**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

# Utilizing Lightweight YOLOv8 Models for Accurate Determination of Ambarella Fruit Maturity Levels

**Nurchaya Simanjuntak[1], Raymond Erz Saragih[2,*], Yonky Pernando[2]**

[1] Magister of Informatics, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
[2] Faculty of Computer, Department of Informatics, Universitas Universal, Batam, Indonesia
Email: [1] nurcahaya@students.amikom.ac.id, [2,*]raymondes@uvers.ac.id, [3]yongkyfernando194@gmail.com
Correspondence Author Email: raymondes@uvers.ac.id
Submitted: 29/04/2024; Accepted: 31/05/2024; Published: 31/05/2024

**Abstract**−In the agricultural sector, accurately determining fruit ripeness remains a crucial yet challenging task. Among intriguing Indonesian fruits, the Ambarella presents a particular difficulty. In Ambarella fruit, the peel changes from green to golden yellow as it ripens, serving as a visual indicator for optimal harvest time, thus determining the maturity is crucial for harvesting the Ambarella fruit. Traditionally, ripeness assessment relies on manual methods, which suffer from drawbacks like high labor costs, significant time investment, and inconsistency in results. This work explores the potential of employing YOLOv8, a cutting-edge deep learning model, to automate Ambarella fruit ripeness classification. This work focuses on the YOLOv8n, YOLOv8s, and YOLOv8m, lightweight models within the YOLOv8 family. Our results are promising: all three models achieved 100% accuracy on the training set, with YOLOv8s demonstrating the lowest loss at 0.00286. The web application was utilised to deploy the trained models, allowing users to upload images of Ambarella fruit and run the model for inference.

**Keywords**: Ambarella Fruit; Deep Learning; Fruit Ripeness; YOLOv8

## 1. INTRODUCTION

In Indonesia, the agricultural sector is an important contributor to the country's overall economic development [1]. Rice is one of the many marketable crops that may be harvested from this area, along with a wide variety of vegetables and a diverse product of fruits [2] [3]. The Ambarella is regarded as one of the most fascinating fruits in Indonesia due to its one-of-a-kind flavor and the traditional applications it has. Each and every part of the plant, from the luscious fruit to the leaves, panicles, and even the bark, serves a purpose that is either delicious or medicinal. The plant is more than just a delicious fruit [4]. Ambarella fruit boasts high dietary fiber, making it a natural remedy for digestive issues, anemia, and even blood sugar control [5], [6].

In agricultural sector, one of the main tasks is to determine the ripeness of fruit [7]. By precisely assessing fruit ripeness, farmers, supermarkets, and retailers ensure only the highest-quality produce reaches consumers. For fruits like mangoes, tomatoes, papaya, grapes, and bananas, a blush of color or a shift in hue can reveal their peak ripeness, offering a helpful clue amidst other maturity indicators [8]. The peel of Ambarella fruit undergoes a color change, transitioning from green to golden yellow as the fruit matures. This transformation serves as a reliable visual cue to identify the appropriate stage for harvesting the fruit; therefore, determining the maturity of the Ambarella fruit at different stages is crucial [9]. Ambarella shares the challenge of ripeness evaluation with other fruits [8], as the current workflow reliance on manual assessments falls short [10]. Its drawbacks—high labor costs, time constraints, and inconsistencies, and can produce waste—call for an innovative solution. Introducing an accurate and automated ripeness model holds immense potential to revolutionize fruit production and quality control.

Deep Learning (DL) has surged in popularity in recent years, fuelled by its remarkable ability to automatically extract features from images. This cutting-edge machine learning (ML) technique has revolutionized diverse fields, from web search to image analysis, enabling object classification and detection with unprecedented accuracy [11]. The agricultural and food industries have significantly benefited from the rise of Convolutional Neural Networks (CNNs), a branch of Deep Learning renowned for its prowess in pattern recognition and efficient classification. Recognized as a leading classification technique, CNNs have been extensively deployed to solve diverse challenges within these sectors [12], [13], [14], [15]. Moreover, recent advances in object classification demonstrate their superiority, with CNNs consistently surpassing even the most cutting-edge traditional ML models [12].

Several works have proposed the use of deep learning, specifically CNNs in order to categorize the level of maturity of a variety of fruits. In their research, Khosravi et al. offered a CNN model to identify the ripening stages of olive fruit from the tree branch [12]. The work investigated the use of different optimizers and found that the use of Nadam yields the highest accuracy of 91.91%. The work of Pardede et al. proposed a model to classify the ripening stages of several fruits, which are apple, mango, orange, and tomato [16]. The proposed model utilizes a pre-trained VGG-16 model, and the authors trained it using the transfer learning approach. The proposed method achieved the highest accuracy of 90%, 100% precision, 100% recall, and 100% F1-Score. The work of Tapia et al. proposed a deep learning-based models for various fruits and vegetables maturity grading [17]. The work utilized MobileNetV2 and successfully achieved accuracy of 100%; thus, have proven that deep learning-based model can achieve satisfactory results. Another work by Ajil et al. has proposed the use of

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

YOLOv3, to predict the maturity stages of several fruits [18]. The resulting model achieved confidence level in detecting the fruit maturity up to 99%.

The work of Saragih et al. proposed the classification of Ambarella fruit ripeness using the EfficientNetV2 family models, specifically the EfficientNetV2B0, EfficientNetV2B1, and EfficientNetV2B2 [19]. The work applies standard image augmentation such as rotation, brightness, flipping, contrast, and zooming. To train each model, the work uses the transfer learning via fine-tuning approach. The highest accuracy on the training set achieved was 97.71%.

Recently, a new family of CNN model, known as YOLOv8, was introduced to the computer vision field. The YOLO architecture firstly introduced in 2015 by Joseph Redmon et al. [20]. The current YOLOv8 achieves state-of-the-art performance compared to the previous models as well as previous iteration of the YOLO family [21]. YOLOv8 can be used to tackle computer vision problems, such as object detection, image classification, object segmentation, and human pose detection. Fuelled by recent breakthroughs in Convolutional Neural Networks (CNNs), researchers are exploring exciting possibilities in fruit maturity assessment, including the Ambarella fruit. Therefore, this work investigates the performance of newer models, specifically the YOLOv8, to know the comparison between the current and the previous model, used for the same task.

## 2. RESEARCH METHODOLOGY

This work follows the steps shown in figure 1.


**Figure 1.** Process flow of the current work

As shown in figure 1, the process flow in this work begins by acquiring a dataset. This dataset is then pre-processed to ensure its quality before being used to train the chosen models. After training, the models are evaluated to assess their performance. Finally, the best performing model is deployed. The detailed explanations are as follows:

### 2.1 Acquiring Dataset and Dataset Preparation

This work uses a dataset of Ambarella fruit images captured at four different ripening stages used in previous study by Saragih et. al [19]. Figure 2 shows several samples of the Ambarella fruits in the dataset.

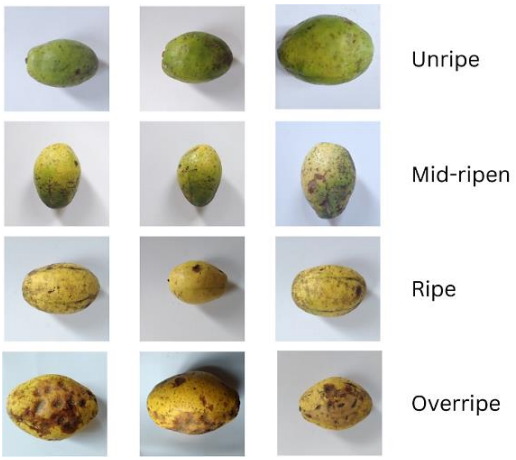


**Figure 2.** Samples of the Ambarella fruit dataset [19]

The original dataset, meticulously curated for this study, comprises 374 high-resolution images of Ambarella fruits. As shown in figure 2, these images capture the fruit's transition through four distinct ripening stages – unripe, mid-ripen, ripe, and overripe – under natural lighting conditions. While the dataset contains a sizeable number of images, there is still the need to apply augmentation, as without it, overfitting is likely to occur. The augmentation in this work uses several image processing techniques, such as vertical and horizontal flipping, rotation, brightness adjustment, random gaussian blurring, salt and pepper noise, and cutout. The result of applying the image augmentation techniques is shown in figure 3.

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

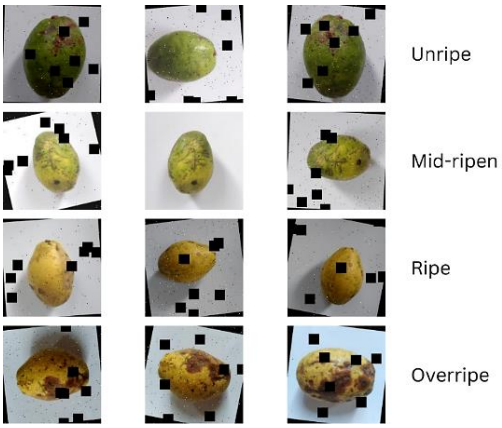


**Figure 3.** Examples of the Ambarella fruit dataset following the use of image augmentation techniques.

Figure 3 shows several results after applying augmentation to the training images. The images with black boxes are the results of applying the cutout augmentation. The cutout was applied randomly to each image thus resulting in a relatively unique pattern for each image. The images in figure 3 show the result after applying salt and pepper noise as well, thus giving small black and white spots on each image. The application of salt and pepper noise is random in nature as well. The results of the rest of augmentation techniques (blurring, rotation and flipping), are shown in figure 3. A rotated image is indicated with black parts of the image as a result of transforming the original image based on the degree of rotation. The rotation applied is in the range of ±15°. The final number of images obtained upon utilising image augmentation increases into 972 images. While image augmentation improves data diversity, resizing all images to 224 × 224 pixels, which is the YOLOv8 model's input size, remains crucial for compatibility. Therefore, all the images are resized into the required size.

### 2.2 Training

This work proposes the following method shown in figure 4. While previous research relied on established models like EfficientNets, AlexNet, VGG-16, and InceptionV3, the evolving landscape demands exploration of newer architectures. This work focuses on leveraging the potential of cutting-edge models like YOLOv8 to address the task at hand, thus highlighting our contribution (indicated by the blue box), differs from the previous works.

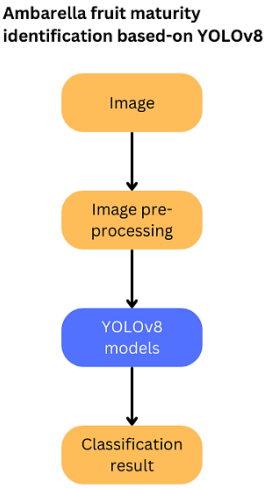


**Figure 4.** Proposed method in this work

As shown in figure 4, the first stage is to load an image for further processing. The image needed in this case is a coloured image (Red Green Blue – RGB) image. The following process after loading the image is the pre-processing stage that is carried out to prepare the image before being fed to the models. As mentioned previously, the required size for the YOLOv8 classification architecture is 224 × 224 pixels, thus in the pre-processing stage, the input image is resized into the appropriate size required by the model.

The YOLOv8 itself is the newest architecture in the YOLO family. Through their publication titled "You Only Look Once: Unified, Real-Time Object Detection" [20], Joseph Redmon and his colleagues presented YOLO to the community of researchers working within the scope of computer vision in the year 2015.
The object recognition issue was rethought in the study as a single-pass regression task, beginning with picture pixels and progressing on to bounding box and class probability estimates. The unified strategy that they used

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

made it possible to estimate probabilities simultaneously for several different classes and bounding boxes, which resulted in an increase in both speed and accuracy. Rapid progress has been made in the YOLO concept ever since it was first introduced in the year 2015.

Officially referred to as YOLO9000, the YOLOv2 was released in 2016, and the YOLOv3 was released in 2018. Other models were produced by a variety of authors, such as YOLOv4 in 2020, which was suggested by Bochkovskiy, Wang, and Liao [22], and YOLOv5 (formally known as Ultralytics YOLOv5) in 2020, which was offered by Glenn Jocher [23]. The original creator, Joseph Redmon, made the decision to halt the development of YOLO in the future. YOLOv6 was presented by the researchers from Meituan in the year 2022 [24], while the authors of YOLOv4 presented YOLOv7 in the same year [25]. The most recent upgrade, the YOLOv8, which was released in the beginning of 2023, was produced by the Ultralytics team [26], which considerably extended the accuracy and efficiency of the earlier versions available. There are a variety of important computer vision tasks that may be accomplished using the YOLOv8 model. These tasks include object segmentation, object identification, and image classification. The YOLOv8 is offered in a variety of sizes, beginning with the more compact YOLOv8n, moving up to the YOLOv8s, YOLOv8m, and YOLOv8l, and culminating in the YOLOv8x, which is the biggest variant. While the larger YOLOv8 variants, YOLOv8l and YOLOv8x, boast superior accuracy in object detection, their significant size comes at a cost. These complex models require more processing power to run inferences, making them sluggish on standard computers or mobile devices. This is particularly true for resource-constrained devices like Raspberry Pi, which lack the dedicated graphical processing units (GPUs) found in high-performance machines. Without a powerful GPU to accelerate computations, deploying these larger YOLOv8 models will inevitably result in slow inference speeds.

To achieve higher inference speed on standard computer, this work utilizes the YOLOv8n, YOLOv8s, and YOLOv8m variants due to their compact size, enabling deployment on computationally limited devices. The training is done by allowing the YOLOv8n to predict on the maturity level of a given ambarella fruit. The models were trained for 100 epochs. The AdamW optimizer were utilized with learning rate of 0.000714 and momentum of 0.9. Training the models were done through the use of Kaggle platform and utilize the available GPU for accelerated training of the models.

### 2.3 Evaluation and Deployment

Evaluation is needed for each model in order to know the performance after training. To evaluate the performance of each model, the accuracy and loss are observed, and in addition are the precision and recall. The formula to calculate the accuracy, precision, and recall are shown in equation 1, 2, and 3, respectively.

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

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

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

The terms TP, TN, FN, and FP represent the key evaluation metrics in classification tasks: true positive rate, true negative rate, false negative rate, and false positive rate, respectively. These numbers are used to calculate the accuracy, precision, and recall of the model. During training and validation, the accuracy was automatically performed by the Ultralytics. The model that has been evaluated is then deployed. The deployment was done through the use of web-based application. The backend of the web application is based on python. The aim of deployment is to enable user to upload ambarella fruit images and let the model predict on the given image.

## 3. RESULT AND DISCUSSION

This section presents the results of training YOLOv8n, YOLOv8s, and YOLOv8m models on the Ambarella fruit dataset. To leverage available resources, training for all models was conducted on Kaggle, utilizing the provided GPU. Instead of starting from scratch, we employed pre-trained versions of the YOLOv8 models, accessed through the Ultralytics library [26]. This approach not only saved training time but also allowed us to benefit from the base features learned on large datasets like ImageNet [27], [28]. All models were trained for 100 epochs. During each epoch, the weights were updated for 57 iterations. The results for all models are shown in image 4 and in detailed in table 1.

**Table 1.** Training results of each model

| Model | Test Accuracy (%) | Loss | |
|---|---|---|---|
| | | Train | Validation |
| YOLOv8n | 100 | 0.0219 | 0.74512 |
| YOLOv8s | 100 | 0.00286 | 0.74438 |

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

| Model | Test Accuracy (%) | Loss | |
|---|---|---|---|
| | | Train | Validation |
| YOLOv8m | 100 | 0.0092 | 0.7439 |

As shown in table 1, all three YOLOv8 models (YOLOv8n, YOLOv8s and YOLOv8m) achieved 100% test accuracy, indicating excellent performance in correctly identifying objects within the test dataset. Training losses varied, with YOLOv8s demonstrating the lowest training loss (0.00286), followed by YOLOv8m (0.0092) and YOLOv8n (0.0219). This suggests that YOLOv8s might have converged more efficiently during training. Validation losses were relatively close across models, ranging from 0.7439 to 0.74512. This implies consistency in performance across the models when evaluated on unseen data. Figure 5 depicts the training procedure carried out by each YOLOv8 model, represented by the fluctuating patterns of their learning curves.

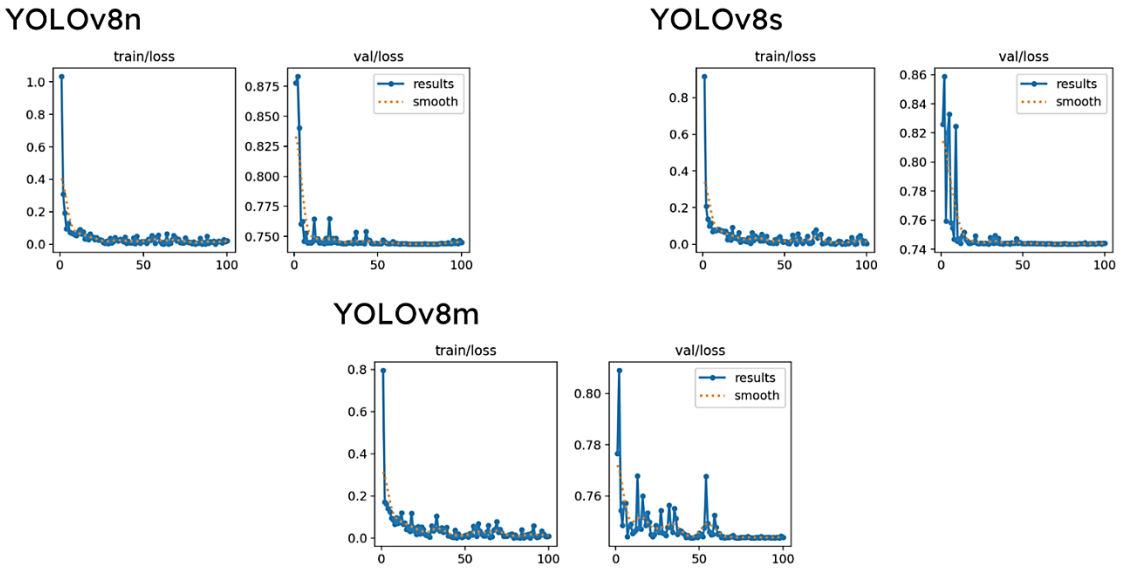


**Figure 5.** Results of training the YOLOv8n, YOLOv8s and YOLOv8m

Figure 5 shows the different training progress achieved by each model. The YOLOv8n model initially exhibits promising learning behaviour. Its loss curves show a steady and significant decrease on both the training and validation sets in the early epochs. For instance, the training loss might start above 1.0 and plummet below 0.2 within the first four epochs, suggesting the model is rapidly grasping the training data. Similarly, the validation loss might experience a substantial drop, potentially reaching values around 0.76. However, this initial progress is challenged by fluctuations in the validation loss later in training. While the training loss stabilizes around 0.2 for the next 100 epochs, indicating the model is effectively learning the training data, the validation loss exhibits more erratic behaviour. During the first 50 epochs following the initial decrease, the validation loss can fluctuate and even show temporary increases up to 0.76, however the loss decreases in the range of 0.74 for the last 50 epochs.

Compared to YOLOv8n, YOLOv8s starts with a higher initial loss, exceeding 0.8 on both the training and validation sets. However, this is still an improvement over YOLOv8n's initial loss. During the initial training stage, YOLOv8s' training loss significantly drops and stabilizes below 0.2, exhibiting more fluctuations compared to YOLOv8n but remaining consistently low. However, YOLOv8s initially struggles with larger validation losses. The validation loss in the early epochs experiences significant fluctuations, dropping and rising between 0.76 and even exceeding 0.8. Fortunately, after around 10 epochs, the validation loss stabilizes around 0.74 and remains steady for the rest of the training.

YOLOv8m exhibits the most significant training loss fluctuations compared to YOLOv8n and YOLOv8s. Similar to the other models, its initial loss on both training and validation sets starts high, around 0.8 or even higher. Throughout training, the training loss for YOLOv8m experiences more substantial fluctuations than the previous models; however, it remains consistently below 0.2 after the initial drop. The validation loss presents a similar challenge. During the first 50 epochs, it undergoes significant fluctuations, even exceeding 0.76 on several occasions. Fortunately, like the other models, the validation loss eventually stabilizes around 0.75 towards the end of training.

The performance of our models can be further evaluated using the confusion matrix presented in figure 6. This matrix visualizes the distribution of true labels (actual fruit ripeness) versus the predicted labels (model's classification) for the validation set. The validation set consists of 38 images: 12 mid-ripen, 9 overripe, 7 ripe, and 10 unripe. Examining the confusion matrix enables us to compute accuracy and recall measures, which offer valuable information about the model's capacity to accurately classify each maturity category. The background

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

class is not related to any other class, but by default was added when training using Ultralytics. The addition of background class does not affect the result.

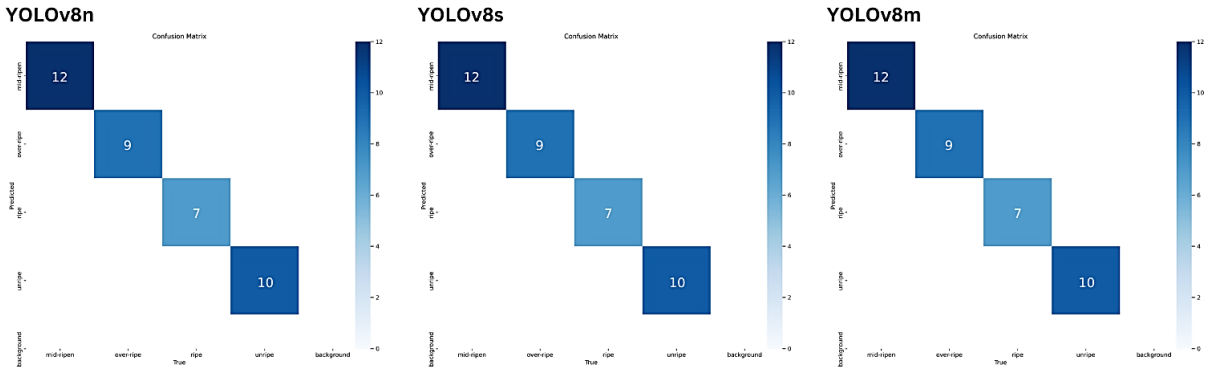


**Figure 6.** Confusion matrix of each model.

Based on the results showed in the confusion matrix on figure 6, the precision and recall of each model can be calculated using equation 2 and 3, respectively. The calculation is as follows, by taking example from the YOLOv8n confusion matrix:

$$Mid-ripen\ class\ precision = \frac{12}{12+0} = 1$$

$$Mid-ripen\ class\ recall = \frac{12}{12+0} = 1$$

$$Overripe\ class\ precision = \frac{9}{9+0} = 1$$

$$Overripe\ class\ recall = \frac{9}{9+0} = 1$$

$$Ripe\ class\ precision = \frac{7}{7+0} = 1$$

$$Ripe\ class\ recall = \frac{7}{7+0} = 1$$

$$Unripe\ class\ precision = \frac{10}{10+0} = 1$$

$$Unripe\ class\ recall = \frac{10}{10+0} = 1$$

An ideal scenario is observed in the validation set, where all models (YOLOv8n, YOLOv8s, and YOLOv8m) achieved perfect precision and recall of 100% for each ripeness class. This indicates that the models made no false positive (FP) or false negative (FN) predictions. While promising, it is important to acknowledge that validation sets are controlled environments. Further evaluation on a more diverse test set will provide a more robust assessment of the models' generalizability.

To gain a deeper understanding of our findings, a comparison with previous works is crucial. This section compares our deep learning-based approach for fruit ripeness classification with relevant prior research. The results of this comparison are presented in Table 2. The findings demonstrate that, in comparison to earlier studies, the suggested models attained a notable level of accuracy. By using MobileNetV2, the work of [17] was able to determine the maturity of fruit with 100% accuracy; this study can also achieve a similar level of accuracy. Considerable improvements in training accuracy are shown when compared to the comparable work of [19]. The models utilized in this work all achieved 100% training accuracy, compared to the best training accuracy of 97.71% attained by the prior work. This demonstrates that, when it comes to the image classification job, the current YOLOv8 family outperforms the earlier models.

**Table 2.** Result comparison with previous works

| Work | Model | Accuracy (%) |
|---|---|---|
| [12] | Proposed CNN | 91.91 |
| [16] | VGG-16 | 90 |
| [17] | MobileNetV2 | 100 |
| [18] | YOLOv3 | 99% |
| [19] | EfficientNetV2B0, EfficientNetV2B1, EfficientNetV2B2 | 97.33, 97.71, 96.57 |
| Ours | YOLOv8n, YOLOv8s, YOLOv8m | 100, 100, 100 |

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

Based on the data given in table 2, the YOLOv8n model has a size of 3.1 megabytes after training. The YOLOv8s model has a size of 10.4 megabytes. With a size of 31.82 megabytes, the YOLOv8m model is the one with the largest size. This shows that all three models are suitable for execution on a mobile device, as indicated by the size of the models, with YOLOv8n as the smallest of the models. On the other hand, when compared to the size of MobileNetV2, which is approximately 10 megabytes, it is close in size to the YOLOv8s, but it is far smaller than the YOLOv8m. The fact that this is the case suggests that the MobileNetV2 model is also appropriate for mobile deployment. As an example, the prediction made by YOLOv8n on the validation set is displayed in figure 7.


**Figure 7.** Sample prediction by YOLOv8n on several images from the validation set

The results on figure 7 shows that the model can correctly predict the maturity phases of the ambarella fruit exists on the images. After the models were evaluated, they were used and deployed through a web-based application. This application allows users to input images of ambarella fruit, and the model then makes a prediction based on the image that was supplied. The application that allows users to upload data and make predictions is depicted in figure 8. The application gives the user the ability to upload multiple images. The example that follows illustrates when the user uploaded four pictures of ambarella fruit. This web-based application was developed using the Dash library. To facilitate deployment and improve efficiency, each model was converted into a TFLite format. This conversion significantly reduces the model size, making the application more lightweight and suitable for deployment on various platforms.

The web application offers a user-friendly interface for image-based Ambarella fruit maturity assessment. Users can conveniently upload their chosen images for analysis. Once uploaded, the application seamlessly handles the image processing and model inference in the background, providing the inference results. The uploaded image undergoes several steps before the maturity prediction is displayed. First, in the pre-processing step, the image is resized to a specific size required by the chosen model. In this case, the model expects images of 224 × 224 pixels. This ensures compatibility between the image and the model's defined architecture. Following pre-processing, the resized image is fed into the selected model. This triggers the model's inference process and the model generates a set of confidence scores, one for each maturity class (ripe, unripe, mid-ripen, and overripe). These scores represent the model's certainty level for each class, indicating how strongly the model believes the Ambarella fruit in the image belongs to that particular maturity stage. The class with the highest confidence score is taken and displayed as the predicted maturity of the Ambarella fruit. This provides a clear and concise prediction for the user.

For instance, as illustrated in figure 8, the model predicts a confidence level of 99.61% for ripe, unripe, and overripe images. This indicates a very high level of certainty in these predictions. However, the confidence

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

score for a mid-ripen image is lower (89.06%). This suggests that the model is still confident in its prediction of mid-ripen, but with a slightly lesser degree of certainty compared to the other classes. By presenting these confidence scores, the application allows users to not only receive a maturity prediction but also gain insights into the model's reasoning process.

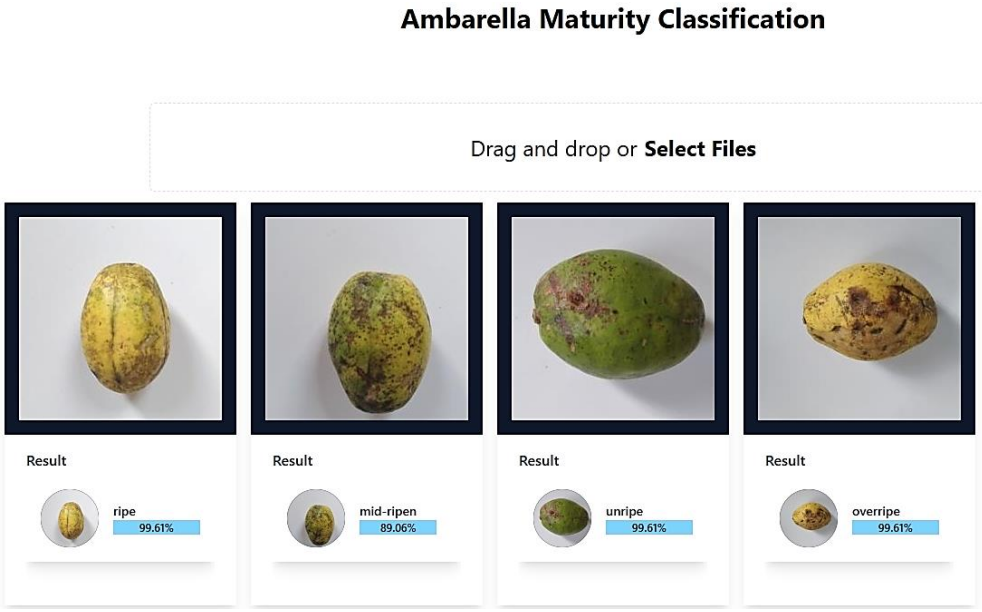


**Figure 8.** Sample predictions using the web-based application.

One of the limitations of this work is the whole-image classification approach. While effective in controlled settings, this method fails to differentiate between the Ambarella fruit and irrelevant background elements like leaves, branches, flowers, or the sky in real-world farm deployments. This poses a challenge for computer vision tasks, demanding a more fine-grained approach. To address this limitation, future research should explore object detection techniques utilizing deep learning models. By focusing on Ambarella fruit detection, the model can learn to isolate and classify the fruit of interest, regardless of the surrounding environment. Research can be focused on the creation of a farm robot which can navigates through the Ambarella orchards without the need of human control, combined with the ability gained from utilizing computer vision model.

Another limitation of this work lies in the dataset itself. The current dataset is restricted to controlled environments, lacking the diversity of real-world scenarios. To address this, future work should involve capturing a more extensive dataset encompassing a wider variety of Ambarella fruit appearances and backgrounds. These images would then require thorough preparation and annotation to train an object detection model effectively. Additionally, data augmentation techniques such as random augmentation, CutMix, and Mixup can be employed to artificially expand the dataset and enhance the model's robustness on unseen variations during deployment.

## 4. CONCLUSION

This work explores automatic Ambarella fruit maturity classification using state-of-the-art YOLOv8 models, specifically the compact YOLOv8n, YOLOv8s, and YOLOv8m variants. These models were chosen for their small size, making them suitable for deployment on resource-constrained environments like Ambarella farms.

Initial evaluation on the training set yielded impressive results. All three YOLOv8 models achieved 100% accuracy in classifying ripeness stages. The YOLOv8s model even achieved the lowest training loss of 0.00286, indicating exceptional learning efficiency. This strong performance was mirrored in precision and recall metrics, where all models achieved a score of 100%. To facilitate user interaction, the trained models were deployed through a web application. Users can upload Ambarella fruit images, and the chosen model delivers the predictions. Looking ahead, there's potential for further improvement. Expanding the dataset with a wider variety of Ambarella fruit images at various ripeness stages can enhance the models' accuracy and generalizability in real-world scenarios. Additionally, exploring fruit segmentation techniques before classification could enable direct ripeness prediction on the tree, eliminating the need for fruit picking beforehand – a valuable advancement for non-destructive monitoring in Ambarella orchards. Another challenging task is to create a robot that can navigates through the orchard and combined with the computer vision model to automatically monitor the Ambarella fruit in the orchard.

ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

# REFERENCES

[1] N. Ngadi *et al.*, "Challenge of Agriculture Development in Indonesia: Rural Youth Mobility and Aging Workers in Agriculture Sector," *Sustainability 2023, Vol. 15, Page 922*, vol. 15, no. 2, p. 922, Jan. 2023, doi: 10.3390/SU15020922.

[2] F. Sarie and I. Harsono, "Measuring the Impact of Pesticide Use, Labor Availability, and Agricultural Technology on Vegetable Farming Efficiency in Central Java," *West Science Interdisciplinary Studies*, vol. 2, no. 01, pp. 244–253, Jan. 2024, doi: 10.58812/WSIS.V2I01.615.

[3] M. M. Lubis, M. Abror, and J. Djuhjana, "Effect of fertilizer discount on fresh fruit bunches production and Nutrients in North Sumatra Indonesia," *IOP Conf Ser Earth Environ Sci*, vol. 1308, no. 1, p. 012004, Feb. 2024, doi: 10.1088/1755-1315/1308/1/012004.

[4] É. M. dos Santos *et al.*, "Spondias sp: Shedding Light on Its Vast Pharmaceutical Potential," *Molecules 2023, Vol. 28, Page 1862*, vol. 28, no. 4, p. 1862, Feb. 2023, doi: 10.3390/MOLECULES28041862.

[5] F. Jemziya, A. M. Rikasa, B. S. G. M. Basnayake, M. B. F. Jemziya, R. T. B. Rambodagalla, and A. M. Rikasa, "Development of Ambarella (Spondias dulsis) fruit pulp incorporated ice cream," 2022, Accessed: Jan. 02, 2024. [Online]. Available: https://www.researchgate.net/publication/370939069

[6] I. Hayati, A. Hartana, N. Ratna Djuita, and N. Sri Ariyanti, "Morphological Variation of Kedondong (Spondias Dulcis Parkinson) in Central Part of Sumatra," *Floribunda*, vol. 6, no. 8, Apr. 2022, doi: 10.32556/floribunda.v6i8.2022.375.

[7] P. Tyagi, R. Semwal, A. Sharma, U. S. Tiwary, and P. Varadwaj, "E-nose: A low-cost fruit ripeness monitoring system," *Journal of Agricultural Engineering*, vol. 54, no. 1, Nov. 2022, doi: 10.4081/jae.2022.1389.

[8] M. Rizzo, M. Marcuzzo, A. Zangari, A. Gasparetto, and A. Albarelli, "Fruit ripeness classification: A survey," *Artificial Intelligence in Agriculture*, vol. 7, pp. 44–57, Mar. 2023, doi: 10.1016/J.AIIA.2023.02.004.

[9] P. L. I. Jayarathna, J. A. E. Jayawardena, and C. Vanniarachchy, "Identification of Physical, Chemical Properties and Flavor Profile of Spondias dulcis in Three Maturity Stages," *International Research Journal of Advanced Engineering and Science*, vol. 5, no. 1, pp. 208–211, 2020, doi: 10.5281/zenodo.3695472.

[10] P. Siricharoen, W. Yomsatieankul, and T. Bunsri, "Fruit maturity grading framework for small dataset using single image multi-object sampling and Mask R-CNN," *Smart Agricultural Technology*, vol. 3, p. 100130, Feb. 2023, doi: 10.1016/J.ATECH.2022.100130.

[11] M. Momeny *et al.*, "Greedy Autoaugment for classification of mycobacterium tuberculosis image via generalized deep CNN using mixed pooling based on minimum square rough entropy," *Comput Biol Med*, vol. 141, p. 105175, Feb. 2022, doi: 10.1016/J.COMPBIOMED.2021.105175.

[12] H. Khosravi, S. I. Saedi, and M. Rezaei, "Real-time recognition of on-branch olive ripening stages by a deep convolutional neural network," *Sci Hortic*, vol. 287, p. 110252, Sep. 2021, doi: 10.1016/J.SCIENTA.2021.110252.

[13] M. Tools, B. Xiao, M. Nguyen, and W. Qi Yan, "Fruit ripeness identification using YOLOv8 model," *Multimedia Tools and Applications*, vol. 83, no. 9, pp. 28039–28056, 2023, 123AD, doi: 10.1007/s11042-023-16570-9.

[14] L. Chuquimarca, B. Vintimilla, and S. Velastin, "Banana Ripeness Level Classification Using a Simple CNN Model Trained with Real and Synthetic Datasets," INSTICC, Mar. 2023, pp. 536–543. doi: 10.5220/0011654600003417.

[15] "Deep Learning for Faces on Orphanage Children Face Detection." Accessed: May 31, 2024. [Online]. Available: https://jurnal.stmikroyal.ac.id/index.php/jurteksi/article/view/1858

[16] J. Pardede, B. Sitohang, S. Akbar, and M. L. Khodra, "Intelligent Systems and Applications," *Intelligent Systems and Applications*, vol. 2, pp. 52–61, 2021, doi: 10.5815/ijisa.2021.02.04.

[17] E. Tapia-Mendez, I. A. Cruz-Albarran, S. Tovar-Arriaga, and L. A. Morales-Hernandez, "Deep Learning-Based Method for Classification and Ripeness Assessment of Fruits and Vegetables," *Applied Sciences*, vol. 13, no. 22, p. 12504, Nov. 2023, doi: 10.3390/app132212504.

[18] A. Ajil, A. Ali, A. K. K. Reddy, L. V. C. Reddy, A. V. G. S. Reddy, and O. Goud, "Fruit Ripeness Assertion Using Deep Learning," *International Journal of Human Computations & Intelligence*, vol. 2, no. 2, pp. 63–72, Apr. 2023, doi: 10.5281/ZENODO.7900479.

[19] R. E. Saragih, Y. Roza, and A. Rezki Purnajaya, "Ambarella Fruit Ripeness Classification based on EfficientNet Models," 2022.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02640

[21] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," *Machines 2023, Vol. 11, Page 677*, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/MACHINES11070677.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, Accessed: Sep. 07, 2023. [Online]. Available: http://arxiv.org/abs/2004.10934

[23] G. Jocher, "Ultralytics YOLOv5." 2020. doi: 10.5281/zenodo.3908559.

[24] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," Sep. 2022, Accessed: Sep. 07, 2023. [Online]. Available: https://arxiv.org/abs/2209.02976v1

[25] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Jul. 2022, Accessed: Sep. 07, 2023. [Online]. Available: https://arxiv.org/abs/2207.02696v1

[26] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8." 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[27] T. Akter *et al.*, "Improved transfer-learning-based facial recognition framework to detect autistic children at an early stage," *Brain Sci*, vol. 11, no. 6, 2021, doi: 10.3390/brainsci11060734.

**Journal of Computer System and Informatics (JoSYC)**
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 3, May 2024, Page 719-728
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i3.5123

[28] R. Mishra, S. Goyal, T. Choudhury, and T. Sarkar, "Banana ripeness classification using transfer learning techniques," *Proceedings of International Conference on Computing, Communication, Security and Intelligent Systems, IC3SIS 2022*, 2022, doi: 10.1109/IC3SIS54991.2022.9885244.