Mango and Banana Ripeness Detection based on Lightweight YOLOv8

Raymond Erz Saragih[*1], Akhmad Rezki Purnajaya[2], Ilwan Syafrinal[3], Yonky Pernando[4], Yodi[5]
[1,4] Program Studi Teknik Informatika, Fakultas Komputer, Universitas Universal
[2,3] Program Studi Teknik Perangkat Lunak, Fakultas Komputer, Universitas Universal
[5] Program Studi Sistem Informasi, Fakultas Komputer, Universitas Universal
Kompleks Maha Vihara Duta Maitreya, Batam, 29456, Kepulauan Riau, Indonesia
Email: [1]raymondes@uvers.ac.id, [2]rezki.purnajaya@uvers.ac.id, [3]ilwan@uvers.ac.id, [4]yonkypernando194@gmail.com, [5]yodi@uvers.ac.id

Abstrak. *Buah-buahan seperti pisang dan mangga dipanen setelah mencapai tingkat kematangan tertentu. Secara tradisional, petani mengandalkan pemeriksaan manual untuk menentukan kematangan, yang merupakan proses yang melelahkan, memakan waktu, mahal, dan subjektif. Penelitian ini mengusulkan pendeteksi kematangan buah pisang dan mangga otomatis menggunakan teknologi visi komputer. Pisang dan mangga yang dideteksi dibagi menjadi dua kelas, yakni matang dan belum matang. Arsitektur YOLOv8 digunakan sebagai pendeteksi. Tiga varian YOLOv8, YOLOv8n, YOLOv8s, dan YOLOv8m, diinvestigasi kinerjanya. Hasil penelitian menunjukkan bahwa YOLOv8s mencapai kinerja keseluruhan tertinggi, dengan recall 0,9991 dan mean Average Precision (mAP) 0,8897. Sementara YOLOv8m mencapai presisi tertinggi yaitu 0,9995, YOLOv8n muncul sebagai model terkecil, sehingga cocok untuk digunakan pada perangkat dengan sumber daya terbatas.*
Kata Kunci: *mangga, pisang, visi komputer, YOLOv8*

Abstract. *Fruits like bananas and mangoes are harvested after reaching a specific ripeness stage. Traditionally, farmers rely on manual inspection to determine ripeness, a process that can be tedious, time-consuming, expensive, and subjective. This work proposes an automatic bananas and mangoes ripeness detector utilizing computer vision technology. The detected bananas and mangoes fall into two classes: ripe and unripe. The state-of-the-art YOLOv8 architecture serves as the core of the detector. Three YOLOv8 variants, YOLOv8n, YOLOv8s, and YOLOv8m, were investigated for their performance. Results show that YOLOv8s achieved the highest overall performance, 0.9991 recall, and a mean Average Precision (mAP) of 0.8897. While YOLOv8m achieved the highest precision of 0.9995, YOLOv8n is the most miniature model, making it suitable for deployment on devices with limited resources.*
Keywords: *bananas, mangoes, computer vision, YOLOv8*

1. Introduction

Fruits are essential to a healthy diet, providing humans with vital nutrients for growth and overall well-being [1]. The quality of the fruits produced must meet a good standard and be maintained at a high standard, as it is crucial from the economic value perspective [2]. Plenty of fruits are sold in the market, such as bananas, mango, watermelon, dragon fruit, pear, pineapple, and melon. The harvested fruits are then sold to a market, store, or food factory for further consumption. Traditionally, the farmers determine the ripeness of the fruits manually [1], [3]. However, the problems that arise in determining the ripeness manually are that the job is tedious, time-consuming, expensive, and on some subjective to the examiner [2], [4], [5], [6]. To tackle these problems, it is imperative to build an automated ripeness of fruit classifier using state-of-the-art technology.

Over the years, computer vision has found its way to tackle various tasks, including agriculture, related explicitly to fruit ripeness [2], [3], [4], [7]. Computer vision consists of three main tasks: image classification, object detection, and object segmentation. These tasks are accomplished by utilizing deep learning models, such as the renowned Convolutional Neural Network (CNN) [8], [9]. A variety of Convolutional Neural Network architectures have been proposed since the first LeNet to the likes of VGGs, MobileNets, EfficientNets [9], and You Only

Look Once (YOLO) [10]. The latter currently holds the state-of-the-art status as an object detector architecture and has been applied for various purposes.

Deep learning has emerged as a powerful computer vision technique for fruit detection in images and videos. This technology enables the localization and classification of fruits, with applications ranging from ripeness recognition and yield prediction, automation in harvesting robots, fruit quality assessment, and fruit estimation and counting [4]. Computer vision offers a valuable approach to several aspects of fruit production, including determining ripeness, harvesting, and counting. One key benefit of this technology is its non-destructive nature [6], eliminating potential damage to the fruit. Additionally, computer vision enables automation [4], streamlining these processes and improving efficiency.

Among the variety of fruits, this work appoints two specific fruits, banana and mango. After rice, maize, and wheat, bananas rank as one of the vital crops for human consumption worldwide and are widely consumed across Africa, Latin America, and Asia [6]. Banana ripeness can be determined through manual visual inspection. However, a non-destructive approach must be made through computer vision [6], [11]. Another fruit known commercially is the mango fruit [12]. The ripeness of mango can be visually known; however, manually classifying the ripening stage of mango can lead to inconsistencies [13].

This work proposes an automatic system for assessing the ripeness of bananas and mangoes using the cutting-edge YOLO architecture. This approach enables non-destructive and automated ripeness assessment by developing a model to detect and identify fruit type and ripeness stage. The object detection architecture used in this work is the current YOLOv8, specifically investigating light architectures, such as YOLOv8n, YOLOv8s, and YOLOv8m. These light architectures offer the potential for deployment on resource-constrained devices such as Raspberry Pi.

2. Literature Review

Research on applying computer vision to the agricultural sector has been done throughout the years to determine fruit maturity. Worasawate et al. developed a supervised and unsupervised machine learning approach to evaluate the maturity stage of one of the mango varieties called "Nam Dok Mai Si Tong" [12]. To visually represent the distribution of biochemical information regarding mangoes and identify potential outliers, the authors employed the k-means unsupervised learning algorithm. Subsequently, the authors leveraged three supervised machine learning techniques commonly used in mango ripeness stage classification: Feedforward Artificial Neural Network (FANN), Support Vector Machine (SVM), and the Gaussian Naive Bayes (GNB). The GNB, SVM, and FANN achieved an average accuracy of 73%, 75%, and 85%, respectively. Saragih and Emanuel proposed a method for classifying the ripeness of bananas based on images using the MobileNet V2 and NASNetMobile models. [14]. The methodology employed in their research was utilizing transfer learning by fine-tuning both previously trained models. The findings demonstrate that the MobileNet V2 model has attained a remarkable accuracy of 96.18% in correctly determining the maturity phases of bananas. Xiao et al. proposed a maturity stage identification of apple fruit using Transformers and YOLO5 architecture [15]. The best model achieved by YOLO, although the transformer, specifically the Swin Transformer, achieved a fast detection rate. The average precision of the YOLOv5 model is above 0.999.

Xiao et al. proposed the identification of apple and pear ripeness using YOLOv5 [7]. In their work, they incorporated transformers into the YOLO model. Four YOLOv5 variations were investigated: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The YOLOv5s achieved the highest of 0.9995 in terms of AP50 and 0.9390 in terms of AP50-95. The YOLOv5m, YOLOv5l, and YOLOv5x achieved a higher result in terms of AP50, which is 0.9996; however, with a different result in terms of AP50-95: The YOLOv5m achieved 0.9540, YOLOv5l achieved 0.9850, and YOLOv5x achieved 0.9770. Novian et al. [16] investigated using deep learning to detect palm oil fruit bunch (FFB). The overall performance of four distinct deep learning architectures was compared: InceptionV2, ResNet50, ResNet101, and Inception ResNet V2. These architectures are convolutional neural networks (CNNs) known for their effectiveness in image recognition and object detection tasks. The results were promising, with all four models achieving F1 scores exceeding 80%. Notably, InceptionV2 and ResNet50 performed exceptionally well, reaching a peak F1 score of 90%.

In the work of Xu et al., a YOLO-based model was proposed to detect the Jujube fruit and determine its ripening state [4]. The focus of the proposed model was to be lightweight yet still maintain the robustness of the model. A dataset consisting of 1959 was used to train each model. The proposed model (YOLO-Jujube) achieved 88.8% average precision (AP), outperforming YOLOv3, YOLOv4, YOLOv5s, and YOLOv7. Xiao et al. proposed an approach to determine the maturity of fruits, specifically apple and pear fruits, using the YOLOv8 model [17]. The authors investigated three varieties of YOLOv8, which are YOLOv8n, YOLOv8m, and YOLOv8x. The results show that the highest mAP50-95 are 0.993, 0.994, and 0.993, respectively, while all models achieved mAP50 of 0.995. Those results were achieved after training the models for 100 – 200 epochs.

The previous works have shown that the deep learning approach for object detection was successful when applied to detecting fruit and determining ripeness. However, several aspects can be improved from the previous works, such as the work of [15], the environment in which the apple fruit placed was under control. In real-world cases, the fruit exists on a plantation with varying backgrounds. Previous works incorporated various object detection models. However, investigation is needed against newer state-of-the-art models, as the newer models offer better performance than the last. Although the work of Xiao et al. [17] utilized the YOLOv8, one of the investigated models is the largest in the YOLOv8 variety, the YOLOv8x. Using a large model limits the model being deployed on a resource-constrained computer. For those reasons, this work investigates the use of YOLOv8, specifically the lightweight variations, such as YOLOv8n, YOLOv8s, and YOLOv8m.

3. Research Method

3.1. Dataset

Training a model requires a dataset that shows the model in which a banana or mango is considered unripe and ripe. This work employs an existing dataset called the "Mango and Banana Dataset" [18]. The dataset consists of four classes: raw, raw mango, ripe banana, and ripe mango. Five thousand colored images have been categorized into training and testing. The training group comprises 80% of the total images, while the test group comprises 20%. Figure 1 displays a selection of samples collected from the dataset.

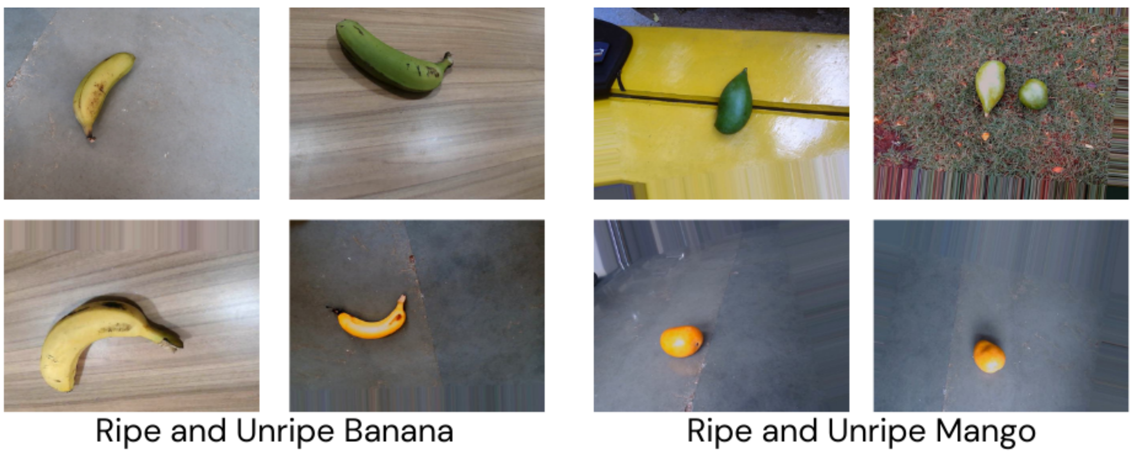

Figure 1. Samples of the ripe and unripe bananas and mangoes from the dataset [18]

As shown in Figure 1, the images were taken under various indoor or outdoor lighting conditions. This reflects the real-world scenarios in which a model is required to detect the banana or mango in varying lighting conditions. The dimension of each image is 640 × 480 pixels with corresponding metadata that specifies the bounding box coordinates for each fruit within the image. Both training and testing images underwent preprocessing steps to ensure the size complied with the model. The process consists of resizing, specifically to the size of 640 × 480 pixels. Data augmentations are applied to the training images to increase the variety and the ability to generalize better.

Data augmentation is one of the techniques used to produce a variation of training images, thus significantly increasing the number of training images and preventing the model from overfitting during training [19]. Data augmentation is crucial to training a model, as it will introduce variations to banana and mango within an image, thus creating a robust model. This work uses several augmentation techniques, such as horizontal and vertical flipping, rotation, blurring, noise, and brightness. Each method was implemented randomly to the image. Figure 2 displays the outcomes of augmentation.

Figure 2. Samples of augmentation were applied to the training set [18] with the corresponding bounding boxes

As shown in Figure 2, an image was transformed into several variations through augmentation. The purple boxes indicate the object of interest in each image, which is the raw or ripe bananas and mangoes. As we applied the transformations to each image in the dataset, the resulting pictures grew more extensive from the original 5000 images into 8000 images, with a total of 8,096 annotations. This was all accomplished using the Roboflow platform. The following Figure 3 shows the overall annotation heatmap of the dataset.

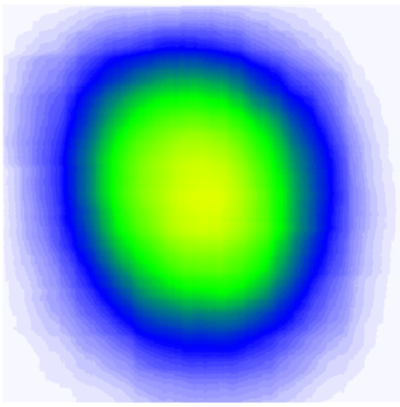

Figure 3. Annotation heatmap of the mango and banana detection dataset

The annotation heatmap shown in Figure 3 shows that the mango or banana objects contained in the dataset are mainly located in the center of the image. However, the blue-colored areas indicate that several images contain mango or banana objects that deviate from the image's center. This suggests that the mango or banana's location within the dataset image varies.

3.2. Method

Computer vision has continuously improved over the years to tackle three main tasks: image classification, object detection, and object segmentation [20]. This work aimed to develop an object identification model to identify raw or ripe bananas and mangoes accurately. Object detection aims to accurately determine the location and identity of a particular object inside an image or frame from a webcam [20]. The output of object detection is in the form of bounding boxes that specify the spatial coordinates of the object of interest. Object detection has found widespread applications in various fields, including industry [20], [21], and agriculture [22].

Various deep learning architectures were suggested to address the object detection challenge [8], [9]. The models that are used for object detection are from the early OverFeat to the rise of Spatial Pyramid Pooling net (SPP-net), Region-based CNN, Single-Shot Multibox Detector (SSD), Faster R-CNN, Fast R-CNN, and the renowned You Only Look Once model (YOLO) [9]. The latter has gone through various improvements since its inception in 2015. The initial YOLO architecture suggested by Redmon et al. treated detecting objects as a regression problem and partitioned the image into a grid of squares [10]. The YOLO object detection algorithm has rapidly evolved, with numerous versions released to improve accuracy, speed, and model size. Newer models like YOLOv4, YOLOv6, and YOLOv8 prioritize efficiency and tackle various computer vision tasks like classification, detection, and segmentation [10]. The recent addition to the YOLO family was the YOLOv8, released in the early 2023 [23]. YOLOv8 has reached state-of-the-art status and can tackle the three main tasks of computer vision: classification, detection, and segmentation. YOLOv8 also offers variations of the model, ranging from the most miniature model, YOLOv8n, to the largest model, YOLOv8x. The structure of YOLOv8 is shown in Figure 4.

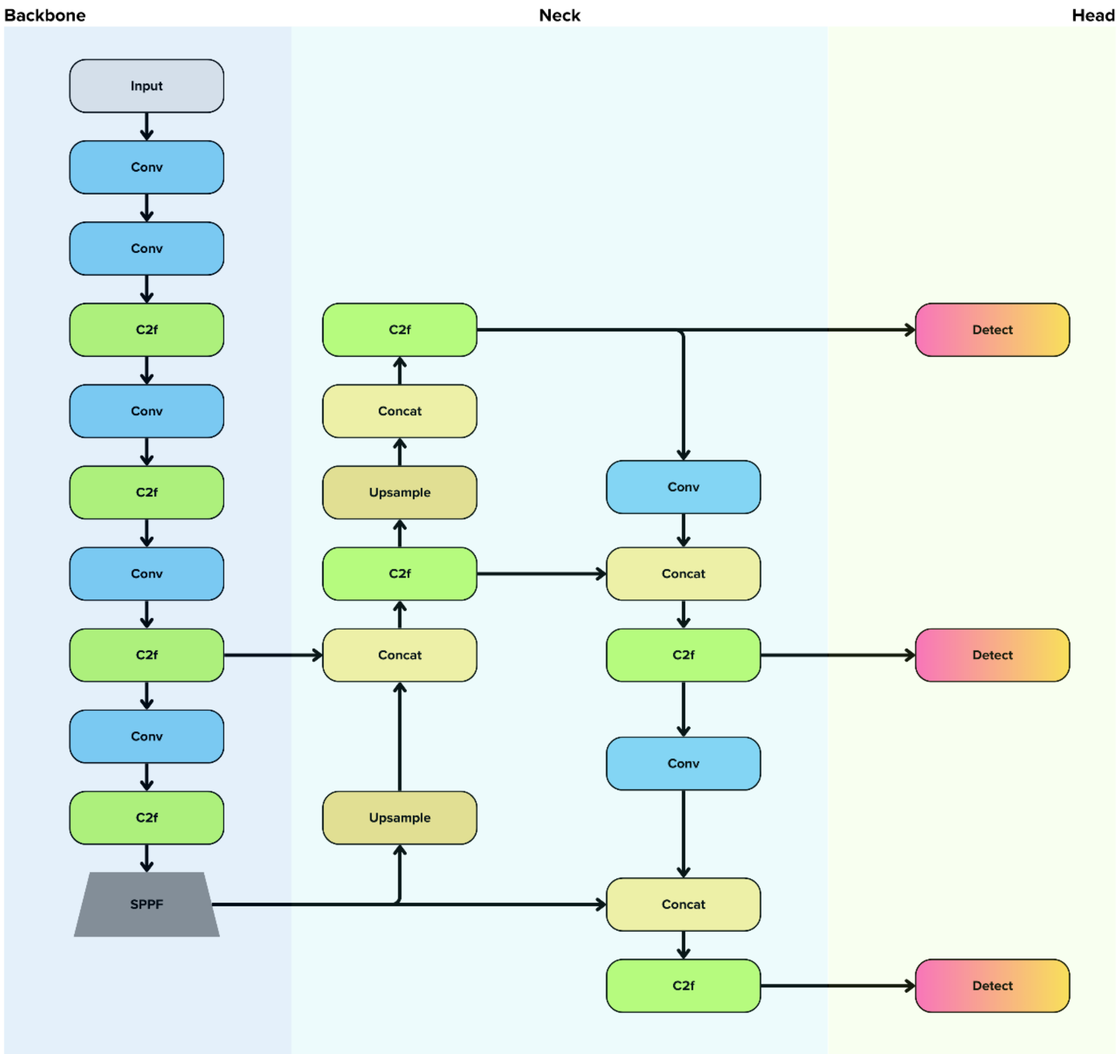

Figure 4. YOLOv8 structure

In this work, the YOLOv8 was selected as the architecture to detect banana and mango fruits based on their ripeness, considering it the newest object detection model. The comparison of performance and inference speed between YOLOv8 and the previous version has become another supporting factor in deciding on the YOLOv8 model. This work investigates the performance between three YOLOv8 variations, namely YOLOv8n, YOLOv8s, and YOLOv8m. Their size primarily drove the selection of those models. YOLOv8n to YOLOv8m are relatively small, with a base size of 6 MB, 21 MB, and 50 MB, respectively. Despite being small, these models have achieved higher performance than the previous YOLO models. Resource efficiency gives the possibility of those models intended for deployment on a device with limited resources, such as the Raspberry Pi.

As shown in Figure 4, YOLOv8 is partitioned into three components: the backbone, neck, and head. These components extract, fuse multiple features, and produce prediction outputs [21]. The primary function of the feature extraction network is to extract distinct scale features from images generated by the C2f and SPPF modules. The C2f module decreases the number of convolutional layers in the network by one, compared to the original C3 module, resulting in a lighter model. SPPF streamlines the network layers by leveraging SPP (spatial pyramid pooling) to remove unnecessary operations and enhance the speed of feature fusion. The detection layer accurately forecasts the targets' positions, categories, confidence ratings, and details.

Evaluating how well the model performed involved utilizing metrics such as Precision, Recall, as well as mean Average Precision (mAP) [8], [9]. In object detection, precision represents the ratio of accurately recognized objects to a model's total number of detections. In contrast, recall represents the proportion of all actual objects (ground truths) that were correctly detected. Finally, mean Average Precision (mAP) summarizes the overall detection performance across all object classes. It is calculated by averaging the Average Precision (AP) scores obtained individually per class. These AP scores consider both precision and recall for each class [9]. Equations 1, 2, and 3 show the Recall, Precision, and mAP formula. TP represents the proportion of actual positive cases, FN represents the proportion of false negative cases, and FP represents the proportion of false positive cases.

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{all\ ground\ truths} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{all\ detections} \tag{2}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3}$$

4. Results and Discussion

This work investigates the performance of three YOLOv8 models. Three models were chosen for comparison: YOLOv8n (YOLOv8 nano), YOLOv8s (YOLOv8 small), and YOLOv8m (YOLOv8 medium). These models represent the more minor YOLOv8 variants, with base sizes of 6 MB, 21 MB, and 50 MB for YOLOv8n, YOLOv8s, and YOLOv8m, respectively. These models could be deployed using resource-constrained devices, such as Raspberry Pi. Each model was trained for 200 epochs, with AdamW selected as the optimizer and a learning rate of 10^{-3}, following the work of [24] and [25], as the results in those works proved satisfactory. The training was done using a Tesla P100 GPU in the Kaggle workspace. The performance comparison of each model is shown in Figure 5.

The results in Figure 5 show that YOLOv8n improves consistently for 100 and reaches the maximum precision of 0.9991 on the training data for the remaining 100 epochs. In terms of recall, the YOLOv8n fluctuates for more than 100 epochs. The recall stabilizes for almost 70 epochs before improving significantly to the maximum of 0.999 and a sudden drop at the end. Regarding mAP50, YOLOv8n improves for the first 20 epochs, reaches its maximum of 0.995, and stays stagnant for the remaining epochs until 200 epochs. Finally, the mAP50-95 metric exhibits fluctuations throughout the training process. However, it demonstrates an incremental increase, achieving a peak score of 0.8854.

The YOLOv8s model exhibits consistent improvement in precision for all object classes, reaching a peak of 0.9994 after exceeding 100 epochs. However, a slight decline is observed at the very end of training. Regarding recall, the model demonstrates improvements throughout training, punctuated by occasional drops. Notably, two significant drops occur during the process. Despite these fluctuations, recall ultimately reaches a maximum of 0.9991 after 100 epochs. Unfortunately, the recall did not improve for the remainder of the training. The mAP50 metric for YOLOv8s initially exhibits fluctuations during the early stages of training. However, after surpassing 30 epochs, the curve stabilizes, reaching a peak of 0.995. Lastly, the mAP50-95 improves for the course of training. However, a significant drop occurred after the 90th epoch. After the drop, the metric improved for the rest of the training and achieved a maximum of 0.8897.

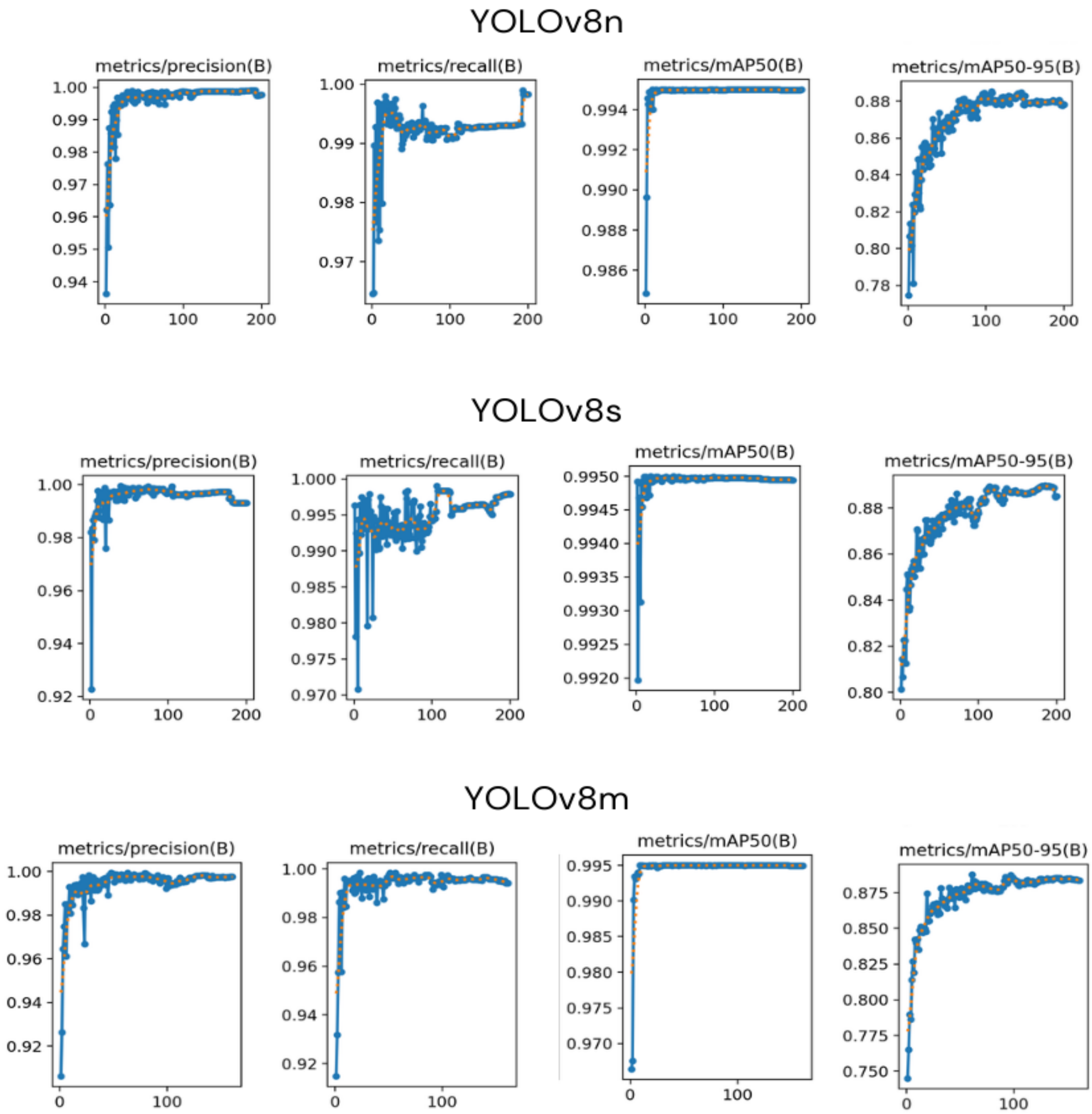

Figure 5. Performance comparison between YOLOv8n, YOLOv8s, and YOLOv8m in precision, recall, mAP50, and mAP50-95

The last model is the YOLOv8m, and the training process can be observed in Figure 4. Similar to the previous experiments, the model fluctuated during the training course. However, the training did not last for 200 epochs; it only lasted for 161 epochs, stopped by the EarlyStopping method, as there is no further improvement. The maximum precision achieved by YOLOv8m is higher than the previous models, 0.9995, after training for 60 epochs. Conversely, the recall attained a maximum of 0.9985 after training for 20 epochs, which is faster than the previous experiments using YOLOv8n and YOLOv8s. Like the YOLOv8n models, the mAP50 reaches stability only after several epochs, reaching the maximum of 0.995. In terms of mAP50-95, the graph shows stable improvement with a slight drop in the middle of the training. However, YOLOv8m achieved a maximum of 0.888 in mAP50-95, which is lower than the previous YOLOv8s. The training results show that each model achieved acceptable performance, with minimum difference.

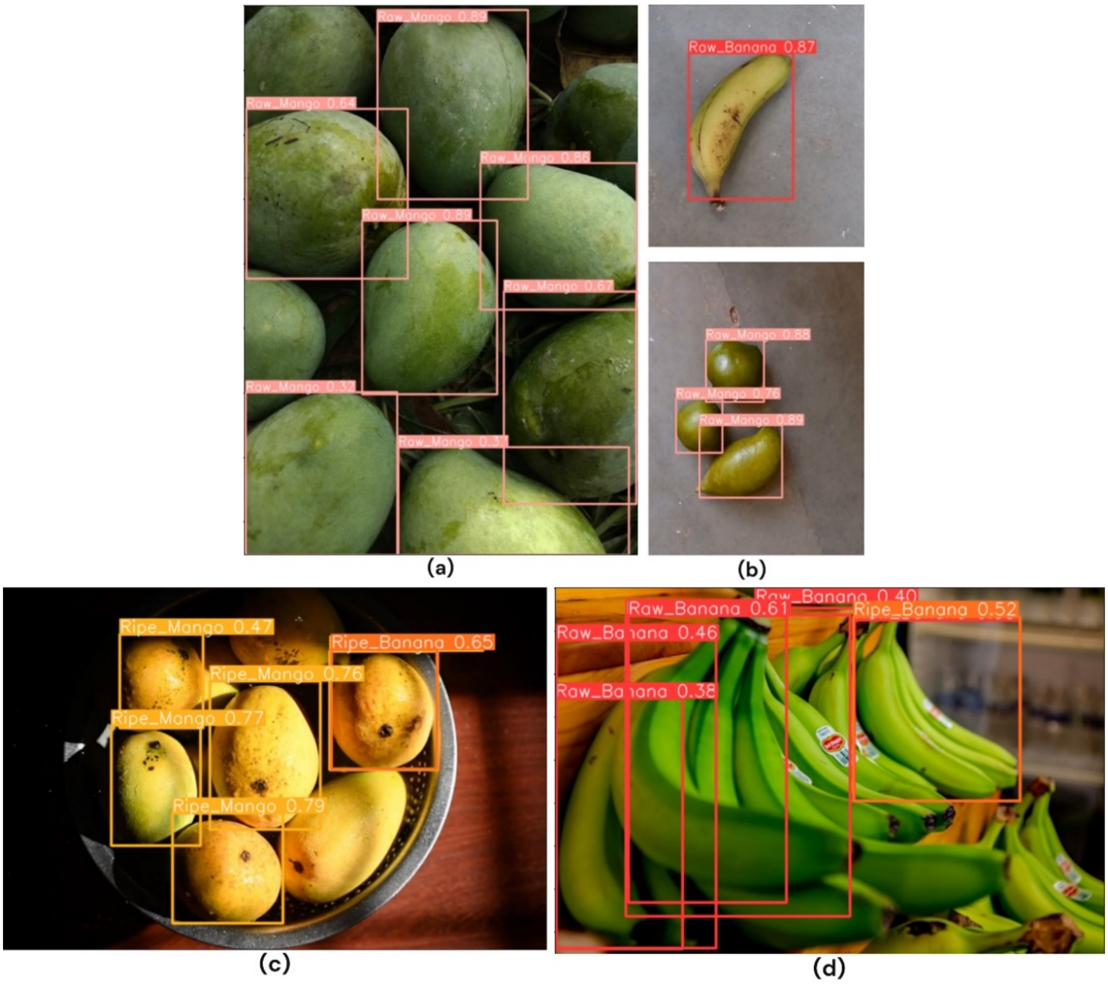

Figure 5. Results of detecting banana or mango based on the ripening stage on the image (a)[26], (b)[18], (c)[27], and (d)[28]

The model detects mango or banana in this last section on an image. The model used is the YOLOv8n. The model receives the image as an input to detect bananas or mangoes in an image. However, before further processing, the image is resized into 640 × 640 pixels, as this is the size specified while training the model; this stage is known as a pre-processing stage. The resized image is then fed into the model. The YOLOv8 model will then process the input image using the layers within, and lastly, the model will generate the coordinate of bounding boxes of the detected mangoes or bananas and the respective label. Figure 5 shows the results of using the YOLOv8n for several pictures. The pictures were taken from the test set in [18], while others were taken from Unsplash (Figure 5 (a) [26], Figure 5 (c) [27], and Figure 5 (d) [28]). The fruits are positioned under varying conditions, such as near or further from the camera. The detection results show that the YOLOv8n model can detect raw or ripe bananas and mangoes inside the test images. The confidence of each detection is acceptable as most objects differ from the training set. Although the model achieved satisfactory results, there are several drawbacks. In Figure 5 (a) [26], the model couldn't detect a clipped mango on the top left, right, or middle. In Figure 5 (c) [27], the model couldn't detect one of the ripe mangoes and falsely identified one on the top right as a ripe banana. Lastly, in Figure 5 (d), the model falsely detects unripe bananas as ripe. These results indicate potential areas for improvement in the model to address its shortcomings. Expanding the dataset to include a wider variety of banana and mango shapes (e.g., clipped, stacked) and encompassing different mango or banana varieties would likely enhance detection accuracy. Additionally, modifications to the model could be explored to improve its ability to detect small objects, such as fruit located farther from the camera.

5. Conclusions

This work proposes a detector for identifying the ripeness stage of bananas and mangoes. Three YOLOv8 models (YOLOv8n, YOLOv8s, and YOLOv8m) were trained on a mango and banana dataset. All models achieved promising concerning mean Average Precision (mAP), recall, and precision. YOLOv8m achieved the highest accuracy (0.9995), followed by YOLOv8s (0.9994) and YOLOv8n (0.9991). YOLOv8s achieved the highest recall (0.9991), followed by YOLOv8n (0.9990) and YOLOv8m (0.9985). In terms of mAP, YOLOv8s again performed best (0.8897). Notably, the models exhibited minimal performance differences across most metrics.

In terms of size, YOLOv8n is the smallest, with satisfactory performance compared to the YOLOv8s and YOLOv8m, thus offering its possibility to be deployed on resource-constrained devices. The YOLOv8n was used to detect several test images. However, the analysis identified some areas for improvement with YOLOv8n during real-world testing. These included occasional false positives (misidentifying objects) and limitations in detecting clipped fruit or fruit positioned farther from the camera. These findings highlight potential areas for further research. Expanding the dataset with various images, including clipped fruit and varying distances, could improve detection accuracy. Additionally, exploring model modifications to enhance small object detection capabilities is warranted.

References

[1] Y. J. Davur, W. Kämper, K. Khoshelham, S. J. Trueman, and S. H. Bai, "Estimating the Ripeness of Hass Avocado Fruit Using Deep Learning with Hyperspectral Imaging," *Horticulturae*, vol. 9, no. 5, p. 599, May 2023, doi: https://doi.org/10.3390/horticulturae9050599.

[2] P. Sajitha, A. D. Andrushia, N. Mostafa, A. Y. Shdefat, S. S. Suni, and N. Anand, "Smart farming application using knowledge embedded-graph convolutional neural network (KEGCNN) for banana quality detection," *Journal of Agriculture and Food Research,* vol. 14, p. 100767, Dec. 2023, doi: 10.1016/j.jafr.2023.100767.

[3] R. Thakur, G. Suryawanshi, H. Patel, and J. Sangoi, "An Innovative Approach For Fruit Ripeness Classification," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, May 2020, doi: 10.1109/iciccs48265.2020.9121045.

[4] D. Xu, H. Zhao, O. M. Lawal, X. Lu, R. Ren, and S. Zhang, "An Automatic Jujube Fruit Detection and Ripeness Inspection Method in the Natural Environment," *Agronomy*, vol. 13, no. 2, Feb. 2023, doi: 10.3390/agronomy13020451.

[5] M. Zhao, Z. You, H. Chen, X. Wang, Y. Ying, and Y. Wang, "Integrated Fruit Ripeness Assessment System Based on an Artificial Olfactory Sensor and Deep Learning," *Foods*, vol. 13, no. 5, Mar. 2024, doi: 10.3390/foods13050793.

[6] P. Baglat, A. Hayat, F. Mendonça, A. Gupta, S. S. Mostafa, and F. Morgado-Dias, "Non-Destructive Banana Ripeness Detection Using Shallow and Deep Learning: A Systematic Review," *Sensors*, vol. 23, no. 2, p. 738, Jan. 2023, doi: 10.3390/S23020738/S1.

[7] B. Xiao, M. Nguyen, and W. Q. Yan, "Fruit ripeness identification using transformers," *Applied Intelligence*, vol. 1, p. 3, Jun. 2023, doi: 10.1007/s10489-023-04799-8.

[8] J. Wang, T. Zhang, Y. Cheng, and N. Al-Nabhan, “Deep Learning for Object Detection: A Survey,” *Computer Systems Science and Engineering*, vol. 38, no. 2, pp. 165–182, 2021, doi: 10.32604/csse.2021.017016.

[9] A. B. Amjoud and M. Amrouch, “Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review,” *IEEE Access*, vol. 11, pp. 35479–35516, 2023, doi: 10.1109/ACCESS.2023.3266093.

[10] M. Hussain, “YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO,” *IEEE Access*, vol. 12, pp. 42816–42833, 2024, doi: 10.1109/ACCESS.2024.3378568.

[11] R. Phoophuangpairoj, T. Ngoenrungrueang, and S. Audomsin, “Ripeness Classification of a Bunch of Bananas Using a CNN,” in *Proceeding - 2023 International Electrical Engineering Congress (iEECON)*, IEEE, 2023, pp. 180–183, doi: 10.1109/IEECON56657.2023.10126585.

[12] D. Worasawate, P. Sakunasinha, and S. Chiangga, “Automatic Classification of the Ripeness Stage of Mango Fruit Using a Machine Learning Approach,” *AgriEngineering*, vol. 4, no. 1, pp. 32–47, Jan. 2022, doi: 10.3390/agriengineering4010003.

[13] D. F. Al Riza et al., “Mango (Mangifera indica cv. Sein Ta Lone) ripeness level prediction using color and textural features of combined reflectance-fluorescence images,” *Journal of Agriculture and Food Research*, vol. 11, p. 100477, Mar. 2023, doi: 10.1016/J.JAFR.2022.100477.

[14] R. E. Saragih and A. W. R. Emanuel, “Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network,” in *3rd 2021 East Indonesia Conference on Computer and Information Technology (EIConCIT)*, IEEE, Apr. 2021, pp. 85–89, doi: 10.1109/EIConCIT50028.2021.9431928.

[15] B. Xiao, M. Nguyen, and W. Q. Yan, “Apple ripeness identification from digital images using transformers,” *Multimedia Tools and Applications*, vol. 83, no. 3, pp. 7811–7825, Jan. 2024, doi: 10.1007/S11042-023-15938-1/TABLES/8.

[16] N. A. Prasetyo, Pranowo, and A. J. Santoso, “Automatic Detection and Calculation of Palm Oil Fresh Fruit Bunches using Faster R-CNN,” *International Journal of Applied Science and Engineering*, vol. 17, no. 2, pp. 121–134, May 2020, doi: 10.6703/IJASE.202005_17(2).121.

[17] M. Tools, B. Xiao, M. Nguyen, and W. Q. Yan, “Fruit ripeness identification using YOLOv8 model,” *Multimedia Tools and Applications*, vol. 83, no. 9, pp. 28039–28056, 2024, doi: 10.1007/s11042-023-16570-9.

[18] A. Sutar, A. Naikare, P. Jadhav, and R. Kute, “Mango and Banana Dataset (Ripe Unripe): Indian RGB Image Datasets for YOLO Object Detection,” Mendeley Data. Accessed: Jan. 08, 2024. [Online]. Available: https://data.mendeley.com/datasets/y3649cmgg6/3

[19] T. Zheng, X. Yang, J. Lv, M. Li, S. Wang, and W. Li, “An efficient mobile model for insect image classification in the field pest management,” *Engineering Science and Technology, an International Journal*, vol. 39, p. 101335, Mar. 2023, doi: 10.1016/j.jestch.2023.101335.

[20] M. Rafiei, J. Raitoharju, and A. Iosifidis, “Computer Vision on X-Ray Data in Industrial Production and Security Applications: A Comprehensive Survey,” *IEEE Access*, vol. 11, pp. 2445–2477, 2023, doi: 10.1109/ACCESS.2023.3234187.

[21] X. Zhai, Z. Huang, T. Li, H. Liu, and S. Wang, “YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection,” *Electronics*, vol. 12, no. 17, p. 3664, Aug. 2023, doi: 10.3390/electronics12173664.

[22] H. Habaragamuwa, Y. Ogawa, T. Suzuki, T. Shiigi, M. Ono, and N. Kondo, “Detecting Greenhouse Strawberries (Mature and Immature), Using Deep Convolutional Neural Network,” *Engineering in Agriculture, Environment and Food*, vol. 11, no. 3, pp. 127–138, 2018, doi: 10.1016/j.eaef.2018.03.001.

[23] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO,” Jan. 2023. Accessed: Dec. 10, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[24] Sandhya and A. Kashyap, “A Novel Method for Real-Time Object-Based Copy-Move Tampering Localization in Videos Using Fine-Tuned YOLO V8,” *Forensic Science International: Digital Investigation*, vol. 48, p. 301663, Mar. 2024, doi: 10.1016/j.fsidi.2023.301663.

[25] H. Lou et al., “DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor,” *Electronics*, vol. 12, no. 10, p. 2323, May 2023, doi: 10.3390/electronics12102323.

[26] A. Kumar, “A pile of green fruit sitting on top of grass,” Unsplash. Accessed: Mar. 10, 2024. [Online]. Available: https://unsplash.com/photos/a-pile-of-green-fruit-sitting-on-top-of-grass-enUG-sjDP14

[27] R. Rajput, “Yellow round fruits on black round bowl,” Unsplash. Accessed: Mar. 10, 2024. [Online]. Available: https://unsplash.com/photos/yellow-round-fruits-on-black-round-bowl-6J3LPpawLa8

[28] J. Olsen, “Green banana fruit on brown wooden table,” Unsplash. Accessed: Mar. 10, 2024. [Online]. Available: https://unsplash.com/photos/green-banana-fruit-on-brown-wooden-table-G6elMNRLFew