Journal of Multidisciplinary Research and Innovation
Vol 1 No 1 Januari 2025
ISSN: XXXX-XXXX (Electronic)
Open Access: https://journals.ai-mrc.com/jmri

Enhancing Agricultural Efficiency through YOLOv8 for Papaya Ripeness Detection

Salma Zahra Dzakia [1], Guntur Eka Saputra[2], Irwan Bastian[3]
[1]Gunadarma University, Indonesia
[2]Gunadarma University, Indonesia
[3]Gunadarma University, Indonesia
[1]salmazahra80@gmail.com, [2]guntur@staff.gunadarma.ac.id, [3]bastian@staff.gunadarma.ac.id*

Article Info

Article history:
Received 12-12-2024
Revised 03-01-2025
Accepted 10-01-2025

Keyword:
Deep Learning; Papaya; YOLOv8; Object Detection; Computer Vision.

ABSTRACT

The increasing production of papaya fruit in Indonesia, which will reach 1.24 million tons by 2023, poses a challenge in the manual identification of fruit ripeness. The ripeness of papaya fruit greatly affects its quality, flavor, and selling value, and is important in determining the use of the fruit, whether as fresh fruit, vegetable, meat tenderizer, or ingredient for salad. Manual identification, which relies on visual, textural, and aroma assessments, is prone to eyestrain, subjective perception, and varying degrees of accuracy, leading to inconsistent results. Therefore, a more effective and efficient solution is needed to overcome this problem. This research aims to develop a papaya fruit ripeness detection system using YOLOv8. The dataset used consists of images of papaya fruit on the tree and those that have been cut, which have gone through the process of bounding box annotation, preprocessing, and augmentation using Roboflow. The hyperparameter used is epoch 50 and learning rate 0.01. The results of the training model show an accuracy rate of mAP50 of 0.873 for all classes, with values in the unripe of 0.883, semi-ripe 0.852, and ripe 0.884. With this model, it is hoped that the public can obtain more accurate and precise information about the level of ripeness of papaya fruit, reduce dependence on manual methods, and make an important contribution to the papaya fruit farming industry by increasing the accuracy of ripeness assessment.

INTRODUCTION

Papaya (*Carica papaya L.*) is one of the most popular tropical fruits in Indonesia, known for its high nutritional and vitamin content, including vitamins A, B1, C, E, protein, and calcium. This fruit has the advantage of growing throughout the year, independent of specific seasons (Utami et al., 2022). According to data from the Food and Agriculture Organization (FAO), Indonesia is one of the largest papaya producers globally. Production reached 1.02 million tons in 2020, increased to 1.24 million tons in 2023, marking a 25.5% increase over the past five years (Sadya, 2022; Fadhlurrahman, 2024). However, with such a large production volume, assessing papaya ripeness still heavily relies on manual methods, such as observing skin color, pressing the fruit to check texture, and evaluating its aroma. These methods are prone to subjectivity, and can result in inconsistencies. Errors in identifying ripeness can have economic consequences, including reduced quality, taste, and market value.

The automation process using computer vision technology has been proposed to overcome these challenges. Computer vision is a branch of artificial intelligence that enables machines to process and interpret visual data in ways similar to human perception (Matsuzaka, Y., & Yashiro, R, 2023). Object detection algorithms such as you only look once (YOLO) is a popular object detection algorithm that uses a single neural network to predict multiple bounding boxes and their associated class probabilities in one pass (Terven et al., 2023). YOLOv8, was chosen for

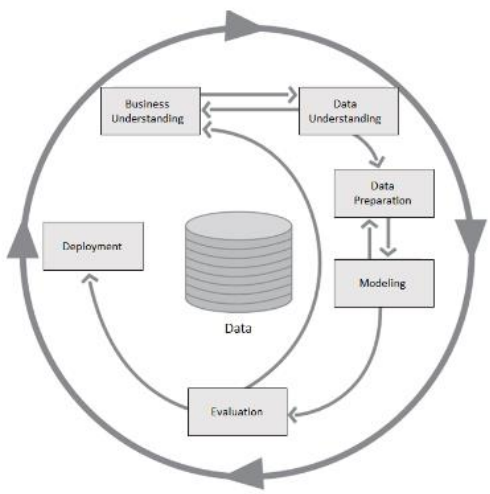
`
this study due to its high processing speed, accuracy, and low error rates. YOLOv8 excels in real-time applications and can detect multiple objects in a single frame with updated datasets, ensuring reliable performance (Syahrudin et al., 2024).

Previous research on papaya ripeness detection has used methods such as Lab Method and DHT11 and RGB analysis but achieved limited accuracy due to small datasets and inadequate training (Brucal et al., 2021; Widyasari et al., 2021). Papaya ripeness research was conducted using a dataset of 300 data using the MobileNet-SSD V2 algorithm and the classification loss results obtained were 0.0490 at the 20000th iteration. This papaya fruit ripeness detection system can be built, but requires a larger number of assets so that model accuracy is better and model overfitting does not occur (Muttaqin et al., 2023). Further research used eight transfer learning architectures with 216 simulations with specified parameter limits, and the accuracy results reached 97%. However, the model still experienced overfitting in recognition in the semi-ripe and unripe papaya classes. This requires additional datasets so that the algorithm can better recognize the classes being tested (Nurmalasari et al., 2023). The use of the CNN algorithm with 2 layers for the classification of ripeness levels, the number of datasets used is still quite small with 3 classes (unripe, semi-ripe, ripe) totaling 630. The accuracy results achieved were 96.97%, but still require more variations and datasets (Hawibowo et al, 2024). And further research using the CNN algorithm using 6 layers for the classification of papaya fruit ripeness. The accuracy results achieved were 96.63%, but still using a dataset with a total of 300 datasets, and the model still experienced overfitting, because the number of datasets in the training process was still small (Sutrisna et al., 2024). In contrast, YOLOv8 demonstrated an accuracy of over 90%, albeit with some limitations under environmental conditions (Xiao et al., 2023).

This research focuses on the development of papaya ripeness detection using annotated papaya image datasets using a platform such as Roboflow. This research aims to improve manual methods, providing an efficient and accurate system for assessing papaya ripeness based on skin color using the YOLOv8 algorithm. The proposed solution is expected to contribute to the advancement of smart agriculture, improve product quality, reduce dependence on manual processes, and support the growing Indonesian horticulture sector.

RESEARCH METHODS

This research follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology for model development. CRISP-DM is a widely adopted framework for data mining projects that provides a structured approach to guide professionals from understanding the business problem to implementing the final model. The methodology outlines the project lifecycle, detailing the various phases, tasks, and their interconnections. By using CRISP-DM, organizations can efficiently manage and execute data mining projects, gaining valuable insights and making informed decisions. This technology-independent model can be applied across different industries.



`

Figure 1. CRISP-DM Reference Model Phases
(Source: Shimaoka et al., 2024)

The data mining project lifecycle consists of six phases. Figure 1 shows the phases of the data mining process. The outer circle in Figure 1 symbolizes the cyclical nature of data mining itself. Data mining does not end once a solution is implemented. The next data mining process benefits from the experience of the previous one. The following are the stages of the CRISP-DM method carried out in this scientific writing.

Business Understanding

The research identifies the manual assessment of papaya ripeness relying on subjective evaluation of skin color, texture, and fragrance as a major challenge due to its time-consuming, prone to human error, inconsistent nature, and particularly in mass production. The lack of advanced tools exacerbates the issue, making reliable ripeness evaluation difficult for stakeholders. This research proposes leveraging artificial intelligence by developing a model using the YOLOv8 algorithm. This approach seeks to enhance efficiency and accuracy in ripeness detection, with an initial plat to collect papaya image datasets, train the model, and evaluate its performance.

Data Understanding

The dataset comprises primary data captured using a smartphone camera at papaya plantation and secondary data sourced from Roboflow. A thorough understanding of this data is critical to ensure the yOLOv8 model can accurately detect papaya ripeness under diverse conditions. This step is crucial in securing high-quality test and training data for developing the model.

Data Preparation

Data preparation is carried out by collecting images of papaya fruit of various maturity levels. Categories or classes of papaya fruit maturity levels in this study are divided into unripe, semi-ripe and ripe maturity levels. Primary data obtained as many as 72 images and secondary data as many as 3043. With a total dataset of 3115 images, the data is divided into 70% for training, 20% for validation, and 10% for testing. The data then entered the annotation and labeling stage with the aim of indicating papaya fruit in the image by creating a bounding box and labeling it according to the object class.

The dataset is carried out at a preprocessing stage where at this stage the image size is changed to 800x800 pixels. From the preprocessing results, an augmentation process is carried out where at this stage. Augmentation features are performed to increase the variation of the model training data expected to help the model understand features that have changes in object orientation in real data, data whose orientation is not fixed, recognize objects from a portion of the image or when the object feature is cropped, reduce the model's dependence on specific orientations in the original image, help the model be more tolerant of object position translations, recognize objects in different lighting, and improve the ability to recognize objects in visual disturbances due to movement or low quality, then the image is rotated horizontally vertically, rotated 90°, cropped, rotated, shifted, brightness, and blurred augmentation feature is used. From the augmentation results, a dataset of 5336 images is generated and the data division can be seen in Table 1 as follows.

Table 1. Datasets Distribution

Total Dataset	Split Datasets		
	Training Data	Validation Data	Testing Data
5336	4394	630	312

Modeling

`

During this phase, various modeling techniques are selected and applied to the prepared data, focusing on optimizing model performance through parameter tuning. For this research, the yOLOv8 model was trained using carefully prepared dataset through a series critical steps conducted on Google Colaboratory. This platform provided the computational resources needed to handle the intensive processes involved in training and validating the model. The primary goal was to train the YOLOv8 model to accurately detect papaya fruits into three classes, unripe-semi-ripe, and ripe based on annotated data.

The dataset, comprising both primary data (collected directly from papaya plantations) and secondary data, was divided into three subsets: training, validation, and testing. The training process involved configuring key performances such as the number of epochs, batch size, and learning rate. These configurations were fine-tuned to ensure the model achieved high accuracy in detecting papaya ripeness across diverse conditions.

Evaluation

After modelling phase is complete, the evaluation process assesses the model's alignment with predefines business objectives. Key performance metrics such as precision, recall, mean Average Precision (mAP), and mAP50-95 are used to measure the success of the model. These metrics provide insight into its accuracy and overall performance, guiding necessary adjustment like hyperparameter tuning. This iterative process ensures the model achieves optimal functionality and readiness for integration into web applications.

Deployment

The final phase of model deployment, which involves operational use and performance monitoring, is excluded from this research. The study focuses solely on development and evaluation, emphasizing the model's accuracy in detecting papaya ripeness under controlled conditions.

RESULTS AND DISCUSSION

The CRISP-DM method provides a systematic framework for managing data mining projects, ensuring alignment with technical and business objectives. This approach minimizes risk and maximizes the scientific and pratcial value of data mining techniques The CRISP-DM stages can be seen in Figure 2.

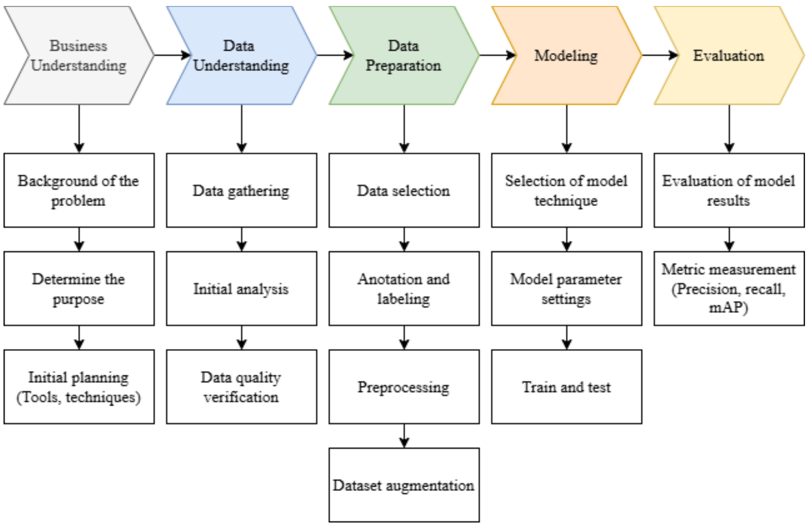


Figure 2. CRISP-DM Stages

Business Understanding

At this stage, a literature study was conducted and it was found that the process of identifying the ripeness of papaya fruit, especially on a mass production scale, is still a challenge. Manual identification of papaya fruit ripeness is done based on visual skin color, texture examination by pressing the skin, and aroma examination. This method has several disadvantages, such as human eye fatigue, the need for a lot of labor, and different perceptions regarding the level of ripeness. Errors in identification occur due to dependence on individual understanding of papaya fruit characteristics and varying levels of accuracy.

These problems are made further by the lack of a reliable app for determining papaya ripeness, which makes it difficult for people to guarantee the highest possible fruit quality. The suggested remedy for this is to create a detection model using the YOLOv8 algorithm. The ripeness evaluation procedure will be automated by the model, which will yield accurate and consistent findings. To guarantee scalability and cost-effectiveness, the research makes use of cloud-based resources by executing the algorithm on Google Colaboratory.

The expected outcome of this approach includes achieving a detection accuracy of at least 80% and reducing the time and effort required for manual identification. Additionally, the model's implementation aims to streamline operations, particularly in mass production environments, and offer a practical tool for improving decision-making in the agricultural sector.

Data Understanding

At this stage, the data collected consisted of primary data and secondary data. Primary data was obtained directly from papaya fruit plantations in the neighborhood of Taman Cimanggu, Bogor. A total of 72 images were taken using a smartphone camera with 48 MP (megapixel) resolution. Images were taken from angles and distances to get a perspective of each papaya fruit. Then, the primary data is uploaded into the workspace on Roboflow where it is combined with secondary data.

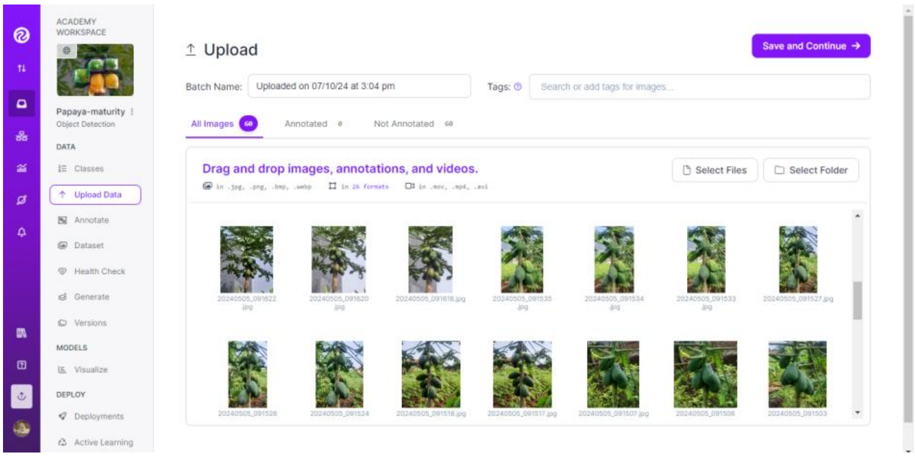


Figure 3. Primary Data Upload Process

The secondary data itself is obtained from the Roboflow website by cloning the papaya fruit image into the workspace. The dataset can be downloaded at the link https://universe.roboflow.com/academy-workspace/papaya-maturity-lvicf/dataset/15. The secondary data obtained amounted to 3043 images.

`

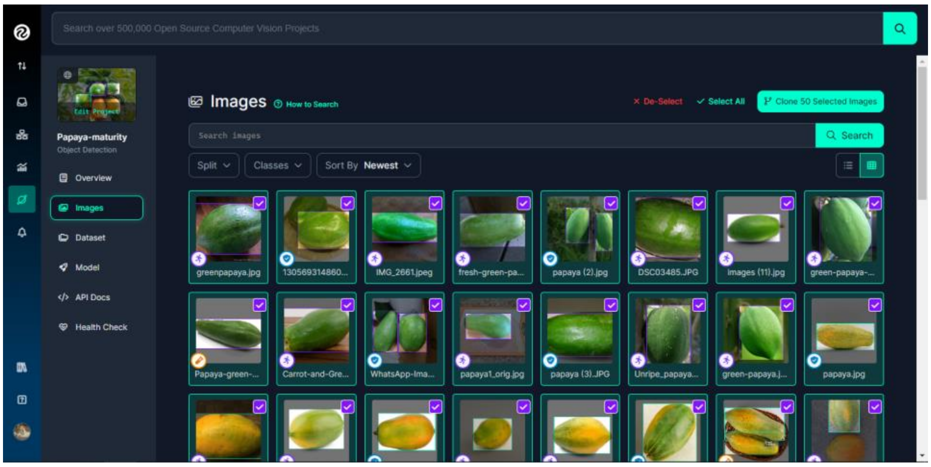


Figure 4. Secondary Data Clone Process

In total, 3115 images were used to obtain the test and training data required for model development. The data includes images of papaya fruits with various levels of ripeness, namely unripe, semi-ripe, and ripe.

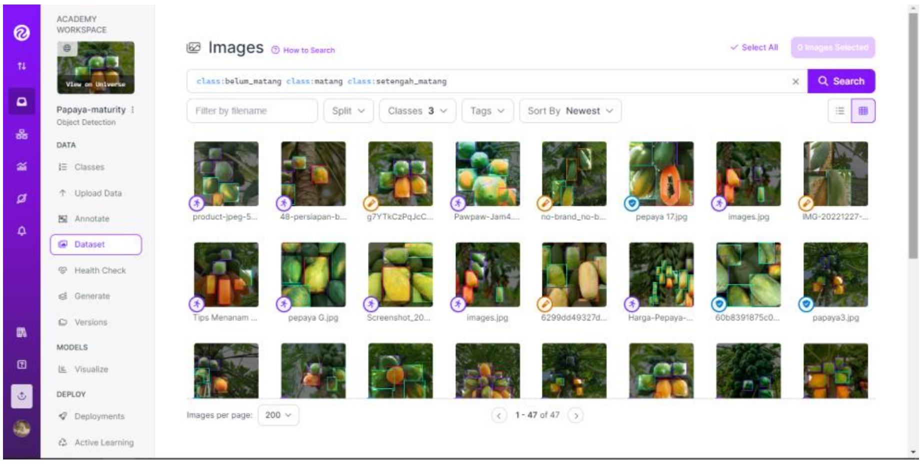


Figure 5. Dataset In Workspace

Data Preparation

This data preparation stage includes annotation, split data, preprocessing, and augmentation. The 3115 datasets were annotated by applying labels to the images that indicate the position and type of papaya fruit in each image. This process involves marking the object with a bounding box around the papaya fruit and then labeling it according to the ripeness level of the fruit.

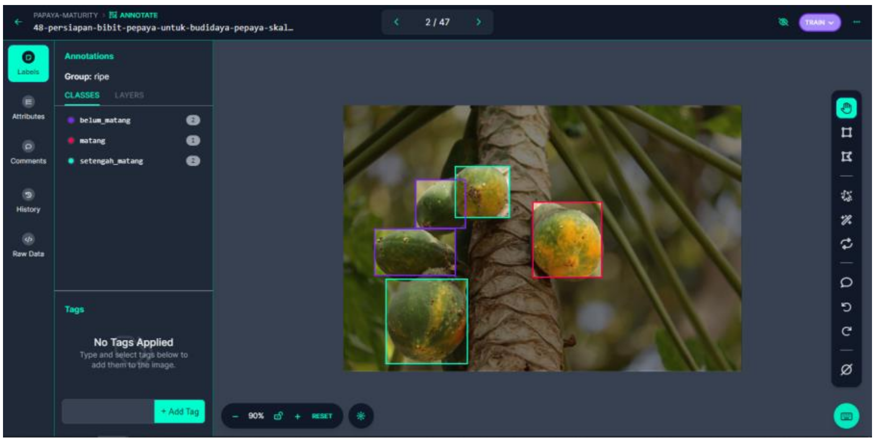


Figure 6. Data Labeling Using Roboflow

The bounding box annotation results in this study include a total of 6.130 annotations, which are divided into three categories of papaya fruit ripeness levels. The following is the number of annotations for each class in the dataset.

Table 2. Number of Annotations For Each Class

No	Class	Number of Annotations
1	Unripe	2.144
2	Semi-ripe	1.876
3	Ripe	2.110

Then the data that has been annotated is divided into three different subsets, namely train, valid, and test. Training data is data that is trained by the model in determining the accuracy of the model to detect objects based on the class that has been determined in the labeling process. Validation data is data used to validate the model during the training process. Testing data is data used in testing the model when the training process has been completed. In this research, the dataset is divided with a composition of 70% for training, 20% for validation, and 10% for testing. This data split process can be seen in the Health Check navigation. Then the data division is obtained as follows.

Table 3. Dataset Subset Before Augmentation

No	Subset	Total
1	Train	2180
2	Valid	625
3	Test	315

Datasets that have been divided into train, valid, and test, then pass the preprocessing stage. In the preprocessing stage, resize and auto-orient are performed. The dataset goes through the data orientation stage to ensure all images have the correct and consistent orientation. Resize the image to 800x800 pixels so that the image size is consistent with the dimensions required by the deep learning model.

From the preprocessing results, the dataset goes through an augmentation stage. This augmentation stage increases the number and diversity of datasets used in training deep learning models. Data augmentation helps the model to become more robust and can generalize better by introducing additional variations in the training data. Augmentations applied to the dataset are vertical horizontal rotate, 90° rotate, crop, rotation, shear, brightness, and blur. The augmentation results in a dataset of 5336 images and the division of data can be seen in the following figure.

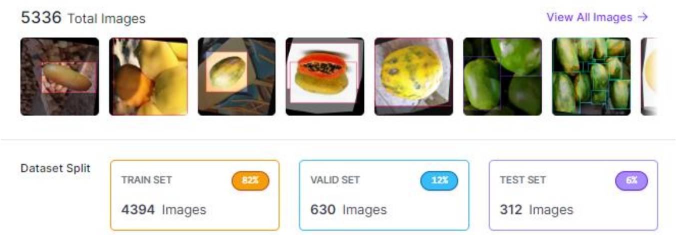


Figure 7. Dataset Split After Augmentation

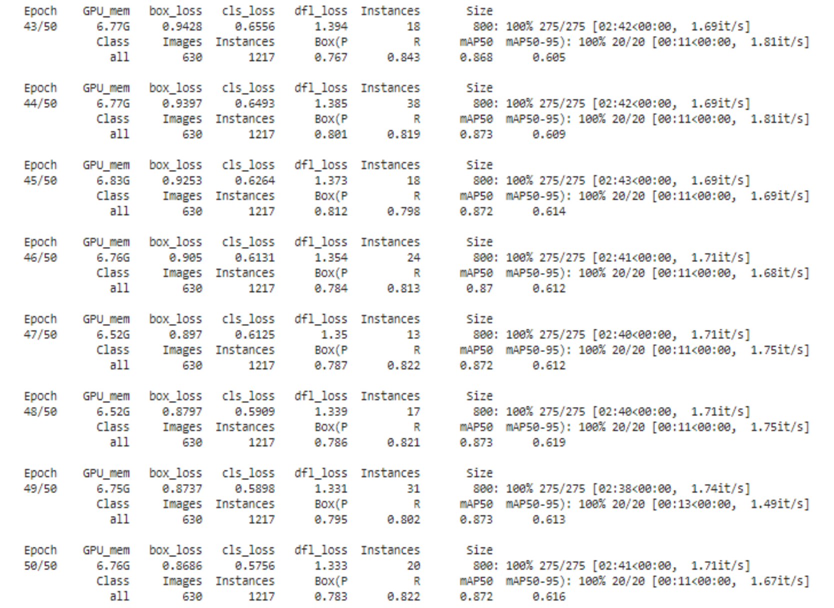
Modeling

At this stage, the YOLOv8 model is built using the dataset that has been prepared in Roboflow. The model was trained to recognize papaya fruits in three classes, namely unripe, semi-ripe, and ripe. The model training was done with the help of Google Colaboratory using the T4 GPU with 15,102 MB of memory, which speeds up the training and validation process of the model. The system also utilized 2 CPUs, 12.7 GB of RAM, and a disk capacity of 78.2 GB, with approximately 30.2 GB available for use during the training process. The software environment was configured using Python 3.10.12 as the programming language and PyTorch 2.3.0+cu121 as the deep learning

`

framework, enabling CUDA 12.1 support for efficient GPU utilization. The Ultralytics 8.0.196 package was installed to provide the necessary tools and functionalities for YOLOv8, ensuring compatibility with the papaya ripeness detection.

The dataset version that was created in Roboflow was inserted into the Colab notebook by exporting the dataset version and getting the generated code snippets. The code is responsible for installing the Roboflow package so that it can be integrated with the Roboflow platform easily, as well as accessing projects that have been defined in the workspace. The next step is to train the YOLOv8 model. The training of this model consists of several stages, starting from determining the object detection task, indicating the training model. The required data configuration is set through the path to the data configuration file. The model training process is run with 50 epochs with an image size of 800x800 pixels.



```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
43/50      6.77G     0.9428     0.6556      1.394         18        800: 100% 275/275 [02:42<00:00,  1.69it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:11<00:00,  1.81it/s]
             all        630       1217      0.767      0.843      0.868      0.605

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
44/50      6.77G     0.9397     0.6493      1.385         38        800: 100% 275/275 [02:42<00:00,  1.69it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:11<00:00,  1.81it/s]
             all        630       1217      0.801      0.819      0.873      0.609

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
45/50      6.83G     0.9253     0.6264      1.373         18        800: 100% 275/275 [02:43<00:00,  1.69it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:11<00:00,  1.69it/s]
             all        630       1217      0.812      0.798      0.872      0.614

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
46/50      6.76G      0.905     0.6131      1.354         24        800: 100% 275/275 [02:41<00:00,  1.71it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:11<00:00,  1.68it/s]
             all        630       1217      0.784      0.813       0.87      0.612

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
47/50      6.52G      0.897     0.6125       1.35         13        800: 100% 275/275 [02:40<00:00,  1.71it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:11<00:00,  1.75it/s]
             all        630       1217      0.787      0.822      0.872      0.612

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
48/50      6.52G     0.8797     0.5909      1.339         17        800: 100% 275/275 [02:40<00:00,  1.71it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:11<00:00,  1.75it/s]
             all        630       1217      0.786      0.821      0.873      0.619

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
49/50      6.75G     0.8737     0.5898      1.331         31        800: 100% 275/275 [02:38<00:00,  1.74it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:13<00:00,  1.49it/s]
             all        630       1217      0.795      0.802      0.873      0.613

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
50/50      6.76G     0.8686     0.5756      1.333         20        800: 100% 275/275 [02:41<00:00,  1.71it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:11<00:00,  1.67it/s]
             all        630       1217      0.783      0.822      0.872      0.616
```

Figure 8. YOLOv8 Model Training Process

Evaluation

At this stage, a thorough evaluation of the model is conducted. Ensuring that the trained model functions well is also able to maintain performance on data that has not been seen before. Evaluation results such as precision, recall, mean Average Precision (mAP), and mAP50-95 are used to measure the success of the model and ensure the resulting model can function optimally in the application. The results of training the model for 2 hours with 50 epochs and learning rate lr0=0.01 and lrf=0.01, the model achieved precision 0.785, recall 0.821, mAP50 0.873, and mAP50-95 0.618.

Table 4. Results of Model Training

No	Class	Precision	Recall	mAP50	mAP50-95
1	All	0.785	0.821	0.873	0.618
2	Unripe	0.765	0.823	0.883	0.643
3	Semi-ripe	0.778	0.78	0.852	0.614
4	Ripe	0.814	0.86	0.884	0.598

For a more detailed description of the results of the training model, refer to Figure 13. This figure provides the training performance metrics, including the accuracy, precision, and recall during the training process. The optimizer and the final weights from the training process were successfully saved, with the best weights located in the specified directory. The validation phase provides an overview of the model's performance for three classes: *belum matang* (unripe), *matang* (ripe), and *setengah matang* (semi-ripe). Key metrics, such as precision (P), recall (R), mean average precision (mAP) at thresholds of 50% (mAP@50) and 50-95% (mAP@50-95), highlight the effectiveness of the model.

```
50 epochs completed in 2.535 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.6MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.6MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 20/20 [00:18<00:00,  1.10it/s]
                   all        630       1217      0.785      0.821      0.873      0.618
           belum_matang        630        446      0.763      0.823      0.883      0.643
                matang        630        421      0.814       0.86      0.884      0.598
        setengah_matang        630        350      0.778       0.78      0.852      0.614
Speed: 0.5ms preprocess, 7.4ms inference, 0.0ms loss, 4.2ms postprocess per image
Results saved to runs/detect/train
```

Figure 9. Model Training Results

The trained model produces a confusion matrix of size 4x4 as shown in Figure 10. Each cell in the matrix shows the number of predictions that correspond to the combination of the prediction and the actual label. This matrix helps in understanding the performance of the model by showing how many predictions are correct and incorrect for each class. The model detected "unripe" well, with 392 correct predictions. The model was fairly accurate in detecting "ripe", with 369 correct predictions. The model correctly predicted "semi-ripe" 274 times. The model often incorrectly predicted "background" as another class, especially "unripe" (163 times) and "ripe" (99 times). The model performed well in classifying "unripe", "ripe", and "semi-ripe", with a fairly high TP value. However, there was some confusion especially in distinguishing between "background" and other classes, as well as between "unripe" and "semi-ripe".

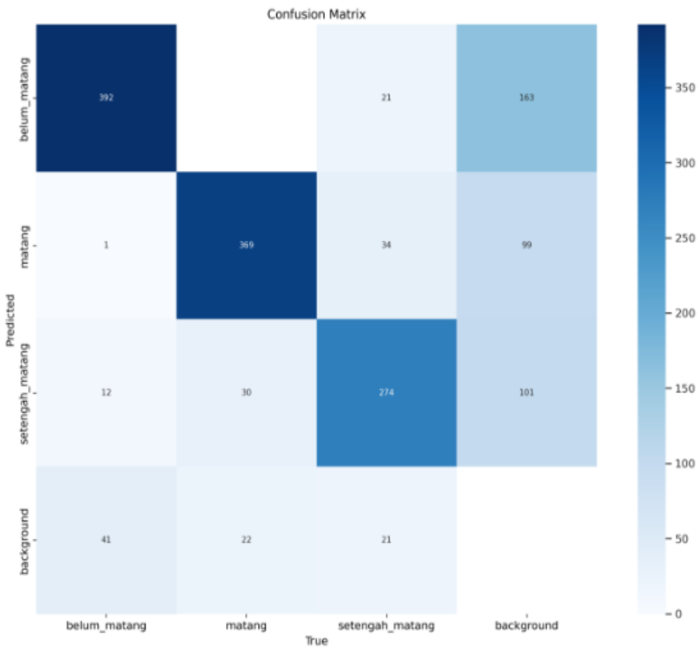


Figure 10. Confusion Matrix

Based on figure 10 Significant errors can occur between distinguishing the background with the unripe, semi-ripe and ripe classes can occur several factors. This error can occur because of the visual similarity between the background and the papaya object which is possible because it has a similar color, texture or pattern, and the training data is not enough to cover the variation of the background, so the model can misidentify the background area as an object. Data imbalance should also be considered because this imbalance makes the model more likely to be "biased" to the class with more data for the three existing classes. Then, inadequate augmentation data, which does not cover variations in the background, the model becomes robust in dealing with more complex and diverse real data.

Furthermore, the training and validation metrics of the YOLOv8 model are displayed in graphical form. Overall, these graphs show a positive trend in the model training and validation process. The decrease in loss and increase in evaluation metrics (precision, recall, mAP) indicate that the model becomes more effective in detecting and classifying objects as the training iterations increase. It is important to ensure that the model learning data from the training data is also able to maintain its performance on data not seen during training (validation data).

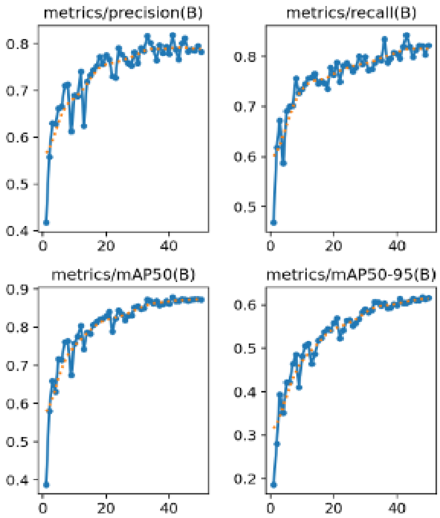


Figure 11. Graph of Model Training and Validation Metrics

1. metrics/precision(B): Precision indicates how good the model is at avoiding false positives. The graph shows that the precision of the model increases from about 0.45 to reach about 0.78 at the end of training. The trend of the graph shows a steady increase with some fluctuations, which indicates that the model is getting better at identifying the correct objects without too many false positives.

2. metrics/recall(B): Recall indicates how good the model is at finding all the objects that actually exist (true positives). The graph shows that the model's recall increases from about 0.45 to reach about 0.82 by the end of training. The trend of the graph also shows a steady increase with small fluctuations, indicating that the model is getting better at detecting all the objects in the dataset.

3. metrics/mAP50(B): mAP50 measures the average precision at various Intersection over Union (IoU) thresholds starting from 0.50. This graph shows the increase in mAP50 from about 0.2 until it reaches about 0.87. The steady increasing trend with little fluctuation shows that the model is getting better at detecting objects with a higher level of precision across various IoU conditions.

4. metrics/mAP50-95(B): mAP50-95 measures the average precision at various IoU thresholds from 0.50 to 0.95. This graph shows an increase from about 0.2 until it reaches about 0.62. The consistent improvement shows that the model is getting better at detecting objects with a high degree of precision at various IoU thresholds, although the improvement is not as great as that of mAP50.

`

Overall, the performance of the model improved significantly over the 50 epoch training. The precision, recall, mAP50, and mAP50-95 metrics show a steady increase, indicating that the model becomes more accurate and reliable in detecting objects.

Furthermore, displaying an image showing the model's prediction on the first batch of the validation dataset after training provides a direct picture of the model's performance on the validation data. By displaying the prediction image, users can quickly evaluate the model's performance visually, seeing how the model recognizes and tags objects in the image.

Figure 12. Prediction Image

The YOLOv8 model validation process is performed using validation data to evaluate the performance of the object detection model. This step involves using the best model generated from training, called best.pt, and is evaluated based on metrics such as precision, recall, and mean average precision (mAP). The dataset configuration file is instrumental in setting the parameters and data structure required for model training and evaluation. The results of this evaluation guide the adjustment of the model to improve the accuracy in detection of the desired object.

```
/content
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/Papaya-maturity-15/valid/labels.cache... 630 images, 0 backgrounds, 0 corrupt: 100% 630/630 [00:00<?, ?it/s]
WARNING ⚠ Box and segment counts should be equal, but got len(segments) = 1, len(boxes) = 1217. To resolve this only boxes will be used and all segments will be removed.
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 40/40 [00:16<00:00,  2.37it/s]
                   all        630       1217      0.786      0.821      0.873      0.618
          belum_matang        630        446      0.764      0.823      0.884      0.644
                matang        630        421      0.814       0.86      0.884      0.597
       setengah_matang        630        350      0.781       0.78      0.852      0.614
Speed: 1.4ms preprocess, 14.3ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to runs/detect/val
💡 Learn more at https://docs.ultralytics.com/modes/val
```

Figure 13. Model Validate Results

The next stage of inference involves applying the trained model, referred to as `best.pt`, to perform predictions on new data that has not been seen before.

```
!yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25
```

Figure 14. Inference Model Code

```
/content
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs

WARNING ⚠ NMS time limit 0.550s exceeded
image 1/312 /content/datasets/Papaya-maturity-15/test/images/0499_jpg.rf.bfc9213d3c2f237f5e4dfa9637400f5a.jpg: 800x800 3 belum_matangs, 22.5ms
image 2/312 /content/datasets/Papaya-maturity-15/test/images/0501_jpg.rf.763b7e10a214a62a27490e2ccc161359.jpg: 800x800 1 belum_matang, 22.5ms
image 3/312 /content/datasets/Papaya-maturity-15/test/images/0522_jpg.rf.407f21b440654ae4d5df81ae957006f2.jpg: 800x800 3 belum_matangs, 22.4ms
image 4/312 /content/datasets/Papaya-maturity-15/test/images/0523_jpg.rf.b87e3e752ac7a4dde30b99a3ae1d8998.jpg: 800x800 1 belum_matang, 22.4ms
image 5/312 /content/datasets/Papaya-maturity-15/test/images/0558_jpg.rf.015252e2dad286548d011564e36d8b31.jpg: 800x800 2 belum_matangs, 22.4ms
image 6/312 /content/datasets/Papaya-maturity-15/test/images/0563_jpg.rf.5cba22d69a8c38a0e8b41772026e03d0.jpg: 800x800 3 belum_matangs, 22.5ms
image 7/312 /content/datasets/Papaya-maturity-15/test/images/0578_jpg.rf.09353065d8aa734f190f91d9b6c09525.jpg: 800x800 3 belum_matangs, 20.0ms
image 8/312 /content/datasets/Papaya-maturity-15/test/images/0588_jpg.rf.82c3369f052d86edeeb5c649544eb324.jpg: 800x800 4 belum_matangs, 19.9ms
image 9/312 /content/datasets/Papaya-maturity-15/test/images/0600_jpg.rf.e976191ceaa94eafb4be535125355f55.jpg: 800x800 3 belum_matangs, 19.9ms
image 10/312 /content/datasets/Papaya-maturity-15/test/images/0607_jpg.rf.733bd75d1c0ab9a2a0d19e8d57aee98b.jpg: 800x800 1 belum_matang, 20.0ms
image 11/312 /content/datasets/Papaya-maturity-15/test/images/0631_jpg.rf.2915f206e81b87a0284a3ddd814f8167.jpg: 800x800 1 belum_matang, 19.4ms
```

Figure 15. Inference Model Results

After completing the inference process, the trained YOLOv8 model was tested on five papaya images to evaluate its ability to accurately detect and classify the ripeness levels of the fruit. This test provided an opportunity to observe the model's performance in recognizing the ripeness categories based on the features displayed in the images. The results of this evaluation served as a benchmark to assess the effectiveness of the model. Additionally, any inaccuracies or misclassifications observed during the test were analyzed to identify potential areas for improvement, such as enhancing the dataset, fine-tuning the model, or adjusting its hyperparameters to achieve better accuracy and reliability in future applications.

Figure 16. Output of the Testing Process On Papaya Images

After the testing using 5 images, it can be seen that there are 2 images tested that do not detect the object completely. From the results, it is known that the unripe class shows poor performance, as seen in the model training results (Table 4), the unripe class has a lower precision level than the semi-ripe and ripe classes.

CONCLUSION

This study developed a papaya ripeness detection model using YOLOv8. The model showed a high mean average precision (mAP50) of 0.873 across all classes and showed excellent accuracy in the semi-ripe (0.94) and ripe (0.89) classes in detecting papaya fruit. The model showed less than optimal results in detecting the unripe class, with an accuracy of only 0.852. This highlights the need for improvement for further research. The limited dataset and small sample size of the test images further limit the model's ability to generalize to real-world conditions. For further research, additional datasets are needed to cover unripe, semi-ripe, and ripe variations with a balanced amount of data, in addition to improving model performance by adjusting appropriate hyperparameters. This research can also be used to apply the model to data integration with mobile or web-based, as well as direct application with Internet of Things (IoT) integration for automatic and real-time monitoring of papaya fruit ripeness which can be placed in papaya gardens where sensors will send real-time images or data to the application for further analysis.

CONFLICT OF INTEREST

The authors declares that there is no conflict of interest regarding the publication of this research. There are no financial, personal, or professional relationships that could have influenced the research process, findings, or outcomes.

REFERENCES

Brucal, G.E., Colobong, J. B., De Guzman, D. B., De Leon, J. N. M., Samaniego, L. A., and Yong, E. D. (2021). Papaya Maturity Classification in MATLAB Platform using Lab Method and DHT11 Sensor. *2021 Fifth World Conference on Smart Trends in Systems Security and*

`

Sustainability (WorldS4), London, United Kingdom, 2021, pp. 171-175, doi: 10.1109/WorldS451998.2021.9514053

Fadhlurrahman, I. (2024). Produksi pepaya nasional naik 25.5% dalam 5 tahun terakhir. *Katadata Databoks*. Retrieved July 29, 2024. https://databoks.katadata.co.id/datapublish/2024/07/25/volume-produksi-pepaya-menurut-provinsi

Hawibowo, M. S., Muhimmmah, I. (2024). Aplikasi Pendeteksi Tingkat Kematangan Pepaya Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android. Jurnal Edukasi dan Peneliitian Informatika (JEPIN), 10 (1), 162-170, doi: https://doi.org/10.26418/jp.v10i1.77819

Matsuzaka, Y., & Yashiro, R. (2023). AI-Based Computer Vision Techniques and Expert Systems. *AI*, *4*(1), 289-302. https://doi.org/10.3390/ai4010013

Muttaqin, M.I. Stefani, A., Nurpulaela, L. (2023).Implementasi Deep Learning Untuk Menentukan Tingkat Kematangan Buah Pepaya California. JATI, 7 (3), 1878-1884, doi: https://doi.org/10.36040/jati.v7i3.6948

Nurmalasari, Setiawan, Y. A., Astuti W., Saelan, M. R. R., Masturoh, S., Haryanti, T. (2023). Classification For Papaya Fruit Maturity with Convolutional Neural Network. Jurnal Riset Informatika, 5 (3), 331-338, doi: https://doi.org/10.34288/jriv5i3.225

Sadya, S. (2022). Produksi pepaya di Indonesia capai 1,17 juta ton pada 2021. *DataIndonesia.id*. Retrieved July 1, 2024. https://dataindonesia.id/agribisnis-kehutanan/detail/produksi-pepaya-di-indonesia-capai-117-juta-ton-pada-2021

Shimaoka, A. M., Ferreira, R. C., & Goldman, A. (2024). The evolution of CRISP-DM for Data Science: Methods, Processes and Frameworks. *SBC Reviews on Computer Science*, *4*(1), 28–43. https://doi.org/10.5753/reviews.2024.3757

Sutrisna, N.P., Sahirah, R. A. Laksono, K. S. S. Permadhi, R. A. S., Nurannisa, N. Larasati, S. S., Asmani, W. W. Yudistira, N. (2024). Deteksi Tingkat Kemetangan Buah Pepaya Menggunakan Model Convolutional Neural Network. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 11 (3), 569-578, doi: 10.25126/jtiik.938119

Syahrudin, E., Utami, E., & Hartanto, A. D. (2024). Enhanced YOLOv8 with OpenCV for Blind-Friendly Object Detection and Distance Estimation. *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *8*(2), 199–207. https://doi.org/10.29207/resti.v8i2.5529

Terven, J., Cordova-Espearza, D.M., Romero-Gonzalez, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knol. Extr, 5, 1680-1716, https://doi.org/10.3390/make5040083

Utami, H. S., Susanto, S., & Hapsari, D. P. (2022). Keragaman Kualitas Fisik dan Kimia Buah Pepaya Calina di Balumbangjaya. *Jurnal Penelitian Pertanian Tropis*, *13*(1), 109–119. https://doi.org/10.29244/jhi.13.2.109-119

Widyasari, K. B. D. R. N., Rosiani, U. D., & Pramudhita, A. N. (2021). Implementasi Sistem Pendeteksi Tingkat Kematangan Buah Pepaya Menggunakan Metode RGB. *SMATIKA Jurnal*, *11*(1), 32–36. http://dx.doi.org/10.32664/smatika.v11i01.536

Xiao, B., Nguyen, M., & Yan, W. Q. (2023). Fruit ripeness identification using YOLOv8 model. *Multimedia Tools and Applications*, 28040–28056. http://dx.doi.org/10.1007/s11042-023-16570-9