International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 5183-5190
http://www.ijcesen.com

ISSN: 2149-9144

Copyright © IJCESEN

Research Article

Application of YOLO and Custom-Designed Intelligent Teaching Aids in Robotic Arm-Based Fruit Classification and Grasping Instruction

Chun-Chieh Wang[1*], Sun-Jing Yan[2]

[1]Department of Electrical Engineering, National Yunlin University of Science and Technology, 640301, Douliu, Yunlin, Taiwan

* **Corresponding Author Email:** jasonccw@yuntech.edu.tw -**ORCID:** 0000-0002-1882-3887

[2]Department of Electrical Engineering, National Yunlin University of Science and Technology, 640301, Douliu, Yunlin, Taiwan

Email: M11212100@yuntech.edu.tw - **ORCID:** 0009-0005-4231-1389

Article Info:

DOI: 10.22399/ijcesen.3319
Received : 20 May 2025
Accepted : 10 July 2025

Keywords

Fruit Recognition
Robotic Arm
YOLOv4
Deep Learning

Abstract:

With the growing impact of deep learning and computer vision, real-time image recognition and robotic systems have become increasingly important in fields such as autonomous vehicles and smart devices. In this study, a practical teaching module was developed by integrating the YOLO (You Only Look Once) algorithm, robotic arm control, and local crop recognition. The proposed system enables automated fruit detection, classification, and sorting using a six-axis robotic arm. This hands-on approach allows students to apply artificial intelligence in real-world agricultural contexts, thereby enhancing their understanding of smart farming technologies. The module supports both theoretical learning and skill development in automation and intelligent systems, aligning with future trends in AI-based agriculture and industrial applications.

1. Introduction

1.1 Research Motivation and Objectives

In recent years, machine vision and robotics technologies have experienced significant advancements, particularly driven by developments in deep learning and object detection algorithms. These technologies have opened new avenues for intelligent automation across various industries. Robotic arms, as a fundamental component of such systems, enable precise manipulation and execution of tasks, making them indispensable tools in modern automated applications. Consequently, equipping students with relevant knowledge and practical skills in this field has become increasingly important.

To address this need, an auxiliary teaching system has been developed that integrates image recognition, robotic arm control, and local agricultural applications. Central to this system is the YOLO (You Only Look Once) algorithm, which has gained widespread attention in computer vision due to its high-speed processing and accurate detection capabilities. YOLO's end-to-end architecture allows for direct mapping of input images to prediction outputs, simplifying the training pipeline. Moreover, its single-stage regression-based approach, combined with multi-scale feature processing, enables efficient and robust object detection in real-time scenarios.

Building on these strengths, the proposed system combines YOLO with a six-axis robotic arm and local fruit datasets to create an automated fruit recognition and sorting platform. This setup allows the robotic arm to autonomously identify and grasp various types of fruits. Through hands-on engagement with this system, students are exposed to practical applications of artificial intelligence and robotics in agriculture, thereby reinforcing their technical understanding.

Furthermore, this project encourages the development of problem-solving abilities, innovative thinking, and interdisciplinary collaboration. By integrating theoretical knowledge with practical experience, the system provides a comprehensive learning environment that enhances students' readiness for future careers in automation, smart agriculture, and intelligent systems.

1.2 Literature Review and Related Research

The YOLO (You Only Look Once) algorithm, first introduced by Joseph Redmon et al. in 2016, revolutionized real-time object detection with its single-stage detection framework. Since the release of YOLOv1, subsequent versions—YOLOv2 and YOLOv3—have progressively enhanced detection accuracy and speed. With continued research contributions, YOLOv4 was introduced by Alexey Bochkovskiy and his team in April 2020, further advancing the network's capabilities in terms of feature representation and generalization.

In recent years, the YOLO family of algorithms has demonstrated performance on par with two-stage detectors, offering a desirable balance between accuracy and computational efficiency. For instance, in [5], a lightweight traffic sign recognition algorithm based on YOLOv4-Tiny was proposed. By incorporating an improved K-means clustering approach, the model achieved a 5.73% increase in mean average precision (mAP) and a 7.29% improvement in recall, while maintaining a real-time detection speed of approximately 87 frames per second (FPS).

Further applications of YOLO include precision forestry, as demonstrated in [6], where a pre-trained YOLO model was used to identify pine trees affected by disease in drone-captured ultra-high-resolution imagery, achieving an average precision of 91.82%. In [7], two lightweight improvements—YOLOv8PANet and YOLOv8CCFPANet—were developed, reducing the number of parameters by 21% and 54.5%, respectively, while retaining detection accuracies of 84.2% and 83.3% on the PASCAL VOC07+12 dataset. Detection speeds reached 100 FPS and 108 FPS, respectively, illustrating their suitability for real-time applications.

Moreover, a general-purpose detection model optimized for aerial systems was presented in [8], attaining a detection speed of 50 FPS, with a mAP@50 of 99.1% and mAP@50–95 of 83.5%. These results confirm YOLO's robustness and adaptability across diverse domains.

Among the existing YOLO variants, YOLOv4 and YOLOv7 stand out due to their superior accuracy and processing speed. YOLOv4 incorporates improvements in feature extraction modules, training strategies, and data augmentation, while YOLOv7 leverages multi-level feature fusion and compression techniques to optimize both accuracy and inference efficiency.

Considering their demonstrated effectiveness and computational advantages, YOLOv4 and YOLOv7 were selected as the primary object detection frameworks in the present study.

2. Hardware Architecture

2.1. Experimental Environment

The experimental system designed in this study consists of a robotic arm, a camera module, a fruit recognition zone, and six designated fruit placement areas, as illustrated in Fig. 1 and Fig. 2. Initially, a fruit sample is positioned within the recognition area, where an image is captured by the camera. The captured image is processed using the YOLOv4 object detection algorithm to identify the fruit type. Upon successful recognition, a command is transmitted via serial communication to the robotic arm, instructing it to move to the corresponding target location. The robotic arm is equipped with an air compressor, which provides pneumatic pressure to the gripper mechanism, allowing the arm to securely grasp the fruit and relocate it to the appropriate placement zone.

Figure 1. *Schematic diagram of the experimental environment*

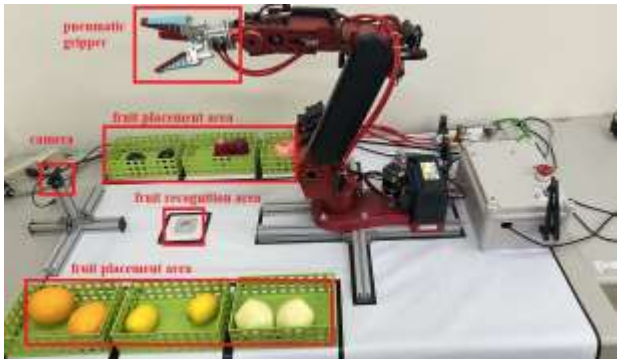

Figure 1. *Actual photo of teaching aids*

The hardware configuration of the system includes an NVIDIA Jetson Nano, which serves as the core AI processing unit. The Jetson Nano is a compact, low-power embedded computing platform developed by NVIDIA, offering substantial AI computational power suitable for real-time applications in constrained environments.

An Arduino microcontroller is utilized to manage the motion control logic of the robotic arm. It receives serial commands from the Jetson Nano and executes corresponding movements. The vision system employs a 1080P, 60 FPS wide-angle camera with a 120-degree field of view. The camera is positioned approximately 30 cm from the object and 15 cm above ground level, with a downward tilt angle of approximately 30 degrees to optimize object detection performance.

2.2. Dynamics of the Robotic Arm

2.2.1. D-H method [9]

The D-H method is a mathematical approach used to establish the kinematic model of a robotic arm, proposed by Jacques Denavit and Richard S. Harten-berg in 1955. Initially, the relative position between two joints required six parameters for representation, consisting of three translational and three rota-tional components. However, the D-H method simplifies this by describing the spatial relationship between two joints using only four parameters.

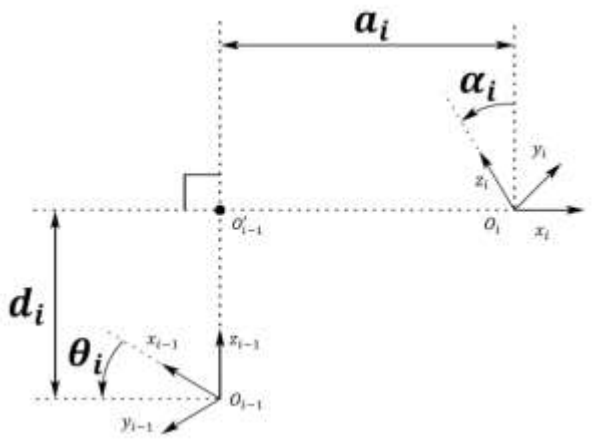

Figure 3. *Parameter explanation diagram for the D-H method*

The four parameters in the D-H method are a_i, α_i d_i and θ_i,where i denotes the i-th joint of the robotic arm. The definitions of these parameters are as follows, as shown in Fig. 3:

- a_i is the distance between point O_i *and* O'_{i-1}.
- α_i is the angle of rotation from z_{i-1} to z_i, with counterclockwise rotation around x_i being positive.
- d_i is the distance between point O_{i-1} and O'_{i-1}'.
- θ_i is the angle of rotation from x_{i-1} to x_i with counterclockwise rotation around z_{i-1} being positive.

Using the D-H method, the transformation relationship from the joint coordinates of the i-th axis to the joint coordinates of the $(i+1)$-th axis can be represented by T_i^{i-1}. Here, T_i^{i-1} denotes the transformation matrix that converts the coordinates from the (i-1)-th axis to the i-th axis, as shown in equation 1.

$$T_i^{i-1} = Rot_{z_{i-1},\theta_i} Trans_{z_{i-1},d_i} Trans_{x_i,a_i} Rot_{x_i,a_i} = \begin{bmatrix} cos\theta_i & -cos\alpha_i sin\theta_i & sin\theta_i sin\alpha_i & \alpha_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & \alpha_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Once the transformation matrices for each axis relative to the previous axis have been obtained, these matrices can be multiplied together to yield the transformation matrix from the base to the end effector.

3. Research Methods

3.1. YOLOv4

As shown in Fig. 4 and Fig. 5, the YOLOv4 framework can be broadly divided into the following components:

• Input: The input image.

• Backbone: The backbone network is utilized for preliminary feature extraction. YOLOv4 employs CSPDarknet53.

• Neck: This component integrates feature maps from various layers of the backbone, utilizing Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN).

• Head: This part makes predictions based on the image features, generating predicted bounding boxes and class predictions, using the head architecture from YOLOv3.

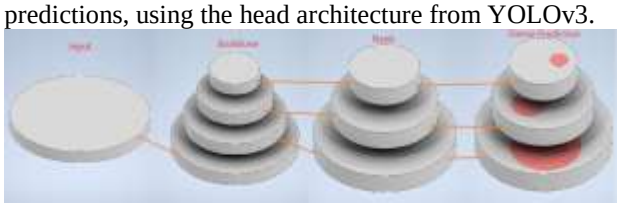

Figure 4. *YOLOv4 development framework*

The YOLOv4 development framework is illustrated in Figure 5, providing an overview of the system's overall architecture and detailing how each component contributes to the object detection process.

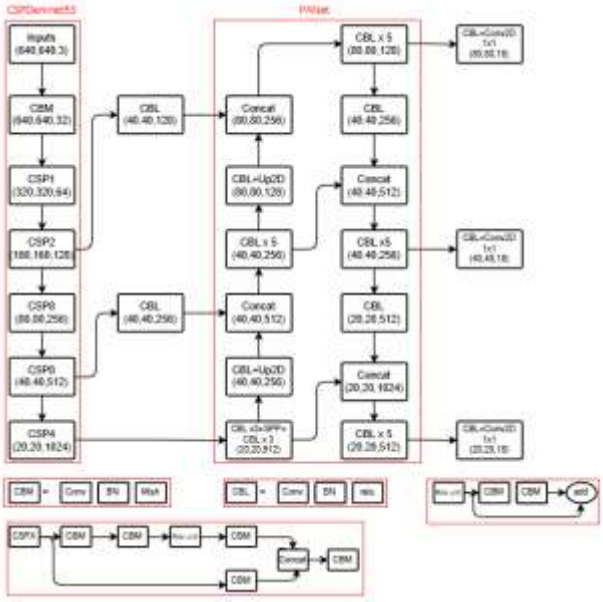

Figure 5. *YOLOv4 architecture*

3.2 YOLOv7[10]

YOLOv7, introduced by WongKinYiu et al. in 2022, represents a significant advancement in the YOLO family of real-time object detection algorithms. It offers an improved balance between detection accuracy and computational speed, making it particularly suitable for real-time applications. Inheriting the fast inference capabilities characteristic of previous YOLO versions, YOLOv7 integrates several architectural and training enhancements that further boost both efficiency and accuracy.

Architecturally, YOLOv7 adopts a modular structure combined with re-parameterization techniques. This dual-mode design enables the model to utilize a more complex architecture during the training phase enhancing its feature extraction and learning capacity while simplifying the network during inference to ensure low-latency, high-speed performance (Fig. 6). This separation between training and inference structures optimizes the trade-off between learning ability and runtime efficiency.

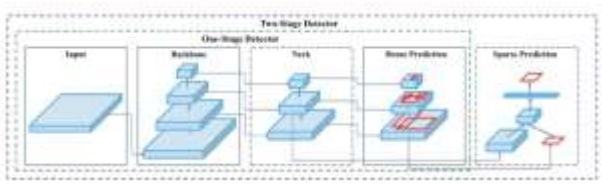

Figure 6. *YOLOv7 Architecture*

In addition, YOLOv7 introduces innovative training strategies such as the coarse-to-fine head design and task alignment optimization. These methods improve the integration of features across multiple learning objectives, including object classification, bounding box regression, and object localization, thereby enhancing overall model performance in multi-task scenarios.

Experimental evaluations on benchmark datasets such as COCO and PASCAL VOC demonstrate that YOLOv7 achieves superior results in both speed and accuracy compared to previous YOLO versions, including YOLOv5 and YOLOv6. In particular, YOLOv7-tiny delivers high frame rates (FPS) while maintaining a competitive mean Average Precision (mAP), rendering it ideal for edge devices and embedded systems requiring real-time object detection capabilities.

In summary, YOLOv7 signifies a mature and efficient solution in the domain of real-time object detection. Its design innovations and empirical performance make it well-suited for diverse applications, including intelligent surveillance, autonomous driving, and machine vision systems.

3.3 Dataset [11]

In this study, the PASCAL VOC dataset format was adopted to construct the training architecture, as illustrated in Fig. 6. The PASCAL VOC dataset is widely used in object detection tasks and contains annotations for 20 object categories, including people, vehicles, and animals. Each image is accompanied by an XML annotation file that specifies the object class, position, and bounding box dimensions. The dataset is typically partitioned into training, validation, and testing subsets to facilitate model development and performance evaluation.

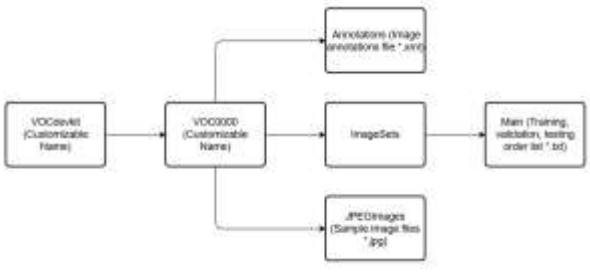

Figure 6. *Training architecture based on VOC annotation structure used in this study*

To align with the objectives of this research—namely, intelligent recognition of common fruits for educational and agricultural applications a custom dataset was constructed focusing on six fruit categories. The classification criteria were based on physical characteristics, nutritional value, and common processing methods. The selected categories are:

Pear: High in water content and dietary fiber, pears (e.g., water pears, duck pears, Fengshui pears) are consumed fresh or processed into juice and dried products.

Lemon: Known for their strong acidity and high vitamin C content, lemons are widely used in juice, flavoring, and the production of essential oils from the peel.

Orange: Juicy and moderately sweet, oranges (e.g., navel, tangerines, Ponkan) are rich in flavonoids and antioxidants. Their peels are commonly used in traditional medicine and food seasoning.

Apple: With varieties such as Fuji and Granny Smith, apples are rich in polyphenols and pectin. They can be eaten fresh or used in processed products like juice, jam, and baked goods.

Wax Apple: Bell-shaped with high water content, wax apples grow in tropical regions and are rich in vitamin C and potassium. They are often consumed fresh or used in dried and candied forms.

Mangosteen: Referred to as the "queen of fruits," mangosteen features a thick rind and sweet, antioxidant-rich pulp. It is typically consumed fresh but may also be processed into juices and desserts.

The dataset developed for this experiment includes a total of 198 labeled images. A stratified sampling approach was employed, with 70% of the data allocated to both training and testing subsets. This dataset not only enables robust training of the object detection model but also provides students with valuable insights into the nutritional and commercial significance of these fruits, thereby supporting smart agricultural applications.

Table 1. *Distribution and Quantity of Training Samples*

Category	Quantity
Pear	38
Lemon	30
Orange	34
Apple	37
Wax Apple	34
Mangosteen	25

3.4 Model training

Prior to initiating the training of a YOLO-based object detection model, it is imperative to conduct a comprehensive data preparation process to facilitate supervised learning. This process involves the annotation of sample images using dedicated tools such as LabelImg or Roboflow. These tools enable the precise labeling of target objects by specifying their class categories, spatial locations (bounding boxes), and respective dimensions. Each annotated image is associated with a corresponding text file in the YOLO format, which includes critical information such as the class index, the normalized coordinates of the bounding box center, and its width and height. These annotations serve as essential input data for model training.

To ensure robust model generalization and mitigate overfitting, the annotated dataset is typically partitioned into three subsets: the training set, validation set, and test set. Conventionally, 70–80% of the dataset is allocated to

training, 10–20% to validation for hyperparameter tuning and performance monitoring, and the remaining 10% is reserved for final testing and accuracy evaluation. This strategic division of data supports consistent model evaluation across varying environmental conditions and object distributions.

Following the completion of data annotation and partitioning, the YOLO training environment must be appropriately configured. This step involves modifying configuration files such as yolov4-tiny.cfg, which are often renamed (e.g., obj.cfg) to reflect specific application contexts. Critical parameters within the configuration file must be updated accordingly: the classes parameter should correspond to the number of object categories in the custom dataset (e.g., 6 for six fruit types). Furthermore, the filters parameter in the final convolutional layer must be adjusted using the formula: filters = 3 × (classes + 5)

This adjustment ensures that the detection layer generates outputs compatible with the updated number of classes, thereby aligning the architecture with the customized detection task shown in Fig. 7. Proper configuration of these parameters is vital to the success of the training process and the resulting model accuracy.

```
stride=1
pad=1
filters=33
activation=linear


[yolo]
mask = 3,4,5
anchors = 10,14,  23,27,  37,58,  81,82,  13
classes=6
```

***Figure 7**. Parameter Modification for Training*

The training phase requires careful tuning of multiple hyperparameters to ensure optimal model performance. In the present study, the batch size was configured to 32, and the learning rate was set at 0.001. These values were selected to promote stable convergence during backpropagation. Additionally, momentum and weight decay were employed to enhance the effectiveness of gradient descent optimization. Momentum assists in accelerating convergence by dampening oscillations, while weight decay acts as a regularization mechanism to prevent overfitting, thereby improving the generalization ability of the model.

To further expedite the training process and enhance model accuracy, transfer learning was applied through the use of pre-trained weights. These weights, typically derived from training on large-scale datasets such as ImageNet, provide the model with a strong initialization point. By leveraging pre-learned feature representations, the training process becomes significantly faster and more effective. The pre-trained weights for the YOLOv4-tiny model were obtained from the official YOLO repository, as illustrated in Fig. 8.

```
yolov4-tiny.conv.29
```

***Figure 8.** Pre-trained Model of YOLOv4*

3.4 Training Results

To evaluate model performance, a comparative experiment was conducted using both YOLOv4 and YOLOv7 frameworks on the fruit recognition dataset. The training outcomes are visualized in Fig. 9 and 10, respectively.

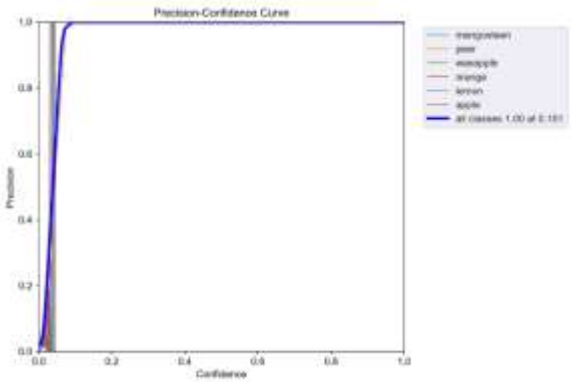

***Figure 9.** Training Results of YOLOv4*

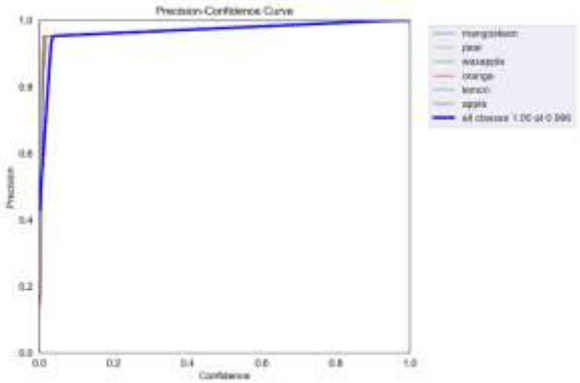

***Figure 10.** Training Results of YOLOv7*

An analysis of the Precision–Confidence (P–C) curves reveals that the YOLOv7 model demonstrates superior classification performance. As shown in Figure 10, YOLOv7 maintains a high precision level even at a near-maximum confidence threshold of 0.996. The curve exhibits smoothness and stability, indicating that the model consistently produces reliable predictions at higher confidence levels. This observation confirms the model's robustness and high accuracy in real-world detection tasks. In contrast, although YOLOv4 (Figure 9) achieves a precision score of 1.00, its optimal performance is concentrated at a significantly lower confidence threshold of 0.101. This implies that the model tends to generate detections even when confidence levels are low, potentially increasing recall at the cost of a higher false-positive rate. The comparative results underscore YOLOv7's advantage in maintaining a stronger alignment between prediction confidence and precision, making it a

more appropriate choice for deployment in the proposed fruit classification system.

3.5 Coordinate Transformation

After completing the object detection process, the YOLO model outputs a series of bounding box coordinates and associated information relative to the image size. These include the relative position of the bounding box center, its width and height, and a confidence score. Since these coordinates are expressed as normalized ratios, they must be converted into pixel coordinates. By multi-plying the relative coordinates by the image width and height, the actual pixel values for the bounding box center, width, and height can be obtained. These pixel coordinates form the basis for subsequent spatial coordinate calculations.

Next, using the camera intrinsic matrix K and the depth value Z, the pixel coordinates are transformed into three-dimensional coordinates in the camera coordinate system. The intrinsic matrix K defines the optical characteristics of the camera, including focal lengths and the principal point. This transformation is achieved using Equation 2:

$$\begin{bmatrix} X_{camera} \\ Y_{camera} \\ Z_{camera} \end{bmatrix} = Z \cdot K^{-1} \cdot \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \qquad (2)$$

The structure of the camera intrinsic matrix K is shown in Equation 3 :

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

Through this transformation, the center of each bounding box can be mapped to 3D coordinates in the camera coordinate system. The accuracy of this step heavily relies on the precision of the depth value Z, as it directly affects the spatial localization accuracy of the target.

Finally, in order to allow the robotic arm to perform precise positioning, the 3D coordinates in the camera coordinate system must be transformed into the world coordinate system. This is done using a rotation matrix R and a translation vector T, as expressed in Equation 4:

$$P_{world} = R \cdot P_{camera} + T \qquad (4)$$

where:

$$P_{camera} = \begin{bmatrix} \mathrm{X_{camera}} \\ \mathrm{Y_{camera}} \\ \mathrm{Z_{camera}} \end{bmatrix} \qquad (5)$$

Here, P_world represents the 3D coordinates of the object in the world coordinate system. The rotation matrix R describes the orientation of the camera relative to the world frame, while the translation vector T defines the camera's position within the world coordinate system.

After completing these transformations, the precise position of the object in the world coordinate system is obtained and can be passed to the robotic arm for defect repair or other operational tasks. This process ensures an accurate and efficient pipeline from detection to localization, meeting the requirements of practical applications.

4. Experiment Research and Results

4.1 Experiment Process

Fig. 11 illustrates the operational workflow of the experiment. As shown in Fig. 11(a), the robotic arm begins in its initial standby position. Next, a fruit is placed within the recognition area, as depicted in Fig. 11(b). After the YOLO model identifies the target, the robotic arm proceeds to grasp the object, as shown in Fig. 11(c). Finally, the fruit is relocated and placed back in its original position, completing a full operation cycle, as illustrated in Fig. 11(d).

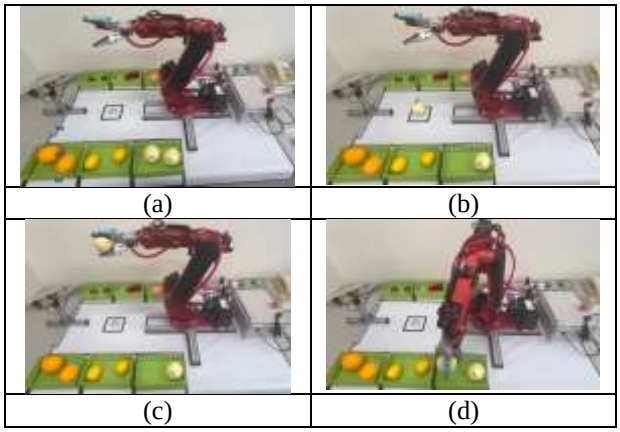

Figure 11. *Experimental steps (Using pear as an example)*

In this experiment, each type of fruit was placed in six different orientations, as shown in Fig. 12. The robotic arm was then tasked with returning each fruit to its original position after detection. This process was repeated for every type of fruit, and the confidence scores corresponding to each orientation were recorded. Additionally, the success rate was calculated based on 50 trials per fruit type and orientation, evaluating how effectively the robotic arm could reposition the fruit accurately in each scenario.

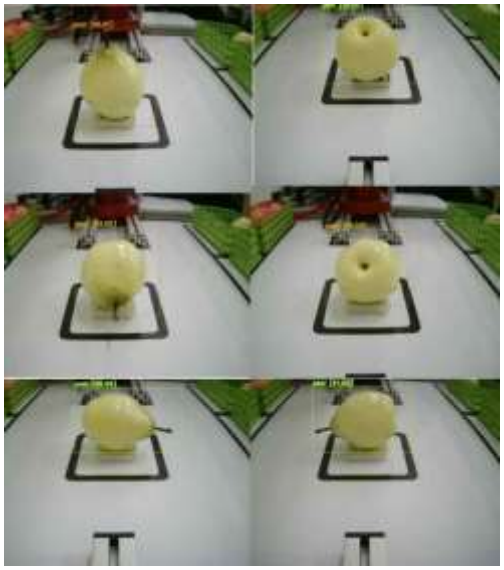

Figure 12. *Six-angle fruit placement example (pear)*

4.2 Experiment Results

Table 2. *Success Rate of Fruit Placement (YOLOv7)*

Fruit Type	Pear	Lemon	Orange	Apple	Wax Apple	Mangosteen
Successful Cases	50	48	49	47	48	43
Success Rate (%)	100 %	96 %	98 %	94 %	86 %	86 %

Table 3. *Success Rate of Fruit Placement (YOLOv4)*

Fruit Type	Pear	Lemon	Orange	Apple	Wax Apple	Mangosteen
Successful Cases	44	46	48	45	44	39
Success Rate (%)	88 %	92 %	96 %	90 %	88 %	78 %

In the conducted experiment, the YOLOv7 model demonstrated near-perfect recognition and placement performance for certain fruits, particularly pears and lemons. Even in more challenging cases such as mangosteen, YOLOv7 surpassed the performance of YOLOv4 by approximately 8%, highlighting its enhanced robustness and stability under varied environmental and object conditions.

A comprehensive performance analysis confirms that YOLOv7 possesses superior feature extraction and spatial localization capabilities. These characteristics significantly improve the reliability of fruit recognition and increase the success rate of automated placement tasks. As a result, YOLOv7 is deemed a more suitable candidate for deployment in real-world intelligent fruit sorting systems and automated agricultural applications.

Despite these improvements, it was observed that both YOLOv4 and YOLOv7 exhibited relatively lower recognition accuracy for mangosteen. Several contributing factors were identified. First, the rear view of mangosteen lacks distinctive visual features such as prominent texture or high-contrast coloration, which limits the model's ability to distinguish the fruit from the background. Second, the training dataset included insufficient samples of mangosteen from non-frontal perspectives, thus impeding the model' s capacity to generalize across different orientations. Additionally, external variables such as inconsistent lighting, shadow presence, and oblique camera angles may have adversely affected the recognition outcomes.

To address these limitations, the following enhancement strategies are proposed:

Dataset Expansion: Increase the volume and diversity of mangosteen images, particularly from rear-facing angles and under various environmental conditions, to improve model generalization.

Data Augmentation: Apply techniques such as rotation, horizontal/vertical flipping, and brightness/contrast modulation to artificially enrich the training dataset and allow the model to learn invariant features.

Training Optimization: Modify the loss function weights to emphasize difficult-to-classify or low-confidence samples. Additionally, implement multi-view fusion techniques, where multiple frames from different angles are analyzed to yield more consistent recognition results.

Hardware Adjustments: Alter the camera installation angle to capture a more comprehensive view of the fruit, reducing the likelihood of occlusion or blind spots. Enhancing the illumination in the recognition area may also reduce shadow interference and improve the visibility of key fruit features.

By integrating these strategies, the recognition performance for complex scenarios—particularly involving irregular fruit orientations—can be significantly improved. This will contribute to the development of a more adaptive, robust, and accurate vision-based system, laying a stronger technological foundation for future applications in smart agriculture, automated harvesting, and intelligent robotic manipulation.

5. Conclusion and Future Prospects

Based on the experimental results, the proposed system demonstrated high accuracy in recognizing and classifying a variety of fruit types under most test conditions. However, a notable performance issue was observed when the backside of the mangosteen was oriented toward the robotic arm. In such cases, the confidence scores of the predicted bounding boxes dropped significantly, leading to unsuccessful grasping and placement operations. This result indicates that the system's recognition accuracy can degrade when confronted with challenging visual angles or indistinct object features, thereby revealing certain limitations in the integration of computer vision algorithms with robotic manipulation mechanisms in real-world applications.

This phenomenon underscores the need for a deeper exploration into the underlying factors affecting detection performance. Specifically, the influences of illumination variability, camera viewpoints, and the surface geometry or texture characteristics of the fruit merit further investigation. Addressing these limitations is essential to improving the robustness and reliability of vision-guided robotic systems.

Future work will focus on a multifaceted approach to mitigate these challenges. Several strategies are proposed:

Dataset Enhancement: Increase the diversity of the training dataset by incorporating fruit images captured from multiple orientations, under various lighting conditions, and including non-frontal views of fruits like mangosteen. This would improve the model's generalization ability and detection consistency.

Algorithm Optimization Fine-tune the YOLO model by adjusting hyperparameters or experimenting with advanced network architectures (e.g., transformer-based detectors or attention mechanisms) to enhance feature extraction from visually ambiguous regions.

Systematic Analysis: Conduct controlled experiments to quantitatively assess the impact of angle, lighting, and fruit surface variation on prediction accuracy.

Robotic Arm Improvements: Refine the control strategy of the robotic manipulator to enable better adaptability when interacting with fruits of varied shapes, softness, or irregular structures. Incorporating sensor fusion or feedback mechanisms may also contribute to enhanced grasping success rates.

In addition to addressing the current limitations, future research will expand the application scope of the system to accommodate a wider range of fruit types beyond commonly encountered varieties. The recognition of fruits with irregular shapes, soft textures, or non-uniform surfaces will be explored to evaluate how these features affect detection and manipulation performance.

Furthermore, improving the overall automation capability and flexibility of the system is crucial. By integrating the aforementioned enhancements, the system is expected to demonstrate improved performance in dynamic environments, thereby supporting broader applications in smart agriculture, intelligent harvesting, and automated sorting systems. These advancements not only contribute to a more comprehensive understanding of practical deep learning deployments but also offer robust technical support for the development of next-generation intelligent systems in industry and agriculture.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

[2]Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263–7271.

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

[4]Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint* arXiv:2004.10934. https://arxiv.org/abs/2004.10934

[5] Wang, L., Zhou, K., Chu, A., Wang, G., & Wang, L. (2021). An improved light-weight traffic sign recognition algorithm based on YOLOv4-tiny. *IEEE Access, 9*, 124963–124971. https://doi.org/10.1109/ACCESS.2021.3109882

[6] Safonova, A., Hamad, Y., Alekhina, A., & Kaplun, D. (2022). Detection of Norway spruce trees (Picea abies) infested by bark beetle in UAV images using YOLOs architectures. *IEEE Access, 10*, 10384–10392. https://doi.org/10.1109/ACCESS.2022.3143981

[7] Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-time flying object detection with YOLOv8. *arXiv preprint* arXiv:2305.09972. https://arxiv.org/abs/2305.09972

[8] Zhong, J., Qian, H., Wang, H., Wang, W., & Zhou, Y. (2024). Improved real-time object detection method based on YOLOv8: A refined approach. *Journal of Real-Time Image Processing, 22*(4). https://doi.org/10.1007/s11554-024-01358-4

[9] Wang, Y.-S. (2020). *Adaptive inverse dynamics motion control with image-based visual servoing for UR5 manipulator* (Master's thesis, National Dong Hwa University, Taiwan).

[10] Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464–7475.

[11] Chiu, F.-W. (2024). *Comparison of detection results of bird's nests on transmission line towers by YOLOv3, YOLOv4 and YOLOv5* (Master's thesis, National Changhua University of Education, Taiwan).