# Real-Time Autonomous Detection and Localization of Loose Fruits in Oil Palm Plantations Using YOLOv4 and RGB-D

Lee Teng Ching[1], Aqilah Baseri Huddin[1,2], Fazida Hanim Hashim[1], Mohd Faisal Ibrahim[1]

[1]Department of Electric, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia
[2]Centre for Engineering Education Research, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia
aqilah@ukm.edu.my

**Abstract.** Loose Palm Fruit (LPF) is an oil palm fruit that has ripened and fallen from its bunch, containing high oil content. Each loss of LPF affects the oil extraction rate and results in financial losses. Existing LPF collection methods are not very effective as they require human control and supervision. Conventional methods, such as mechanical and roller-type LPF collectors, are inefficient because LPF is scattered over extensive plantations. Therefore, an autonomous LPF detection system is necessary. However, image-based detection systems are often disturbed by environmental factors such as brightness and grass, and the LPF location changes with the robot and camera position. The general objective of this study is to develop an accurate and efficient image-based LPF detection algorithm. This requires an efficient detection algorithm for real-time applications based on deep learning. Additionally, accurately determining the LPF location using image depth (RGB-D) is essential. This project employs a YOLOv4 object detector with high efficiency and accuracy to achieve real-time LPF detection. The LPF location is determined through the distance between the center coordinates of the LPF bounding box and the camera using depth images and the horizontal field of view of the Intel RealSense D435i camera. This system is integrated into the Robot Operating System (ROS) to ensure usability in robots. The system achieved a Mean Accuracy (mAP@IoU 0.5) of 98.74%, an average loss of 0.124, and a detection time of 5.14ms. For LPF location determination, the difference between the algorithm's calculated locations and manual measurements is only 3.82cm for the X coordinate and 1.80cm for the Y coordinate.

**Keywords:** Loose oil palm fruit, autonomous detection, YOLO, RGB-D

## 1 Related works

The oil palm plantation sector plays a vital role in Malaysia due to its oil production, ranked second among the most significant contributors to the nation's coffers. The oil


T. Amrillah et al. (eds.), *Proceedings of the International Conference on Advanced Technology and Multidiscipline (ICATAM 2024)*, Advances in Engineering Research 245,
https://doi.org/10.2991/978-94-6463-566-9_5

fruits contain 26% more oil than fresh fruit bunches. Loose fruit is the ripening and falling palm fruit and is rarely collected. According to the survey, 11.1% of full-time and 10% of part-time smallholders neglect loose fruit collecting. Two common approaches, mechanical and roller-typed loose fruit collectors, have been introduced to facilitate loose fruit collection [1]. However, these manual approaches need a large workforce and are less efficient since loose fruit is scattered everywhere, and oil palm fields are vast.

Image-based loose fruit identification using deep learning has been introduced, such as a two-stage object detection model (Faster R-CNN) by [2] and a single-stage detection model (YOLOv3) [3]. Two-stage detection models require region proposals, while single-stage models do not. The single-stage detection approach is faster but less accurate than the two-stage model. Brightness and grass can hinder image-based loose fruit detection. Hence, the loose fruit detecting algorithm must be efficient and accurate to track the camera's movement. Table 1 compares the single-stage detection model's efficiency and accuracy. YOLOv4 achieves the highest detection speed with higher accuracy.

One key aspect that existing approaches for detecting loose fruit lack is a robust localization system. The ability to determine the precise location of loose fruit is crucial in improving the efficiency of loose fruit collection. With a reliable localization system, workers can focus on collecting loose fruit, rather than wasting time searching for it. This paper presents a solution to these challenges, an algorithm that uses RGB-D images and the YOLOv4 detection model with an IntelRealsense D435i camera to automatically recognize and locate loose fruits. By eliminating the need for manual detection, this algorithm has the potential to significantly enhance the efficiency of loose fruit collection in oil palm plantations.

TABLE 1. Comparison of the speed and accuracy of different object detector

| Detection Model | Reference | PASCAL VOC 2007 + 2012 | | COCO dataset (test-dev 2017) | |
|---|---|---|---|---|---|
| | | mAP, % | FPS | mAP, % | FPS |
| Fast R-CNN | (Girshick 2015) [4] | 68.4 | 0.5 | | |
| Faster R-CNN VGG-16 | (Ren et al. 2017) [5] | 73.2 | 7 | | |
| SSD300 | (Liu et al. 2016) [6] | 74.3 | 46 | | |
| SSD512 | (Liu et al. 2016) [6] | 76.8 | 19 | | |
| YOLOv1 | (Redmon et al. 2015) [7] | 63.4 | 45 | | |
| YOLOv2 | (Redmon & Farhadi 2016) [8] | 76.8 | 67 | | |
| YOLOv3 | (Redmon & Farhadi 2018) [9] | | | 57.4 | 60 |

| | | | |
|---|---|---|---|
| YOLOv4 | (Bochkovskiy et al. 2020) [10] | 62.8 | 96 |
| YOLOv5- | (Ge et al. 2021) [11] | 63.1 | 90.1 |

## 2 Methodology

This section outlines the methodology for loose fruit detection and localization, comprising four key steps. First, pre-processing is performed to enhance image quality through normalization, augmentation, and alignment. Next, the YOLO detection framework is employed to identify and classify loose fruit within the images, utilizing various YOLO models to determine the most effective architecture. The detection performance is then evaluated based on mean average precision (mAP), average loss, and detection time to assess the models' accuracy and efficiency. Finally, the localization algorithm calculates the precise real-world coordinates of the detected loose fruit by processing the bounding boxes and camera parameters. Together, these steps ensure a comprehensive approach to achieving accurate and efficient loose fruit detection and localization.

### 2.1 Pre-processing

The data preprocessing procedures include resizing, enhancing, labeling, and annotating; 405 loose fruit images were collected and resized to 416×416. Blurring, rotation anti-clockwise, rotation clockwise, flipping, brightness fluctuation, saturation fluctuation, noise, and shearing are used to boost the detection model's efficiency from the fewest datasets. Mixed and separate dataset augmentations were used. Mixed augmentation uses all augmentation methods on one image. Separate augmentation applies single augmentation to a single image, producing more datasets.

Figures 1 and 2 illustrate mixed and separate augmentation. Mixed augmentation yields 1164, and separate yields 11964. This dataset has been split meticulously in a 75:20:5 ratio between training, validation, and testing, ensuring a comprehensive training process. The dataset will be labeled and annotated with Roboflow and saved as text.

Figure 1. Mixed augmentation – combining rotation and noise augmentation

Figure 2. Separate augmentation – blurring

## 2.2 Loose Fruit Detection Model using YOLOv4 and YOLOv4-Tiny

Object detection is crucial in computer vision, enabling systems to identify and locate objects within an image. YOLOv4 (You Only Look Once, version 4) is one of the leading Convolutional Neural Network (CNN)-based object detectors, recognized for its high accuracy and speed in single-frame detection. It utilizes YOLOv3 as the detection head. Equations (1) and (2) are used to calculate object location, while Equations (3) and (4) determine object size, with parameters illustrated in Figure 3.

During model training and validation, YOLOv4 employs 137 pre-trained convolutional layers. YOLOv4-Tiny, a variant of YOLOv4, shares the same detection architecture but differs in the training and validation process, using only 29 pre-trained convolutional layers. YOLOv4 and YOLOv4-Tiny are trained, validated, and tested on datasets with mixed and separate augmentation for comparison.

$$b_{x\_YOLOv4} = \beta \cdot \sigma(t_x) - \frac{\beta-1}{2} + C_x \quad (1)$$

$$b_{y\_YOLOv4} = \beta \cdot \sigma(t_y) - \frac{\beta-1}{2} + C_y \quad (2)$$

$$b_{w\_YOLOv4} = \rho_w e^{t_w} \quad (3)$$

$$b_{h\_YOLOv4} = \rho_h e^{t_h} \quad (4)$$

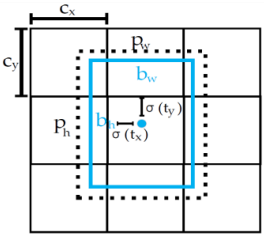


Figure 3. Target object position in image

### 2.3 Performance Evaluation for Loose Fruit Detection Model

The performance of the detection model was evaluated using several metrics: precision, recall, F1-score, mean average precision (mAP), and Intersection over Union (IoU).

Precision and recall are derived from the Confusion Matrix, with precision indicating the accuracy of positive predictions and recall measuring the model's ability to identify all relevant instances.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

The F1-score, a harmonic mean of precision and recall, provides a single metric for model accuracy based on the dataset. Mean average precision (mAP) gives an overall measure of precision across different recall levels. IoU assesses the overlap between the predicted bounding box and the ground truth, reflecting the model's localization accuracy.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (8)$$

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (9)$$

### 2.4 Loose Fruit Localization Algorithm

Accurate localization is achieved by calculating the distance between the center point of the detected bounding box and the camera/robot, utilizing the 87° horizontal field of view of the Intel RealSense D435i camera, which is determined based on the camera's position on the robot.

The center point of the detected bounding box is calculated using Equation (10).

$$(X_{LF}, Y_{LF}) = \left(\frac{X_{top}+X_{bot}}{2}, \frac{Y_{top}+Y_{bot}}{2}\right) \quad (10)$$

Since the horizontal field of view of the camera is 640 pixels, the 320th pixel serves as the reference center. This horizontal reference point is used to determine the angle between the camera/robot and the bounding box, as illustrated in Figure 4. The angle between the bounding box and the reference point is calculated using Equation (11).

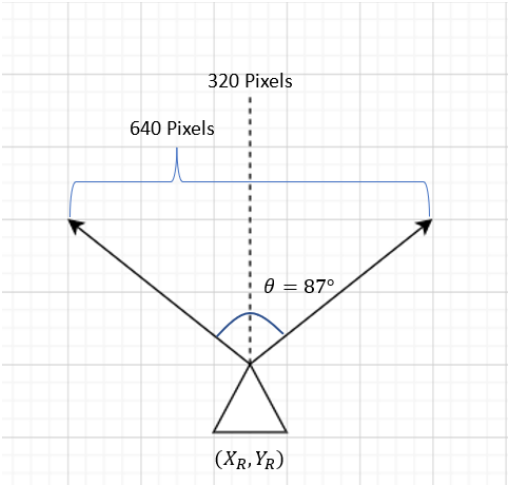


Figure 4. Reference point for localization

The distance between the reference point and the center point of the bounding box, $X_D$ and $Y_D$ can be determined by the Equation (12) and Equation (13).

$$\theta_{LF} = \begin{cases} (X_{LF} - 320) \times \left(\frac{87}{640}\right), X_{LF} \geq 320 \\ (320 - X_{LF}) \times \left(\frac{87}{640}\right), X_{LF} < 320 \end{cases} \quad (11)$$

$$X_D = D_{LF} \times \sin(\theta_{LF}) \quad (12)$$

$$Y_D = D_{LF} \times \cos(\theta_{LF}) \quad (13)$$

To determine the real-world coordinates$(X_W, Y_W)$ of the loose fruit, the distances $X_D$ and $Y_D$ are added to or subtracted from the camera/robot's location $(X_R, Y_R)$ as described in Equations (14) and (15). In this scenario, $(X_R, Y_R)$ is assumed to be (0,0) due to the stationary position of the camera. Figures 5 and 6 illustrate the overall concept of localizing the real-world coordinates of the loose fruit.

$$X_W = \begin{cases} X_R + X_D, X_{LF} \geq 320 \\ X_R - X_D, X_{LF} < 320 \end{cases} \quad (14)$$

$$Y_W = Y_R + Y_D \quad (15)$$

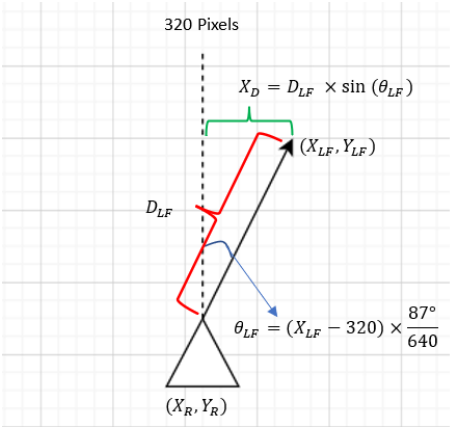


Figure 5. Concept of localizing the real-world coordinate X

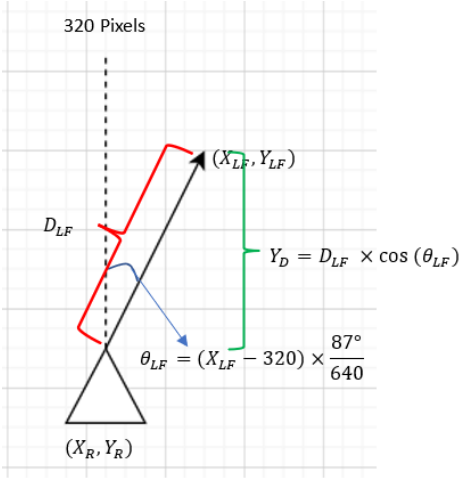


Figure 6. Concept of localizing the real-world coordinate Y

## 3 Results and Discussions

In this study, we evaluated the performance of several models for loose fruit detection and localization. The hyperparameters set for training, validation, and testing as follows:

learning rate = 0.00261, momentum = 0.9, decay = 0.0005, and batch size = 64. Four distinct models were developed: YOLOv4-Mixed Augmentation, YOLOv4-Separate Augmentation, YOLOv4-Tiny-Mixed Augmentation, and YOLOv4-Tiny-Separate Augmentation. Each model was rigorously trained and tested to assess its effectiveness.

The result shown in Figure 7, reveals that YOLOv4-Tiny-Separate Augmentation achieved the highest mean average precision (mAP) across all Intersections over Union (IoU) thresholds and had the lowest minimum average loss.

Detailed performance metrics are summarized in Table 2, including mAP@IoU0.5, true positives (TP), false positives (FP), false negatives (FN), precision, recall, F1-score, and detection time. YOLOv4-Tiny-Separate Augmentation emerged as the top performer with a mAP@IoU0.5 of 98.74%, precision of 0.97, recall of 0.96, F1-score of 0.96, and the fastest detection time of 5.140 ms. YOLOv4-Mixed Augmentation followed as the second-best model, with a mAP@IoU0.5 of 91.93%, precision of 0.64, recall of 0.96, F1-score of 0.77, and a detection time of 32.98 ms.

Based on these results, YOLOv4-Tiny-Separate Augmentation is selected for loose fruit detection and localization. Despite the theoretical advantage of YOLOv4 in terms of accuracy, it could not be fully trained due to time and memory limitations, requiring over 13 hours of training. However, since the aim of the project is to develop an algorithm suitable for implementation on a portable GPU, achieving comparable accuracy with shorter training times is crucial. YOLOv4-Tiny-Separate Augmentation offers a balance between performance and practicality, making it the optimal choice for real-world applications where efficiency and portability are key considerations.

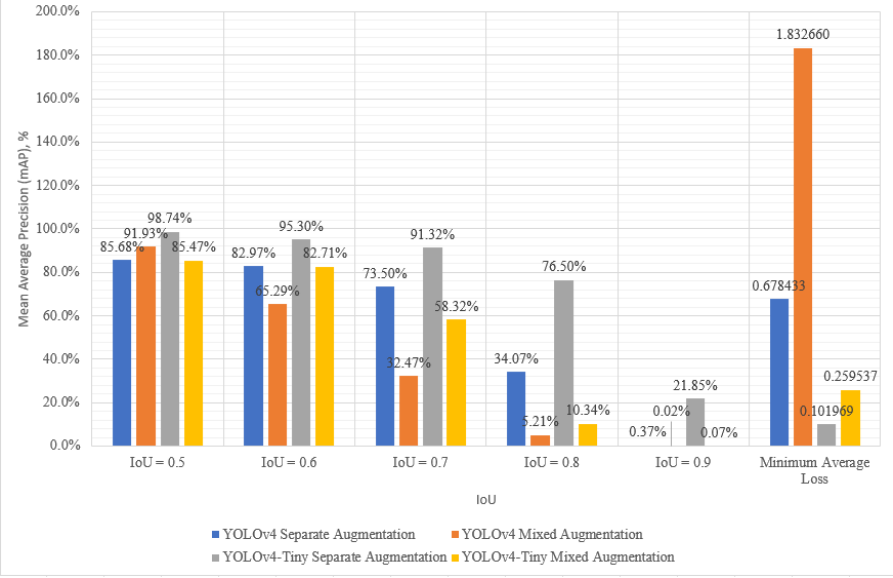


Figure 7. The mAPs for each model and their corresponding minimum average loss

TABLE 2. Performance evaluations for each detection models

| Model | mAP@IoU 0.5 | TP | FP | FN | Precision | Recall | F1-score | Detection time, ms |
|---|---|---|---|---|---|---|---|---|
| YOLOv4 Separate Augmentation | 91.93% | 107 | 60 | 5 | 0.64 | 0.96 | 0.77 | 32.98 |
| YOLOv4 Mixed Augmentation | 85.68% | 111 | 68 | 1 | 0.62 | 0.99 | 0.76 | 32.95 |
| YOLOv4-Tiny Separate Augmentation | 98.74% | 204 | 7 | 8 | 0.97 | 0.96 | 0.96 | 5.14 |
| YOLOv4-Tiny Mixed Augmentation | 85.47% | 107 | 46 | 5 | 0.70 | 0.96 | 0.81 | 5.14 |

The algorithm-determined world coordinates of loose fruit will be compared to manual measurements. Measure the distance between the loose fruit and the camera and the real-world reference point to determine loose fruit's world coordinate. First, place an object in 320 pixels in the camera's image to find the reference point. Measure the distance between the loose fruit and the reference point, $X_{D\_Measured}$. Next, $Y_{D\_Measured}$ is measured between loose fruit and camera. Using Equation (16) and Equation (17). Assuming $(X_R, Y_R) = (0,0)$. Figures 7 and 8 show how to measure real-world fruit coordinates.

$$X_W = \begin{cases} X_R + X_{D\_Measured}, LF\ on\ right\ side \\ X_R - X_{D_{Measured}}, LF\ on\ left\ side \end{cases} \quad (16)$$

$$Y_W = Y_R + Y_{D\_Measured} \quad (17)$$

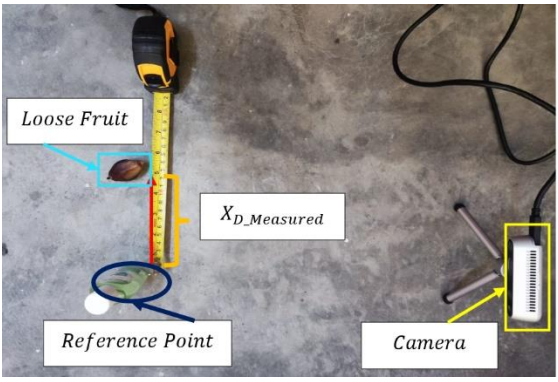


Figure 8. Determination of distance between loose fruit and reference point

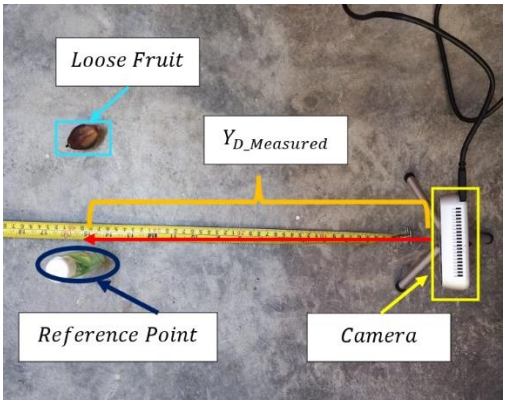


Figure 9. Determination of distance between loose fruit and camera

The performance of the localization algorithm was assessed by comparing the algorithm-determined locations with manual measurements. A total of 25 locations were used for this comparison. The results, illustrated in Figure 10, are presented as boxplots for the differences in both the X and Y coordinates. The differences in the X coordinate are within ±3.28 cm, while the differences in the Y coordinate are within ±1.80 cm. These small discrepancies indicate that the algorithm's localization accuracy is acceptable and aligns well with manual measurements.

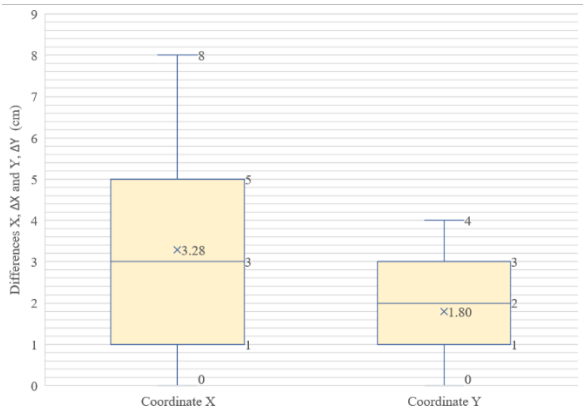


Figure 10. Boxplot for difference of coordinate X and coordinate Y.

## 4 Conclusions

In this work, we have successfully trained and evaluated four loose fruit detection models: YOLOv4-Separate Augmentation, YOLOv4-Mixed Augmentation, YOLOv4-Tiny-Separate Augmentation, and YOLOv4-Tiny-Mixed Augmentation. Among these, the YOLOv4-Tiny-Separate Augmentation has emerged as the most effective model,

achieving an impressive mAP@IoU0.5 of 98.74%, a minimum average loss of 0.102, and a detection time of 5.14 ms.

This high level of performance has led us to select this model for integration into the loose fruit detection and localization algorithm. Our comparison between the algorithm-detected locations and manual measurements has shown minimal discrepancies, with differences of only ±3.28 cm in the X coordinate and ±1.80 cm in the Y coordinate. These minor differences are well within acceptable limits, considering the potential for technical imperfections in manual measurements. Overall, the algorithm has demonstrated high accuracy, with a mAP@IoU0.5 of 98.74%, a rapid detection time of 5.14 ms, and minimal location deviation, confirming its effectiveness for loose fruit detection and localization.

For future work, several opportunities to enhance the system's performance and applicability may be addressed. First, the incorporation of additional data sources or sensor modalities, such as thermal imaging or multispectral data, could significantly improve detection capabilities in varying environmental conditions. Second, the exploration of advanced training techniques and model architectures could lead to substantial improvements in the algorithm's accuracy and robustness, especially in challenging scenarios. Third, the optimization of the algorithm for real-time processing on embedded systems could greatly enhance its versatility for on-field applications. However, it is the final opportunity that holds the most promise-conducting extensive field trials with diverse fruit types and plantation conditions. These trials provide invaluable insights into the system's real-world performance and robustness, and we believe they are a crucial step in the system's development.

**Acknowledgments.** The authors would like to acknowledge the support of Faculty of Engineering and Built Environment, UKM in providing facilities for this research. This research is supported under research university grant GUP-2022-027 Universiti Kebangsaan Malaysia (UKM).

## References

1. M. Z. Mohd Yusoff, "Loose Fruit Collector Machine in Malaysia: A Review," International Journal of Engineering Technology and Sciences, vol. 6, no. 2, pp. 65–75, 2019.
2. J. Xiang, A. B. Huddin, M. F. Ibrahim, and F. H. Hashim, "An Oil Palm Loose Fruits Image Detection System using Faster R-CNN and Jetson TX2," in IEEE, 2021, pp. 1–6.
3. M. H. Junos, A. S. Mohd Khairuddin, S. Thannirmalai, and M. Dahari, "Automatic detection of oil palm fruits from UAV images using an improved YOLO model," Visual Computer, 2021.
4. R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
5. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.
6. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," LNCS, vol. 9905, pp. 21–37, 2016.

7. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
8. J. Redmon and A. Farhadi, "YOLOv2," CVPR, vol. 2017, pp. 187–213, 2016.
9. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.
10. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.
11. Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," pp. 1–7, 2021.