Comparative Analysis of CNN and YOLO for Aromatic Leaf Detection on Android-Based Deep Learning Applications

Yuli Safrina [1]*, Muhammad Fikry [2]*, Mukhlis Abd Muthalib [3]*
* Departement of Information Technology, Universitas Malikussaleh, Lhokseumawe, Indonesia
yulisafrina7@gmail.com [1], muh.fikry@unimal.ac.id [2], muchlis.abd@unimal.ac.id [3]

Article Info

Article history:

Received 2026-05-16
Revised 2026-06-11
Accepted 2026-06-12

Keyword:

Deep Learning,
Convolutional Neural Network,
YOLOv11,
Aromatic Leaf Detection,
Image Classification,
Android.

ABSTRACT

Commonly used aromatic leaves in Indonesian cuisine include bay leaves (*Syzygium polyanthum*), pandan leaves (*Pandanus amaryllifolius*), lime leaves (*Citrus hystrix*), curry leaves (*Murraya koenigii*), and turmeric leaves (*Curcuma longa*). Their similar shapes, colors, and textures often make manual identification difficult. Therefore, deep learning technology can be utilized to automatically identify and detect aromatic leaf types through digital images. This study aims to analyze the performance of a Convolutional Neural Network (CNN) using the EfficientNet-B0 architecture and the YOLOv11 model with the AdamW optimizer in detecting and classifying aromatic leaves. The system is implemented using a Python Flask framework for the web based backend and Flutter for the mobile application interface on Android devices. The dataset used in this study consists of 671 digital images obtained through direct image collection and supporting datasets. The dataset is categorized into five classes: bay leaf, pandan leaf, lime leaf, curry leaf, and turmeric leaf. Furthermore, the dataset is divided into training data (89%), validation data (7%), and testing data (4%). The results show that the YOLOv11 model outperforms the CNN (EfficientNet-B0) model. YOLOv11 achieved a precision of 73.36%, recall of 84.35%, mAP50 of 83.93%, and mAP50-95 of 71.38%. Meanwhile, EfficientNet-B0 achieved a best validation accuracy of 81.40% and a test accuracy of 62.07%. Based on experimental results, YOLOv11 demonstrates higher detection confidence and more consistent performance compared to EfficientNet-B0. In addition, YOLOv11 is more suitable for mobile deployment due to its real-time object detection capability with faster inference speed, while the system is supported by a Flask based backend and a Flutter mobile application interface.

I. INTRODUCTION

Indonesian cuisine is known for its diverse spices and aromatic leaves, which are the hallmarks of each dish. Leaves are plant parts often used as spices, primarily to enhance the flavor and aroma of food. Commonly used leaves include lime leaves, bay leaves, celery, and pandan leaves[1]. These leaves are able to enrich the taste so they are usually added as a spice in cooking, not only as a food additive, these leaves with a distinctive aroma are often used as herbal medicine because they contain bioactive compounds[2]. However, for the general public, it is very difficult to differentiate between the types of aromatic leaves used in cooking.

Technological developments in recent years have had a major impact on various sectors, artificial intelligence (AI) has become a hot topic that is widely applied in various industrial sectors, including agriculture[3]. One of the rapidly developing branches of artificial intelligence (AI) today is deep learning, thanks to its capabilities in classifying objects in images or recognizing patterns. One of the most effective algorithms for image detection and processing is You Only Look Once (YOLO). This algorithm uses a Convolutional Neural Network (CNN)-based object detection approach that can learn patterns and features from image data[4].

In today's digital era, the use of digital images is increasingly widespread and significant. Better digital technology and internet connectivity, and the increasing use of social media have resulted in rapid growth in the use and exchange of digital images[5]. Therefore, Android as an application base has the largest number of users, including Indonesia. Deep learning modeling, especially Convolutional Neural Network (CNN), plays an important role in various real-world recognition tasks, including image labeling, multi-language classification, and language translation[6]. Meanwhile, You Only Look Once (YOLO) is able to detect quickly and accurately, and create real-time detection applications[7].

The development of deep learning technology is one of the areas of AI that is in the spotlight because of its ability to recognize objects[8]. One method widely used in image recognition is the Convolutional Neural Network (CNN). CNN is included in Neural Network, which is one of the deep learning neural network architectures that can be said to be inspired by the human brain. One CNN architecture that is quite efficient is EfficientNet, which is able to improve classification performance by using more optimal parameters [9]. CNN has proven to be the best learning algorithm to understand the information available in the image and is considered as an ideal model for various image related tasks like segmentation, classification, tagging, detection and others[10].

Besides CNN, the You Only Look Once (YOLO) method is also widely used in object detection research[11]. YOLO has the advantage of detecting objects in real-time with a fast and accurate process[12]. This study uses YOLOv11 with the AdamW optimizer to improve the stability of the training process and model performance. YOLO is not only capable of classifying objects but also detecting object locations directly in images[13][14]. In this study, a comparison of two methods was conducted to analyze the performance of a Convolutional Neural Network (CNN) for image classification and YOLOv11 for object detection in identifying aromatic leaf images used in cooking. The CNN model is applied for classification tasks to determine the type of leaf based on visual features, while YOLOv11 is used for object detection, which not only identifies the leaf category but also localizes the object within the image using bounding box coordinates. Furthermore, the robustness of both models is evaluated to assess their ability to maintain stable and consistent performance under varying image conditions, such as differences in lighting, background complexity, and object orientation. The objective of this study is to determine the most effective and reliable model for aromatic leaf identification based on accuracy, confidence level, and robustness across different test scenarios.

Previous research developed an EfficientNet-B0 Deep Learning model for tomato leaf disease detection and integrated it into a Flutter-based mobile application. The model was optimized using quantization and pruning techniques to improve efficiency on limited-resource devices. The results showed high performance with an accuracy of 99.55%, a loss value of 0.0939, and real-time detection speeds ranging from 0.150 to 0.554 seconds. This study contributed to the implementation of artificial intelligence in agriculture through mobile-based plant disease detection[15]. Previous studies showed that YOLO11 and YOLOv8 achieved better performance in tomato leaf disease detection compared to CNN-based models such as VGG-16 and Inception-V3. YOLO11 obtained the highest accuracy of 99.4% and demonstrated superior real-time detection speed, making it more effective for field applications [16].

Based on this background, this study aims to compare the performance of the Convolutional Neural Network (CNN) and You Only Look Once (YOLO) methods in an Android-based aromatic leaf detection and classification system. The development of artificial intelligence demands object identification methods that are faster, more accurate, and more efficient than manual processes that are prone to subjectivity, limited knowledge, and characteristic similarities between objects. In addition, a standardized image dataset is needed so that it can be processed optimally using deep learning with real-time detection capabilities. The use of images with simple backgrounds is also necessary to improve accuracy and reduce computational complexity. This study compares the two methods based on accuracy, speed, and detection precision, with the hope of producing an effective and applicable system on mobile devices to support aromatic leaf identification in the culinary, agricultural, and educational fields. YOLOv11 was selected for comparison with CNN because it is capable of performing object detection and classification simultaneously in real time with high speed and accuracy. Compared to conventional CNN classification models, YOLOv11 has advantages in detecting object locations directly through bounding boxes while maintaining efficient computational performance. In addition, YOLOv11 is optimized for lightweight and fast processing, making it more suitable for Android-based mobile applications that require responsive real-time detection.

II. METHODOLOGY

A. Research Stages

This research requires several stages that will significantly impact the process and results. Therefore, the research stages need to be structured systematically and planned to facilitate the researcher in carrying out each step. A well-organized research stage will help researchers obtain focused and effective results that align with the stated objectives. In this study, the focus is directed toward culinary applications and culinary education, particularly in identifying aromatic leaves commonly used in cooking. The development of this system is expected to support learning in culinary fields by helping users recognize and differentiate ingredients more easily and accurately. The research stage flow is shown in Figure 1 below.

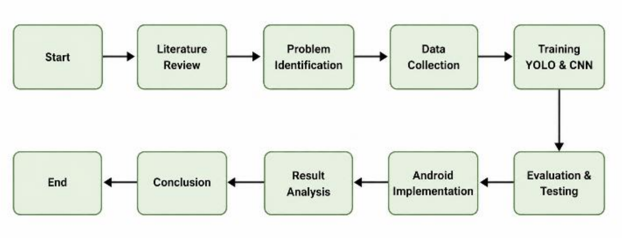

Figure 1. Research Stages

B. Data collection

Data collection is a crucial stage in this research, aiming to obtain relevant, accurate, and accountable data that meets the research objectives. The data collection process is conducted in two parts: primary data (direct observation and image documents) and secondary data (literature sources, journals, and references supporting the theory and methods). The following data collection procedures are shown in Table 1 below.

TABLE 1.
DATA COLLECTION PROCEDURES

Activity Stage	Activity Description	Result
Data Collection Preparation	Determining the types of aromatic leaves used as research objects (pandan, curry, bay, kaffir lime, and turmeric leaves).	List of research objects and equipment ready for use.
Data Acquisition	Capturing images of aromatic leaves in various locations such as markets, gardens, and home environments under different angles and lighting conditions.	A total of 671 image datasets in .jpg/.png format collected.
Data Storage	Storing all captured images into folders based on leaf type and assigning structured file names.	Image data organized systematically by class for processing.
Data Labeling	Labeling each image according to leaf type using LabelImg or Roboflow tools.	Labeled dataset ready for CNN and YOLO training.
Dataset Splitting	Dividing dataset into 89% training, 7% validation, and 4% testing.	Dataset prepared for training and evaluation.
Data Validation	Rechecking image quality to remove blurred, duplicate, or corrupted images.	Clean and valid dataset for model training.

C. Convolutional Neural Network (CNN) algorithm

Convolutional Neural Network (CNN) was first introduced around 1998 and has since developed rapidly[10]. A CNN is a feed-forward deep neural network architecture consisting of multiple convolutional layers, each followed by a pooling layer, an activation function, and optional batch normalization. It also consists of fully connected layers. As the image moves through the network, it becomes smaller, largely due to max-pooling. The final layer outputs the predicted class probabilities[17]. CNN architecture plays an important role in the design of artificial neural network architecture, because a more rational network architecture can enhance the compatibility effect between layers or reduce redundant computation in the network, which usually indicates that it can produce superior performance[18].

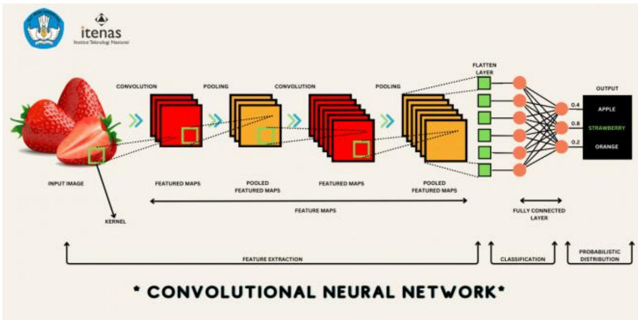

Figure 2. CNN Architecture

The past few years have seen the development of many architectures that have made tremendous progress in the field of image classification. Award-winning pretrained networks (VGG16, VGG19, ResNet50, Inception V3, and Xception) have been used for various image classification challenges, including medical imaging. Transfer learning is a type of practice where you use a pretrained model with multiple layers. It can be used to solve image classification challenges in any field[17]. Since 1989, CNNs have undergone extensive improvements and refinements. These improvements include parameter tuning and optimization, regularization, architectural depth, and more. The improvements that most significantly impact CNN performance include processing unit reordering, network depth, and spatial exploitation[19].

D. EfficientNet-B0

EfficientNet-B0 is part of the EfficientNet family of models, which are designed to be scalable in terms of complexity. Models in the EfficientNet series include variants such as B1, B2, up to B7, with increasing levels of complexity. This scalability allows the architecture to be adapted to different datasets and computational requirements. EfficientNet-B0 also contains convolutional layers capable of extracting important image features at various levels of complexity. This makes it suitable for image classification tasks where features such as color, texture, and shape are important indicators. In this study, the architecture is applied for identifying different types of objects based on their visual characteristics[20].

The CNN model in this study uses the EfficientNet-B0 architecture as the backbone network. The training process is implemented using a custom configuration function with clearly defined hyperparameters to ensure reproducibility. The model is trained with the following settings: 50 epochs, batch size of 16, and an initial learning rate of 0.001. The input image size is standardized to 224 × 224 pixels. A pretrained weight initialization is not used pretrained False, and a dropout rate of 0.5 is applied to reduce overfitting. The training results are saved in a designated directory for evaluation purposes. For optimization, the model uses an adaptive training strategy with early stopping patience 10 to prevent overtraining and improve generalization performance.

E. You Only Look Once (YOLOv11) Algorithm

YOLO11 was released by Ultralytics on September 10, 2024, delivering excellent accuracy, speed, and efficiency. Building upon the impressive advancements of previous YOLO versions, YOLO11 introduces significant improvements in architecture and training methods, making it a versatile choice for a wide range of computer vision tasks. For the latest Ultralytics model with end-to-end NMS-free inference and optimized edge deployment, see YOLO26 [21].

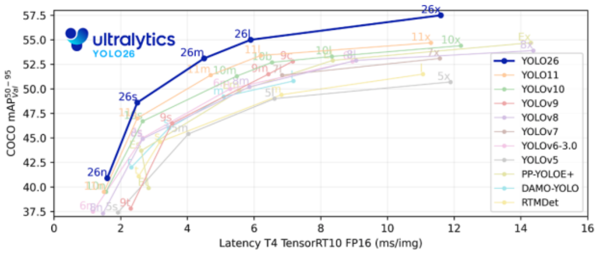

Figure 3. Benchmark Performance Evaluation Results Of The YOLO Algorithm On Various Datasets.

There are several main features of Ultralytics YOLOv11 as follows[21]:

1. Enhanced Feature Extraction YOLO11 employs an improved backbone and neck architecture, which enhances feature extraction capabilities for more precise object detection and complex task performance.
2. Optimized for Efficiency and Speed YOLO11 introduces refined architectural designs and optimized training pipelines, delivering faster processing speeds and maintaining an optimal balance between accuracy and performance.
3. Greater Accuracy with Fewer Parameters With advancements in model design, YOLO11m achieves a higher mean Average Precision (mAP) on the COCO dataset while using 22% fewer parameters than YOLOv8m, making it computationally efficient without compromising accuracy.
4. Adaptability Across Environments YOLO11 can be seamlessly deployed across various environments, including edge devices, cloud platforms, and systems supporting NVIDIA GPUs, ensuring maximum flexibility.
5. Broad Range of Supported Tasks Whether it's object detection, instance segmentation, image classification, pose estimation, or oriented object detection (OBB), YOLO11 is designed to cater to a diverse set of computer vision challenges.

E. Implementation of CNN and YOLOv11

The CNN and YOLOv11 methods were implemented to automatically identify and classify aromatic leaves using deep learning techniques. CNN focuses on feature extraction and image classification, while YOLOv11 performs real-time object detection with high speed and accuracy. Both methods were then evaluated to compare their performance in detecting aromatic leaf types. The following describes the process of both the CNN and YOLOv11 methods in Figure 4.

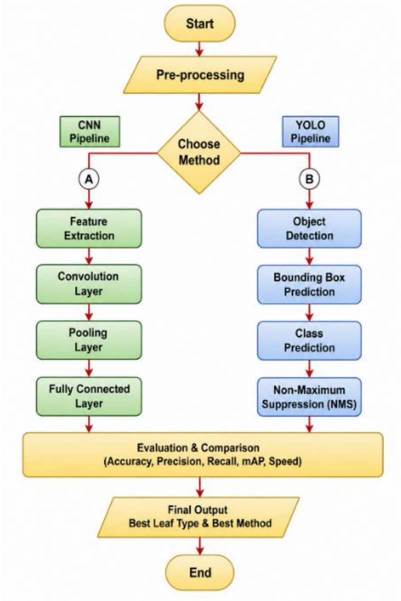

Figure 4. CNN & YOLO Method Flow in the System

The flowchart above explains the implementation process of the CNN and YOLOv11 methods for detecting and classifying aromatic leaves. The process begins with image preprocessing to improve image quality before entering the method selection stage. In the CNN pathway, the system performs feature extraction, convolution, pooling, and a fully connected layer process to classify leaf images. Meanwhile, in the YOLOv11 pathway, the system performs object detection, bounding box prediction, class prediction, and Non-Maximum Suppression (NMS) to detect and recognize objects in real-time. After both methods complete the detection process, the results are evaluated and compared based on accuracy, precision, recall, mAP, and detection speed to determine the best method for recognizing aromatic leaves.

After the implementation and training of the Convolutional Neural Network (CNN) and You Only Look

Once (YOLO) models were completed, the next step was to evaluate their performance. This evaluation aimed to determine the performance, accuracy, and effectiveness of each method in detecting and classifying aromatic leaves in cooking. The training results of both models (CNN and YOLO) were then compared using several evaluation metrics to measure the performance of each method, including[22]:

Accuracy is the level of accuracy of the model in classifying or detecting images correctly.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision is the ratio of true positive predictions to total positive predictions.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall is the ratio of true positive predictions to all actual positive data.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

F1-Score is a harmonious combination of precision and recall.

$$\text{F1-Score} = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision+Recall}} \quad (4)$$

Mean Average Precision (mAP) is used specifically to measure the performance of object detection models such as YOLO, by calculating the average precision at different levels of recall.

mAP Formula:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (5)$$

TABLE 2.
CONFUSION MATRIX

Symbol	Description
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

III. RESULT AND DISCUSSION

This section presents the results obtained from the implementation of the YOLOv11 CNN method for the detection and classification of cooking aromatic leaves based on digital images. The discussion includes evaluation of model performance, analysis of detection results on various types of aromatic leaves, comparison of accuracy and confidence levels of detection between the CNN and YOLOv11 methods, and testing of model performance using test data. In addition, the research results are also analyzed to determine the effectiveness of the system in automatically classifying and detecting leaves on Android devices based on aspects of accuracy, consistency of detection results, and system response speed.

A. Aromatic Leaf Image Dataset

The dataset used in this study was obtained from the Roboflow platform via the Leaf Detection dataset link at https://universe.roboflow.com/detect-rupiah/deteksi-daun-eiqq3/dataset/9.This dataset served as the primary data source for model training, validation, and testing. The number of datasets for each leaf can be seen in Table 3 below:

TABLE 3.
LEAF IMAGE

Leaf Type	Data Format	Number of Images
Bay Leaf	.jpg / .png	200 Images
Curry Leaf	.jpg / .png	200 Images
Pandan Leaf	.jpg / .png	200 Images
Lime Leaf	.jpg / .png	200 Images
Turmeric Leaf	.jpg / .png	200 Images
Total		**1,000 Images**

After going through a selection process and data sharing on the Roboflow platform, the dataset used in this study consisted of 671 images. The entire data was then divided into three parts: the training set, the validation set, and the test set. The training set consisted of 597 images, or approximately 89% of the data, the validation set consisted of 44 images, or approximately 7%, and the test set consisted of 30 images, or approximately 4%. The results of the dataset can be seen in Table 5 as follows:

TABLE 4.
DATASET RESULTS

Dataset Type	Number of Images	Percentage
Training Set	597	89%
Validation Set	44	7%
Test Set	30	4%
Total	**671**	**100%**

The initial image data collected consisted of 1,000 images. However, after a selection process and dataset adjustments on the Roboflow platform, the number of images used in the study decreased to 671. Images that were not used were those that did not meet the research criteria, such as poor image quality, duplicate data, or unclear objects.

B. Preprocessing and Data Augmentation

During the preprocessing stage, all images in the dataset were resized to 224×224 pixels. This size was chosen because it is a standard input dimension commonly used in CNN architectures, which can improve model performance in image classification tasks. The resizing process aims to standardize the input dimensions to match the requirements of the CNN model. Meanwhile, for the YOLO model, the resizing process is automatically handled during training with an input size of 640×640 pixels. With uniform image sizes, the training process becomes more stable and efficient, and it also helps the model to optimally extract features from each image.

The normalization process in this study is applied to the RGB color channels, namely Red (R), Green (G), and Blue

(B), to ensure that the image data is standardized before being used in the model training process. Each channel is normalized based on its mean and standard deviation values. The mean values for the Red, Green, and Blue channels are 0.485, 0.456, and 0.406, respectively, while the standard deviation values are 0.229, 0.224, and 0.225. This normalization helps to scale the pixel intensity values so that they are centered around zero and have a consistent distribution. As a result, the model can learn more effectively by reducing variations in lighting and color intensity across different images. This process also improves training stability and helps the model converge faster during the learning process.

In the data augmentation stage, several image transformation techniques are applied to increase the diversity of the training dataset and help the model become more robust to variations in image conditions. The process begins with resizing the image to a larger size of (img_size + 32), followed by a random crop to the target img_size, which introduces variation in the focused region of the image. Next, random horizontal flipping is applied with a probability of 0.5 to flip images horizontally, enabling the model to recognize objects from different orientations. In addition, random rotation is performed with a maximum angle of 15 degrees to introduce variations in object orientation. The augmentation process also includes color jittering, which adjusts brightness, contrast, and saturation each by 0.2 to simulate different lighting conditions. Furthermore, random affine transformation is applied with translation up to 10% along both horizontal and vertical axes to provide positional variations of objects within the image.

C. Training CNN and YOLOv11

The CNN (Convolutional Neural Network) training process is the stage where the model is trained using a dataset to recognize patterns in images. When the "python cnn_training.py" command is run, the system reads the training data (for example, images of leaves), then processes each image through several CNN layers, including convolution, activation, and pooling, to extract important features.

TABLE 5.
CNN TRAINING

Description	Value
Test Accuracy	62.07%
Best Validation Accuracy	81.40%
Total Parameters	5,038,405

The difference between validation accuracy (81.40%) and test accuracy (62.07%) does not necessarily indicate overfitting. This is because the dataset used in this study is relatively small (671 images), and the test data proportion is only 4%, which may not fully represent the overall data distribution. In addition, variations in image conditions such as lighting, background, and viewing angles can significantly affect model performance during testing. These variations make the test set more challenging compared to the validation set, resulting in lower accuracy. Furthermore, the model shows stable learning behavior during training, as indicated by consistent validation performance. Therefore, the performance gap is more likely caused by dataset limitations and data variability rather than overfitting.

The YOLO model training process begins with preparing a labeled dataset, then training the model to recognize and detect objects in images. At this stage, the model learns object patterns, locations, and characteristics through several training epochs, gradually improving detection accuracy. During the training process, various parameters such as learning rate, batch size, and number of epochs are adjusted to achieve optimal model performance.

TABLE 6.
YOLO TRAINING RESULTS EPOCH 50 BACH 16

Metric	Value
Precision	0.7336
Recall	0.8435
mAP@50	0.8393
mAP@50–95	0.7138
F1-Score	N/A

Based on the training results of the Convolutional Neural Network (CNN) and You Only Look Once (YOLO) models, a comparative evaluation of training time was conducted to determine the efficiency of each model. This comparison aims to analyze how quickly each model learns from the training data, thus identifying the most optimal model in terms of computation time and training efficiency. The results of this comparison are shown in Table 8.

TABLE 7.
COMPARISON OF MODEL TRAINING TIMES

Model	Training Time	Description
YOLO	1 hour 20 minutes	The longer training time is caused by the model learning both object localization and object classification simultaneously. In addition, the image resizing process to a uniform size increases the number of pixels processed, resulting in higher computational cost.
CNN	40–50 minutes	The shorter training time is due to the model performing only image classification. Although image resizing still affects the process, the computational load is lighter compared to YOLO.

C. Evaluation and Testing of CNN and YOLO

The results of the model evaluation and testing were conducted after the training process was completed. The evaluation aimed to determine the model's ability to learn patterns from the data, while testing was conducted to measure the model's performance on previously unused data.

The evaluation and testing results in this study are presented using a Confusion Matrix and graphs of the training and evaluation results.

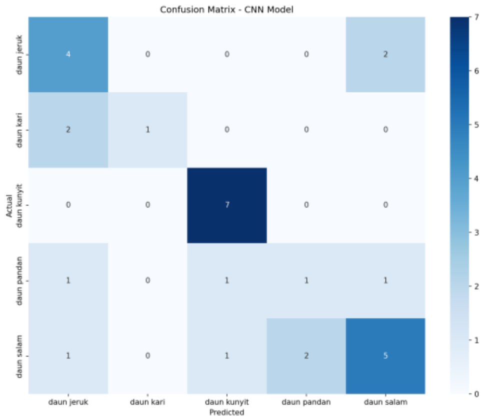

Figure 5. Confusion Matrix CNN

1. Lime Leaves
   TP = 4 , FN = 2 (wrong to bay leaves), FP = 2+1+1 = 4, TN = 30 - (4+2+4) = 20

$$\text{Precision} = \frac{4}{4+4} = \frac{4}{8} = 0{,}5000 = 0{,}50$$

$$\text{Recall} = \frac{4}{4+2} = \frac{4}{6} = 0{,}6667 = 0{,}67$$

$$\text{F1-score} = \frac{2 \times (0{,}50 \times 0{,}67)}{0{,}50 + 0{,}67} = \frac{2 \times (0{,}3333)}{1{,}1667} = 0{,}5714 = 0{,}57$$

2. Curry Leaves
   TP = 1, FN = 2, FP = 0, TN = 27

$$\text{Precision} = \frac{1}{1+0} = \frac{1}{1} = 1{,}0000 = 1{,}00$$

$$\text{Recall} = \frac{1}{1+2} = \frac{1}{3} = 0{,}3333 = 0{,}33$$

$$\text{F1-score} = \frac{2 \times (1{,}0000 \times 0{,}3333)}{(1{,}0000 + 0{,}3333)} = \frac{0{,}6667}{1{,}3333} = 0{,}5000 = 0{,}50$$

3. Turmeric Leaves
   TP = 7, FN = 0, FP = 2 (from pandan leaves & bay leaves), TN = 21

$$\text{Precision} = \frac{7}{7+2} = \frac{7}{9} = 0{,}7778 = 0{,}78$$

$$\text{Recall} = \frac{7}{7+0} = \frac{7}{7} = 1{,}0000 = 1{,}00$$

$$\text{F1-score} = \frac{2 \times (0{,}7778 \times 1{,}0000)}{(0{,}7778 + 1{,}0000)} = \frac{1{,}5556}{1{,}7778} = 0{,}8750 = 0{,}88$$

4. Pandan Leaves
   TP = 1, FN = 3, FP = 2, TN = 24

$$\text{Precision} = \frac{1}{1+2} = \frac{1}{3} = 0{,}3333 = 0{,}33$$

$$\text{Recall} = \frac{1}{1+3} = \frac{1}{4} = 0{,}2500 = 0{,}25$$

$$\text{F1-score} = \frac{2 \times (0{,}3333 \times 0{,}2500)}{(0{,}3333 + 0{,}2500)} = \frac{0{,}1667}{0{,}5833} = 0{,}2857 = 0{,}29$$

5. Bay leaf
   TP = 5, FN = 4, FP = 3, TN = 18

$$\text{Precision} = \frac{5}{5+3} = \frac{5}{8} = 0{,}6250 = 0{,}63$$

$$\text{Recall} = \frac{5}{5+4} = \frac{5}{9} = 0{,}5556 = 0{,}56$$

$$\text{F1-score} = \frac{2 \times (0{,}6250 \times 0{,}5556)}{(0{,}6250 + 0{,}5556)} = \frac{0{,}6944}{1{,}1806} = 0{,}5882 = 0{,}59$$

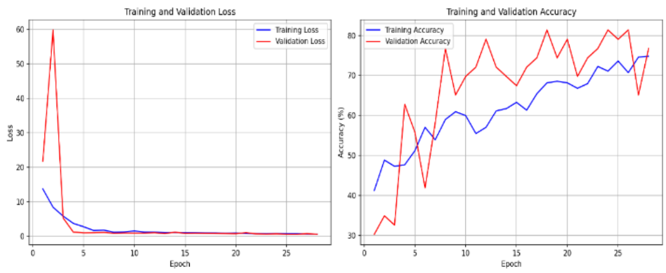

Figure 6. Training and Validation Results of CNN

The training and validation loss show that the training loss (blue line) decreases significantly during the early epochs, dropping from around 13 to nearly 1, and then remaining consistently low. Meanwhile, the validation loss (red line) starts at a very high value of approximately 60, but drops sharply within the first few epochs and stabilizes close to zero afterward, indicating improved model performance over time.

In terms of accuracy, the training accuracy (blue line) gradually increases from about 41% to around 75%, showing consistent learning progress during training. The validation accuracy (red line), although slightly more fluctuating, generally shows an upward trend, increasing from approximately 30% to a range of 75–80%. This indicates that the model is able to generalize well to unseen data after several training epochs.

After the training process was completed, the YOLO model was evaluated and tested using a Confusion Matrix to determine the accuracy of the model's predictions for each leaf class. Furthermore, the evaluation results were presented through a graph of the training and model evaluation results, which was used to track the model's performance during each epoch of training.

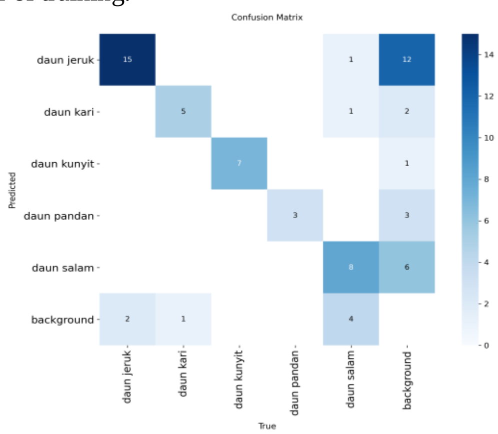

Figure 7. Confusion Matrix YOLO

1. Lime Leaves
TP = 15, FN = 7, FP = 1 + 12 = 13, TN = 76 − (15 + 7 + 13) = 41

$$\text{Precision} = \frac{15}{15 + 13} = \frac{15}{28} = 0{,}5357 = 0{,}54$$

$$\text{Recall} = \frac{15}{15 + 7 =} = \frac{15}{22} = 0{,}6818 = 0{,}68$$

$$\text{F1-score} = \frac{2 \times (0{,}54 \times 0{,}68)}{0{,}54 \times 0{,}68} = \frac{0{,}7344}{1{,}22} = ,0{,}6020 = 0{,}60$$

2. Curry Leaves
TP = 5, FN = 1, FP = 1 + 2 = 3, TN = 76 − (5 + 1 + 3) = 67

$$\text{Precision} = \frac{5}{5 + 3} = \frac{5}{8} = 0{,}6250 = 0{,}63$$

$$\text{Recall} = \frac{5}{5 + 1 =} = \frac{5}{6} = 0{,}8333 = 0{,}83$$

$$\text{F1-score} = \frac{2 \times (0{,}63 \times 0{,}83)}{0{,}63 \times 0{,}83} = \frac{1{,}0458}{1{,}46} = 0{,}7163 = 0{,}72$$

3. Turmeric Leaves
TP = 7, FN = 0, FP = 1, TN = 76 − (7 + 0 + 1) = 68

$$\text{Precision} = \frac{7}{7 + 1} = \frac{7}{8} = 0{,}8750 = 0{,}88$$

$$\text{Recall} = \frac{7}{7 + 0 =} = \frac{7}{7} = 1{,}0000 = 1{,}00$$

$$\text{F1-score} = \frac{2 \times (0{,}88 \times 1{,}00)}{0{,}88 \times 1{,}00} = \frac{1{,}7600}{1{,}88} = 0{,}9362 = 0{,}94$$

4. Pandan Leaves
TP = 3, FN = 0, FP = 3, TN = 76 − (3 + 0 + 3) = 70

$$\text{Precision} = \frac{3}{3 + 3} = \frac{3}{6} = 0{,}5000 = 0{,}50$$

$$\text{Recall} = \frac{3}{3 + 0 =} = \frac{3}{3} = 1{,}0000 = 1{,}00$$

$$\text{F1-score} = \frac{2 \times (0{,}50 \times 1{,}00)}{0{,}50 \times 1{,}00} = \frac{1{,}0000}{1{,}50} = 0{,}6667 = 0{,}67$$

5. Bay leaf
TP = 8, FN = 6, FP = 1, TN = 76 − (8 + 6 + 1) = 61

$$\text{Precision} = \frac{8}{8 + 1} = \frac{8}{9} = 0{,}8889 = 0{,}89$$

$$\text{Recall} = \frac{8}{8 + 6 =} = \frac{8}{14} = 0{,}5714 = 0{,}57$$

$$\text{F1-score} = \frac{2 \times (0{,}89 \times 0{,}57)}{0{,}89 \times 0{,}57} = \frac{1{,}0146}{1{,}46} = 0{,}6949 = 0{,}69$$

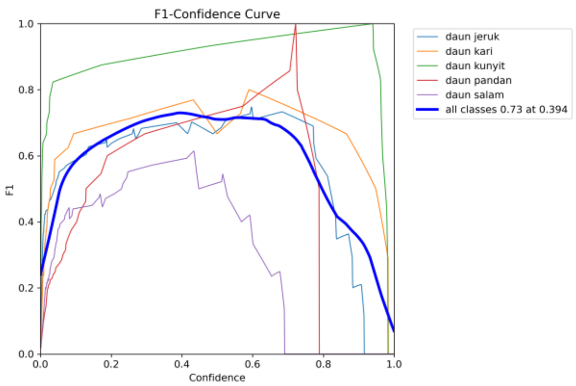

Figure 8. F1–Confidence Curve of YOLOv11 Model

F1-Confidence Curve illustrates the relationship between the confidence threshold (0–1) and the F1-score for a leaf classification model. The x-axis represents the confidence threshold, while the y-axis shows the F1-score, which balances precision and recall. Each line represents a different leaf class (daun jeruk, daun kari, daun kunyit, daun pandan, and daun salam), and the bold blue line indicates overall performance. The model achieves its best performance at a confidence threshold of around 0.394, with an F1-score of 0.73, indicating the optimal balance between precision and recall. Lower thresholds increase false positives, while higher thresholds reduce recall. Among the classes, daun kunyit performs the most consistently and achieves the highest F1-score, whereas daun salam shows lower and more unstable performance. Overall, the model performs well at the optimal threshold.

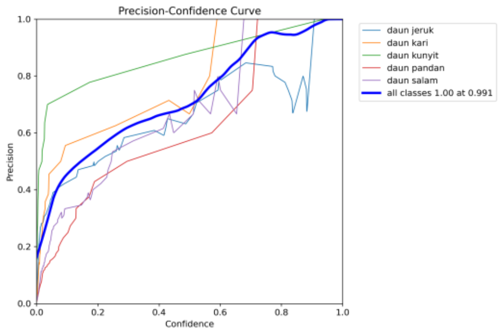

Figure 9. Precision Confidence Curve of YOLOv11 Model

The graph shows the relationship between precision and confidence for a leaf classification model. As the confidence threshold increases, the precision generally improves, meaning the predictions become more accurate. Classes such as *turmeric leaves* and *curry leaves* achieve high precision faster, while *lime leaves* and *pandan leaves* show more fluctuations. Overall, the model performs well, with the average precision reaching nearly 1.00 at a confidence level of around 0.991.

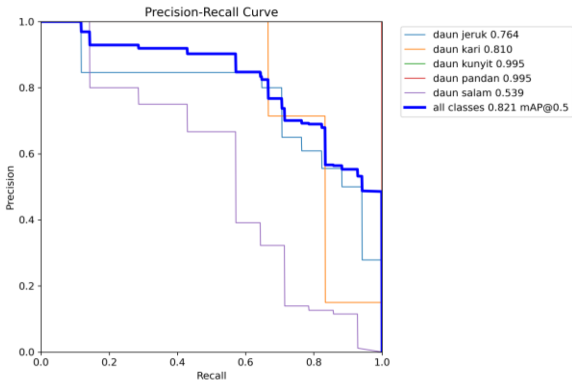

Figure 10. Precision–Recall Curve of YOLOv11 Model

The graph shows the Precision–Recall Curve of the leaf classification model. It illustrates the relationship between precision and recall for each leaf type, namely *lime leaves*, *curry leaves*, *turmeric leaves*, *pandan leaves*, and *Indonesian bay leaves*. Overall, the model performs well with an average mAP@0.5 of 0.821. *Pandan leaves* achieve the best performance with a score of 0.995, while *Indonesian bay leaves* have the lowest performance with 0.539, indicating that some leaf types are classified more accurately than others.

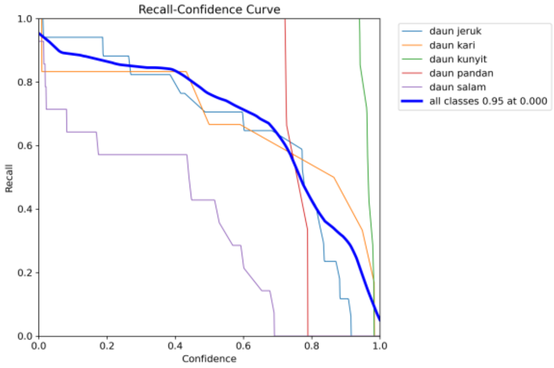

Figure 11. Recall Curve of YOLOv11 Model

The graph shows the relationship between recall and confidence for the leaf classification model. As the confidence threshold increases, the recall generally decreases, meaning the model detects fewer objects at higher confidence levels. Classes such as *turmeric leaves* maintain high recall for longer confidence ranges, while *Indonesian bay leaves* show a faster decline in recall. The thick blue line represents the average performance of all classes, with an overall recall of about 0.95 at a very low confidence threshold. Overall, the graph indicates that increasing confidence improves prediction certainty but reduces the model's ability to detect all objects.

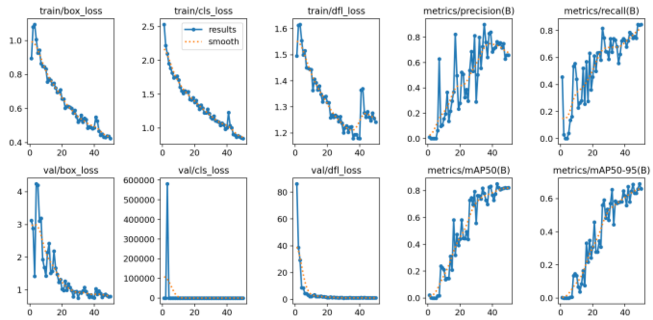

Figure 12. Training and Validation Results of YOLOv11 Model

Figure 12 shows the training and evaluation results of the YOLO model during the training process. Overall, the training results graph shows that the model experienced consistent performance improvements throughout the training process. The training loss values (box, cls, and dfl) decreased gradually, indicating that the model was increasingly able to learn object positions, classify them, and predict bounding boxes more accurately. Furthermore, the precision and recall values also tended to increase, indicating that the model was becoming more accurate in its predictions and better at detecting objects that were actually present in the image. In the validation phase, the loss values (box, cls, and dfl) also showed a downward trend, despite some fluctuations at the beginning of each epoch. This indicates that the model was becoming more stable and able to perform well on previously unseen data. This is reinforced by the increase in mAP50 and mAP50-95 values, indicating that the model's ability to detect objects improved with increasing epochs.

D. Android View

In the implementation stage, the detection model that has gone through the training process is then integrated into an Android-based application. The following is a display of the Android interface.

Figure 13. Initial View

In figure 13 is the initial display of the android interface. Then select the gallery image to be able to perform the next stage, namely image input.

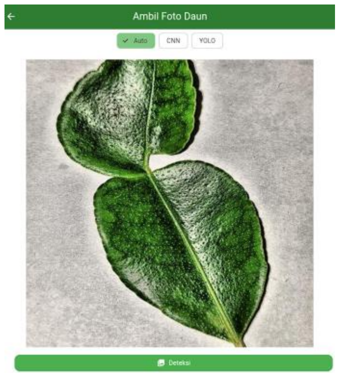

Figure 14. Image Input Process

E. android detection test data results

This study successfully carried out testing using 10 leaf samples for each type of leaf in an Android-based detection system to evaluate the performance of the CNN and YOLO models.

1. Results of The Lime Leaf Test Data

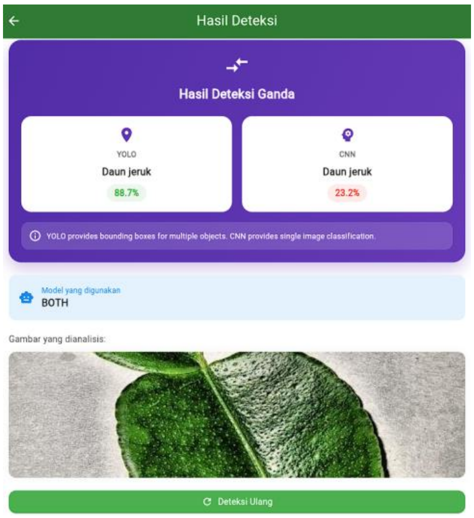

Figure 15. Lime Leaves Detection

TABLE 8.
LIME LEAVES ANDROID DETECTION TEST DATA SAMPLE

Leaf Type	YOLO Confidence & Correctness	CNN Confidence & Correctness
Lime Leaf	Correct – 88.7%	Correct – 23.2%
Lime Leaf	Correct – 78.4%	Correct – 21.7%
Lime Leaf	Correct – 86.6%	Correct – 39.7%
Lime Leaf	Correct – 75.7%	Correct – 35.7%
Lime Leaf	Correct – 85.7%	Correct – 38.0%
Lime Leaf	Correct – 66.2%	Correct – 42.0%
Lime Leaf	Correct – 60.5%	Incorrect – 41.6%
Lime Leaf	Correct – 57.6%	Correct – 29.6%
Lime Leaf	Correct – 46.8%	Incorrect – 26.2%
Lime Leaf	Correct – 93.6%	Incorrect – 29.9%

Based on the results, the YOLO model performs better than the CNN model in citrus leaf detection. Both models were tested using 10 images. YOLO correctly classified all samples, achieving 100% accuracy, while CNN correctly classified 7 samples and misclassified 3, resulting in 70% accuracy. In terms of confidence, YOLO shows higher and more stable values, with a maximum of 93.6%, a minimum of 46.8%, and an average of 74.0%. In contrast, CNN has much lower confidence, with a maximum of 42.0%, a minimum of 21.7%, and an average of 32.8%. This shows that YOLO produces more reliable and confident predictions.

From the table, it can be seen that some CNN detections fail because the model is more sensitive to variations in the input images, such as angle, lighting, blur, and leaf shape differences. CNN relies heavily on detailed features, so when the image is unclear or the leaf looks similar to another class, the prediction can become less accurate or even incorrect. In addition, limited or less diverse training data may reduce the model's ability to generalize well to new images. Compared to YOLO, which is more stable in detecting objects as a whole, CNN tends to produce lower confidence values and is more prone to errors under challenging image conditions.

2. Results Bay Leaf Detection Test Data

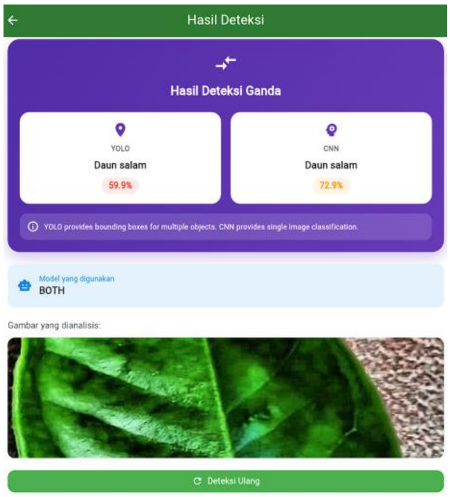

Figure 16. Bay Leaf Detection

TABLE 9.
BAY LEAVES ANDROID DETECTION TEST DATA SAMPLE

Leaf Type	YOLO Confidence & Correctness	CNN Confidence & Correctness
Bay Leaf	Correct – 52.2%	Correct – 26.9%
Bay Leaf	Correct – 59.9%	Correct – 72.9%
Bay Leaf	Incorrect – 63.0%	Correct – 100.0%
Bay Leaf	Incorrect – 72.6%	Incorrect – 34.5%
Bay Leaf	Correct – 48.5%	Correct – 100.0%
Bay Leaf	Incorrect – 41.8%	Correct – 100.0%
Bay Leaf	Correct – 42.8%	Incorrect – 30.3%
Bay Leaf	Correct – 30.6%	Incorrect – 23.5%
Bay Leaf	Incorrect – 78.0%	Correct – 94.1%
Bay Leaf	Incorrect – 80.01%	Correct – 100.0%

Based on the results, the CNN model performs better than YOLO in the bay leaf detection task. Both models were tested using 10 images. CNN correctly classified 7 images and misclassified 3, achieving 70% accuracy, while YOLO correctly classified 5 images and misclassified 5, with 50% accuracy. In terms of confidence, CNN achieved a higher maximum value of 100.0% and an average of 68.22%, while YOLO reached a maximum of 80.01% and an average of 56.94%. CNN also shows slightly lower minimum confidence (23.5%) compared to YOLO (30.6%). Overall, CNN provides more accurate and consistent results for this dataset, while YOLO shows less stable performance in distinguishing bay leaf features.

Some detections fail due to similar leaf shapes, variations in image quality (such as blur and lighting), YOLO can still make mistakes when objects are unclear or similar to other classes, although it is generally more stable. CNN is less consistent because it relies heavily on detailed features, making it more likely to produce incorrect results under challenging image conditions.

3. *Results Turmeric Leaf Detection Test Data*

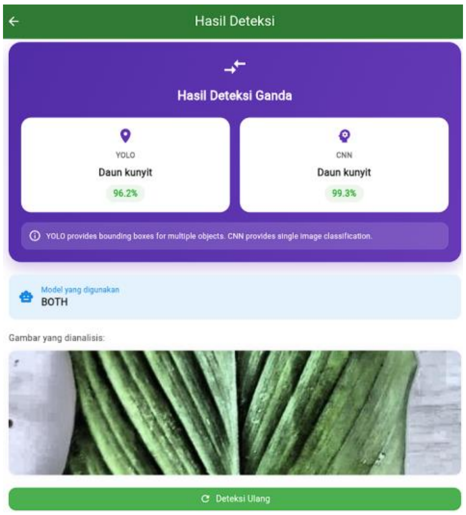

Figure 17. Turmeric Leaf Detection

TABLE 10.
TURMERIC LEAF ANDROID DETECTION TEST DATA SAMPLE

Leaf Type	YOLO Confidence & Correctness	CNN Confidence & Correctness
Turmeric Leaf	Incorrect – 74.4%	Correct – 47.3%
Turmeric Leaf	Correct – 93.9%	Correct – 42.3%
Turmeric Leaf	Correct – 96.2%	Correct – 99.3%
Turmeric Leaf	Correct – 96.4%	Correct – 81.3%
Turmeric Leaf	Correct – 89.3%	Correct – 50.0%
Turmeric Leaf	Correct – 96.4%	Correct – 23.3%
Turmeric Leaf	Incorrect – 67.0%	Correct – 99.9%
Turmeric Leaf	Correct – 95.2%	Correct – 21.2%
Turmeric Leaf	Incorrect – 67.8%	Correct – 28.3%
Turmeric Leaf	Incorrect – 57.8%	Correct – 100.0%

Based on the results, the CNN model performs better than YOLO in turmeric leaf detection. Both models were tested using 10 images. CNN correctly classified all samples, achieving 100% accuracy, while YOLO correctly classified 6 samples and misclassified 4, with 60% accuracy. In terms of confidence, YOLO achieved a higher average of 83.44% with a maximum of 96.4%, while CNN had a lower average of 59.29% but reached a maximum of 100.0%. Overall, despite YOLO having higher confidence values, CNN provides more accurate and consistent detection results for this dataset.

The incorrect detections in the turmeric leaf results are caused by the way YOLO and CNN work and the nature of the dataset. YOLO detects objects as a whole, so it is generally good at recognizing overall leaf patterns, but it can still make mistakes when leaves look similar or image conditions are not clear, even with high confidence. CNN, on the other hand, focuses on detailed features, making it more sensitive to small changes in angle, lighting, and texture, which can lead to inconsistent results. Overall, the errors are mainly due to similar leaf shapes, varying image quality, and limited generalization of both models, where YOLO is more stable in object detection, while CNN depends more on feature consistency.

4. *Results Pandan Leaf Detection Test Data*

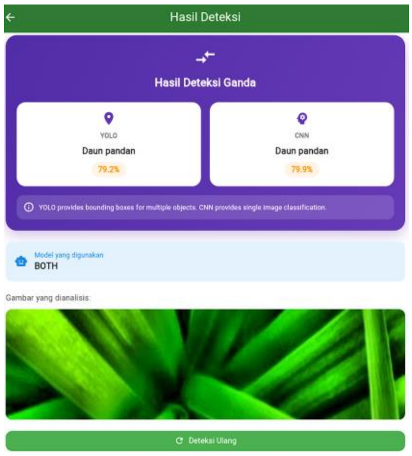

Figure 18. Pandan Leaf Detection

TABLE 11.
PANDAN LEAF ANDROID DETECTION TEST DATA SAMPLE

Leaf Type	YOLO Confidence & Correctness	CNN Confidence & Correctness
Pandan Leaf	Correct – 73.1%	Correct – 58.7%
Pandan Leaf	Correct – 75.9%	Correct – 32.7%
Pandan Leaf	Correct – 64.0%	Incorrect – 43.7%
Pandan Leaf	Correct – 57.2%	Correct – 54.0%
Pandan Leaf	Correct – 65.3%	Incorrect – 28.8%
Pandan Leaf	Correct – 75.3%	Incorrect – 22.9%
Pandan Leaf	Correct – 79.1%	Correct – 87.2%
Pandan Leaf	Correct – 68.4%	Correct – 92.7%
Pandan Leaf	Correct – 79.2%	Correct – 79.9%
Pandan Leaf	Correct – 69.1%	Correct – 53.0%

Based on the results, the YOLO model performs better than the CNN model in pandan leaf detection. Both models were tested using 10 images. YOLO correctly classified all samples, achieving 100% accuracy, while CNN correctly classified 7 samples and misclassified 3, with 70% accuracy. In terms of confidence, YOLO achieved an average of 70.66% (57.2%–79.2%), while CNN had an average of 55.36% (22.9%–92.7%). Although CNN reached a higher maximum confidence, YOLO shows more stable and consistent results because all samples were correctly detected. Overall, YOLO provides more accurate and reliable performance than CNN for this dataset.

Some detections fail due to similar leaf shapes, variations in image quality (blur, lighting, and angle), and limited training data. CNN is more sensitive to small details and can easily make mistakes, while YOLO is more stable but can still fail when the object is unclear or similar to other classes.

5. Results Curry Leaf Detection Test Data

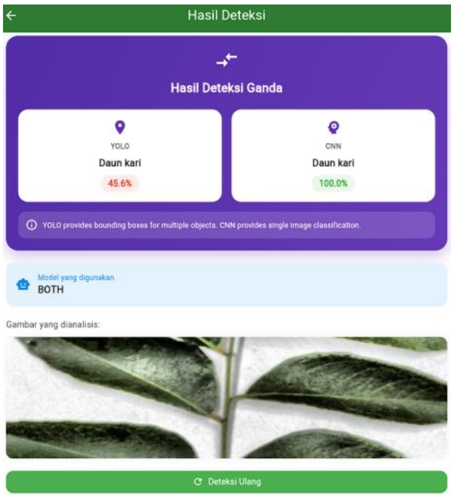

Figure 19. Curry Leaf Detection

TABLE 12.
CURRY LEAF ANDROID DETECTION TEST DATA SAMPLE

Leaf Type	YOLO Confidence & Correctness	CNN Confidence & Correctness
Curry Leaf	Correct – 45.6%	Correct – 100.0%
Curry Leaf	Incorrect – 47.7%	Correct – 67.0%
Curry Leaf	Incorrect – 67.1%	Correct – 100.0%
Curry Leaf	Incorrect – 65.4%	Correct – 100.0%
Curry Leaf	Incorrect – 74.8%	Incorrect – 22.1%
Curry Leaf	Incorrect – 45.5%	Correct – 100.0%
Curry Leaf	Incorrect – 59.5%	Correct – 45.8%
Curry Leaf	Incorrect – 50.5%	Correct – 100.0%
Curry Leaf	Incorrect – 76.8%	Incorrect – 21.5%
Curry Leaf	Incorrect – 59.7%	Incorrect – 21.5%

Based on the results, the CNN model performs better than YOLO in curry leaf detection. Both models were tested using 10 images. CNN achieved 70% accuracy with 7 correct and 3 incorrect predictions, while YOLO achieved 10% accuracy with 1 correct and 9 incorrect predictions. In terms of confidence, YOLO recorded an average of 59.26% (45.5%–76.8%), while CNN achieved a higher average of 67.79% (21.5%–100.0%). Although YOLO had moderate confidence values, most predictions were incorrect, showing low reliability. Overall, CNN provides more accurate and consistent performance than YOLO for this dataset.

Detection failure in curry leaf occurs due to a combination of data factors and model behavior. YOLO detects objects as a whole, but it can make mistakes when leaf shapes are similar to other classes or when images are unclear. CNN extracts detailed features, making it more sensitive to small changes such as lighting, angle, and texture, which can lead to errors if the training data is not diverse enough. In addition, limited training data reduces the ability of both models to generalize well to new images.

IV. CONCLUSION

Based on the results and discussion, this study successfully implemented and compared the performance of Convolutional Neural Network (CNN) using the EfficientNet-B0 architecture and YOLOv11 in an Android-based aromatic leaf detection system. The system was developed to identify five types of aromatic leaves commonly used in Indonesian cuisine, namely bay leaves, pandan leaves, lime leaves, curry leaves, and turmeric leaves. The implementation utilized a Flask-based backend and a Flutter mobile interface to support real-time detection on Android devices. The experimental results show that YOLOv11 generally provides better overall performance compared to the CNN model. YOLOv11 achieved a precision of 73.36%, recall of 84.35%, mAP@50 of 83.93%, and mAP@50–95 of 71.38%, demonstrating strong capability in object detection and localization. Meanwhile, the CNN EfficientNet-B0 model achieved a best validation accuracy of 81.40% and a test accuracy of 62.07%. The results indicate that YOLOv11 is more effective for real-time object detection tasks because it can simultaneously classify and localize objects with higher confidence and more stable predictions.

The Android-based testing using sample leaf images further demonstrated the strengths and weaknesses of each model. YOLOv11 showed excellent performance in detecting lime leaves and pandan leaves, achieving 100% detection accuracy in several test scenarios with stable confidence values. On the other hand, CNN performed better for bay leaves, turmeric leaves, and curry leaves, where detailed feature extraction helped improve classification accuracy in certain conditions. However, CNN predictions were more sensitive to image variations such as lighting, blur, viewing angle, and background complexity, resulting in lower consistency and confidence in some cases. The evaluation results also reveal that detection errors in both models were mainly caused by similarities in leaf shapes, limited dataset diversity, variations in lighting conditions, object orientation, and image quality.

CNN tends to rely heavily on detailed texture and feature consistency, making it more vulnerable to environmental variations. Meanwhile, YOLOv11 is more stable in detecting whole-object patterns but may still misclassify objects when the visual characteristics between classes are highly similar. Overall, this study concludes that YOLOv11 is more suitable for Android-based aromatic leaf detection applications because it provides better detection stability, higher confidence values, and real-time performance capabilities. Nevertheless, CNN EfficientNet-B0 still demonstrates competitive performance in several classification scenarios and can be considered effective for image classification tasks with controlled image conditions. The integration of deep learning methods into mobile applications proves to be a promising solution for supporting automatic aromatic leaf identification in culinary, agricultural, and educational fields.

References

[1] L. Hakim, *Spices & Herbs In Community Home Gardens*, no. 164. 2022.

[2] T. Muhandri, F. M. Taqi, Subarna, and D. D. Widiawati, "Drying Characteristics of Herb Leaves Using Fluidized Bed Drier and Tray Drier," *J. Teknol. dan Ind. Pangan*, vol. 34, no. 2, pp. 187–199, 2023, doi: 10.6066/jtip.2023.34.2.187.

[3] A. I. Saputra, I. Weni, and U. Khaira, "Implementation of The Convolution Neural Network Method for Disease Detection in Arabica Coffe Plants Use Android Based Leaf Images," *Decod. J. Pendidik. Teknol. Inf.*, vol. 4, no. 1, pp. 41–51, 2023, doi: 10.51454/decode.v4i1.231.

[4] M. Ibrahim and U. Latifa, "Application of The Yolo-v8 Algorithm in Website-Based Detection of Harvest Time of Pakcoy Crops," *J. Mhs. Tek. Inform.*, vol. 7, no. 4, pp. 2489–2495, 2023.

[5] M. Mufid Bachri, E. D. Astuti, and H. Sibyan, "Clean Energy and Smart Technology Image Classification Recognition Of Gamelan Musical Instrument Types Using Cnn Method Android Based," vol. 02, no. 02, pp. 56–70, 2024, [Online]. Available: https://doi.org/10.58641/

[6] M. A. A. Mosleh and A. H. Gumaei, "An Efficient Bidirectional Android Translation Prototype for Yemeni Sign Language Using Fuzzy Logic and CNN Transfer Learning Models," *IEEE Access*, vol. 12, no. November, pp. 191030–191045, 2024, doi: 10.1109/ACCESS.2024.3512455.

[7] M. F. Golfantara, "Use of YOLO Algorithm for Spice Identification," vol. 12, no. 3, pp. 3867–3873, 2024.

[8] N. Lysbetti Marpaung, R. Juan Hendri Butar-Butar, and S. Hutabarat, "(Journal of Informatics Education and Research) Implementation of Deep learning for Identification Using Transfer learning Method," vol. 9, no. 3, pp. 348–354, 2023.

[9] A. K. Putri, S. Suroso, and A. S. Handayani, "Application of EfficiencyNet for Creating CNN Models in Sign Language Classification," *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, pp. 758–766, 2024, doi: 10.47065/bits.v6i2.5592.

[10] F. Marpaung, F. Aulia, and R. C. Nabila, *Computer Vision and Digital Image Processing*. 2022. [Online]. Available: www.pustakaaksara.co.id

[11] S. Poonkuntran, R. K. Dhanraj, and B. Balusamy, *Object Detection with Deep Learning Models: Principles and Applications*. 2022. doi: 10.1201/9781003206736.

[12] F. Jupiter, E. S. Negara, Y. N. Kunang, and M. I. Herdiansyah, "Implementation of CNN and YOLO Algorithms to Detect Vehicle Types on Highways," *Explor. J. Sist. Inf. dan Telemat.*, vol. 14, no. 2, p. 110, 2023, doi: 10.36448/jsit.v14i2.3259.

[13] B. Erwanto, A. I. Pradana, and D. Hartanti, "Develoment of a Tomato Plant Disease Detection System Through Leaf Images With the You Only Look Once (YOLO) Method Based on Android," *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 1453–1463, 2024, doi: 10.33379/gtech.v8i3.4327.

[14] N. Khairunisa, . C., and A. Jamaludin, "Comparative Analysis of CNN and YOLO Algorithms in Identifying Road Damage," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024, doi: 10.23960/jitet.v12i3.4434.

[15] I. C. Sari, "Integration of EfficientNet-B0 Deep Learning Model for Tomato Leaf Disease Detection in Flutter Based Mobile Application," *Djtechno J. Teknol. Inf.*, vol. 5, no. 2, pp. 332–346, 2024, doi: 10.46576/djtechno.v5i2.4651.

[16] M. A. K. Majid and D. Ariatmanto, "Comparative Performance Analysis of YOLO11 and YOLOv8 Models in Tomato Leaf Disease Identification," *J. Bangkit Indones.*, vol. 14, no. 2, pp. 63–72, 2025, doi: 10.52771/bangkitindonesia.v14i2.459.

[17] B. Raharjo, *Deep Learning dengan Python*. 2022.

[18] S. Cong and Y. Zhou, *A review of convolutional neural network architectures and their optimizations*, vol. 56, no. 3. Springer Netherlands, 2023. doi: 10.1007/s10462-022-10213-5.

[19] R. Patel and S. Patel, "A comprehensive study of applying convolutional neural network for computer vision," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 6 Special Issue, pp. 2161–2174, 2020.

[20] W. M. Ardana, "Optimization of Convolutional Neural Network Algorithm with Efficientnet-B0 and Resnet-50 Architecture for Waste Type Classification Optimasi Algoritma Convolutional Neural Network dengan Arsitektur Efficientnet-B0 dan Resnet-50 untuk Klasifikasi Jenis Sa," vol. 5, no. October, pp. 1274–1286, 2025.

[21] "Home - Ultralytics YOLO Docs." Accessed: Apr. 03, 2026. [Online]. Available: https://docs.ultralytics.com/

[22] D. Ardianto, J. Jasril, S. Sanjaya, L. Handayani, and F. Syafria, "Beef and Pork Image Classification Using Convolutional Neural Network (CNN) With EfficientNet-B2 Architecture and Data Augmentation," *J. Inform. Univ. Pamulang*, vol. 8, no. 2, pp. 153–164, 2023, doi: 10.32493/informatika.v8i2.30587.