusta menggunakan Pengolahan Citra Digital. Jurnal Ilmiah Mahasiswa Pertanian, 7(2), 490-498.

[11] Santos, F. F. L. D., Rosas, J. T. F., Martins, R. N., Araújo, G. D. M., Viana, L. D. A., & Gonçalves, J. D. P. (2020). Quality assessment of coffee beans through computer vision and machine learning algorithms.

[12] Janandi, R., & Cenggoro, T. W. (2020, August). An implementation of convolutional neural network for coffee beans quality classification in a mobile information system. In 2020 International Conference on Information Management and Technology (ICIMTech) (pp. 218-222). IEEE.

[13] Hakim, M., Djatna, T., & Yuliasih, I. (2020, October). Deep learning for roasting coffee bean quality assessment using computer vision in mobile environment. In 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 363-370). IEEE.

[14] Luis, V. A. M., Quiñones, M. V. T., & Yumang, A. N. (2022, September). Classification of Defects in Robusta Green Coffee Beans Using YOLO. In 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) (pp. 1-6). IEEE.

[15] Redmon, J. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition.