Sensors and Materials, Vol. 38, No. 2 (2026) 785–800
MYU Tokyo

S & M 4348

Design and Implementation of a You Only Look Once Version 7 with Adaptive Spatial Feature Fusion-based System for Real-time Fruit Detection and Grasping

Chun-Chieh Wang,* Chung-Wen Hung, Sun-Jing Yan, and Chian-Cheng Ho

Department of Electrical Engineering, National Yunlin University of Science and Technology,
123 University Road, Section 3, Douliu City, Yunlin 640301, Taiwan

(Received July 23, 2025; accepted September 2, 2025)

Keywords: fruit recognition, robotic arm, YOLO, deep learning

With the continuous development of deep learning and computer vision technologies, the integration of image recognition with robotic manipulation has become an important topic in smart manufacturing and agricultural automation. In this study, we propose an interactive teaching platform for fruit classification and autonomous grasping, which incorporates a vision sensor (camera) with the You Only Look Once Version 7 (YOLOv7) object detection algorithm enhanced by the adaptively spatial feature fusion (ASFF) module. The camera serves as the primary sensing component, providing a real-time visual input that is processed by the detection algorithm to enable robust fruit recognition and precise grasping actions. The system integrates advanced visual recognition, adaptive grasping strategies, and a user-friendly human–machine interface, creating a practical learning environment for hands-on experience in intelligent system applications. Experimental results indicate that the proposed YOLOv7-ASFF model, trained and validated on a self-constructed fruit image dataset, achieved a mean average precision (mAP) of 94.6%, while physical grasping experiments attained a success rate of 93%. These findings confirm the effectiveness of the sensor–algorithm integration and demonstrate the robustness and practical feasibility of the proposed system.

1. Introduction

1.1 Research motivation and objectives

The development of automation systems in agriculture has increasingly focused on intelligent fruit detection and robotic grasping, as these technologies play a crucial role in addressing labor shortages, enhancing productivity, and ensuring product quality. Traditional image processing techniques, such as color and shape-based recognition, have been applied to fruit detection, yet they are often sensitive to variations in illumination, occlusion, and complex background conditions. More recently, deep-learning-based approaches have demonstrated superior performance in object detection; however, their application to real-time fruit grasping remains

*Corresponding author: e-mail: jasonccw@yuntech.edu.tw
https://doi.org/10.18494/SAM5854

ISSN 0914-4935
https://myukk.org/

limited owing to challenges in detection robustness, coordinate transformation, and the dynamic nature of unstructured environments.

To overcome these limitations, we present an interactive teaching platform that integrates real-time fruit detection and robotic grasping. The proposed system employs the You Only Look Once Version 7 (YOLOv7) object detection algorithm enhanced with the adaptively spatial feature fusion (ASFF) module, which improves detection robustness by refining multiscale feature representations. Real-time image data are processed to identify fruit categories and positions, which are subsequently transformed into world coordinates to guide the robotic arm in performing accurate grasping and sorting operations. In contrast to conventional static training tools or simulation-only systems, the platform emphasizes physical interaction, modular hardware design, and continuous visual feedback, thereby supporting hands-on learning in object detection, coordinate transformation, and robotic control.

1.2 Literature review and related research

In recent years, YOLO-based object detection algorithms have demonstrated strong potential in real-time applications across agriculture, transportation, and remote sensing. To address challenges such as occlusion, scale variation, and dense object distribution, researchers have proposed enhancements to the YOLO architecture by integrating modules such as ASFF and Transformer-based components. Li *et al.*[1] introduced YOLOv5-ASFF for strawberry detection, where the ASFF module was used to improve multiscale feature representation. The model achieved a mAP of 91.86% and an F1 score of 88.03%, outperforming SSD, YOLOv3, YOLOv4, and YOLOv5s in complex field environments. Similarly, Liu *et al.*[2] proposed ASFF-YOLOv5 for multiscale traffic element detection by incorporating K-means++ clustering and SPPF modules, which improved the detection of small and overlapping objects, achieving a mAP of 93.1%.

Several studies have focused on improving detection robustness under complex outdoor conditions. YOLOv7-PSAFP was proposed for pest and disease identification by introducing a progressive spatial adaptive feature pyramid and combining varifocal loss with loss rank mining to suppress noise from negative samples.[3] The model achieved mAP values of 84.7 and 93.3% on two datasets, exceeding the baseline YOLOv7. In the maritime domain, Liu *et al.*[4] proposed YOLOv5s-SwinDS, in which the backbone was replaced with a Swin Transformer, introduced deformable convolution (DCNv2), and adopted SIoU loss to improve the detection of irregular targets. Experimental results showed superior performance over YOLOv5s, YOLOv7, and YOLOv8 on the SeaDronesSee dataset.

In addition, Transformer-based improvements have been applied in SAR ship detection and fruit recognition. The ST-YOLOA model described in Ref. 5 incorporated a Swin Transformer and coordinate attention into the STCNet backbone, and employed a residual PANet and a novel sampling strategy to enhance localization accuracy. It achieved an accuracy of up to 97.37% and maintained a real-time speed of 21.4 FPS. For fruit detection, Liu *et al.*[6] proposed YOLO-SwinTF by integrating a Swin Transformer and a Trident Pyramid Network (TPN) into YOLOv7, and introducing the Focaler-IoU loss. The model reached an AP of 98.67% and showed

high robustness under various lighting and occlusion conditions, offering improved performance over the original YOLOv7.

Those studies demonstrated the effectiveness of integrating modules such as ASFF and Transformer structures into YOLO-based models to enhance feature fusion, detection accuracy, and robustness under complex real-world conditions. Building on those advancements, in this study, we develop a YOLO-based fruit recognition framework enhanced with the ASFF module, specifically designed to address scale variation, occlusion, and dense fruit distribution in orchard environments. By emphasizing real-time detection precision and stability, the proposed approach provides a practical solution for reliable fruit recognition and lays a foundation for further applications in intelligent agricultural systems.

2. Research Methods

2.1 YOLOv4[7]

The YOLOv4 object detection framework is composed of four major components: the input module, backbone, neck, and detection head. The input module is responsible for receiving raw image data to be processed by the network. CSPDarknet53 is employed as the backbone network, which performs initial feature extraction from the input images. This backbone architecture enhances learning capability and reduces computational cost through the use of cross-stage partial connections. Following the backbone, the neck component integrates feature maps from different levels using a combination of Spatial Pyramid Pooling (SPP) and the Path Aggregation Network (PAN). These modules effectively enhance the receptive field and facilitate multiscale feature fusion, which are essential for detecting objects of various sizes.

The final component of YOLOv4 is the detection head, for which the structure used in YOLOv3 was adopted. It takes the refined feature maps produced by the neck and performs the final object detection tasks, generating bounding boxes and class probabilities for each detected object. The overall structure of the YOLOv4 framework is illustrated in Fig. 1, while Fig. 2 shows the sequential processing pipeline, highlighting the data flow through each stage of the architecture. These diagrams demonstrate how the combination of feature extraction, multiscale fusion, and prediction modules enables YOLOv4 to achieve efficient and accurate object detection in real time.

2.2 YOLOv7[8]

YOLOv7, introduced by Wang in 2022,[8] is an advanced real-time object detection algorithm that represents a significant enhancement over previous models in the YOLO family. This version retains the rapid inference capability characteristic of YOLO architectures while incorporating several critical improvements aimed at both accuracy and computational efficiency. The core design integrates reparameterized blocks and a modular architecture, allowing the model to operate under distinct configurations during training and inference. In the training phase, a more complex network is utilized to improve the learning capacity, whereas the

Fig. 1. (Color online) YOLOv4 network architecture.

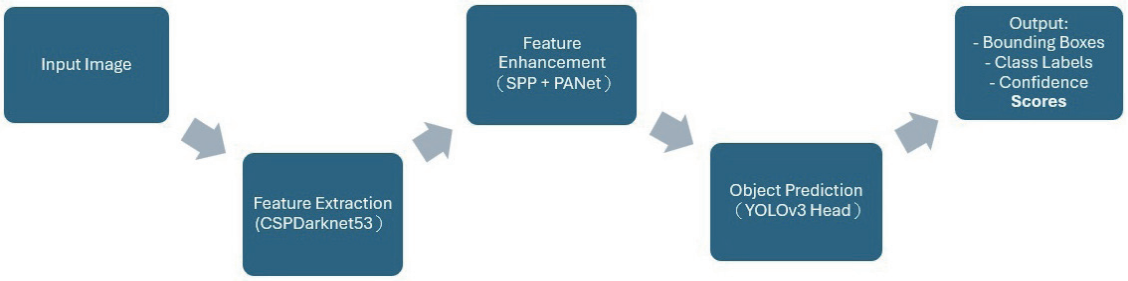

Fig. 2. (Color online) YOLOv4 processing flow.

inference phase adopts a simplified structure to ensure high-speed execution without sacrificing accuracy. Moreover, YOLOv7 adopts novel training strategies such as task-aligned learning and the coarse-to-fine head structure, enabling superior multitask learning performance across object detection, bounding box regression, and classification.

As illustrated in Fig. 3, the overall YOLOv7 framework comprises three major modules: (1) an enhanced backbone employing convolution-BatchNorm-SiLU (CBS) blocks with reparameterization; (2) a multiscale feature aggregation neck designed with ELAN and SPPCSPC modules; and (3) a task-aligned detection head capable of handling object localization and classification with improved precision. Evaluations conducted on benchmark datasets such as COCO and Pascal VOC have demonstrated that YOLOv7 consistently surpasses prior YOLO versions, including YOLOv5 and YOLOv6, in terms of both mean average precision (mAP) and inference speed. In particular, YOLOv7-tiny achieves high frame rates on the COCO dataset while maintaining competitive accuracy, indicating its suitability for deployment in edge-computing scenarios and latency-sensitive applications. These advancements underscore the model's applicability in various domains, including intelligent video surveillance, autonomous driving, and industrial machine vision systems.

2.3 ASFF

The ASFF module serves as an effective feature aggregation strategy designed to enhance multiscale representation in object detection networks. In fruit detection tasks, target objects frequently vary in size and may suffer from partial occlusion or overlap caused by foliage or clustering. ASFF addresses these challenges by allowing the network to dynamically adjust the contribution of features from different scales at each spatial location. Unlike conventional fusion strategies such as summation or concatenation, which statically combine features, ASFF introduces learnable spatial attention weights that selectively emphasize the most informative

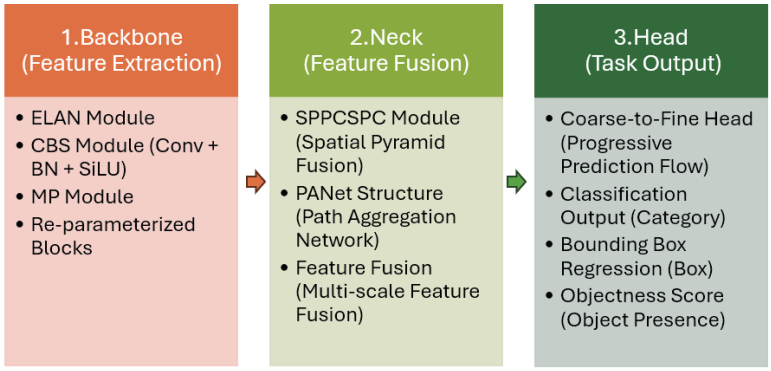

Fig. 3. (Color online) Architecture of the YOLOv7 model.

features. This adaptive mechanism significantly strengthens the network's ability to localize and classify fruits under complex agricultural conditions.

To ensure both precision and computational efficiency, ASFF is integrated into the YOLOv7 framework. YOLOv7, characterized by its optimized backbone and real-time detection capability, provides a robust foundation for edge deployment in smart farming applications. By embedding ASFF into the feature pyramid network of YOLOv7, the proposed YOLOv7-ASFF architecture enhances robustness against small-scale, overlapping, and partially occluded fruits while maintaining near real-time inference. In a previous study,[1] it was further demonstrated that ASFF improves mAP across object sizes with only marginal computational overhead, validating its practical benefits.

The structural role of ASFF within the YOLOv7 framework is illustrated in Fig. 4. Multiscale features extracted from the backbone are first aligned in resolution through up-sampling or down-sampling operations. These features are then adaptively fused at each spatial location by employing learnable attention weights, enabling the network to dynamically prioritize the most relevant information. This process preserves fine-grained details from shallow layers while simultaneously leveraging high-level semantic context from deeper layers, thereby improving detection robustness across diverse fruit sizes and environmental conditions.

The advantages of integrating ASFF into YOLOv7 can be summarized as follows.

- **Improved Multiscale Detection**: ASFF enhances the detector's ability to capture small and partially occluded fruits by adaptively leveraging complementary information across feature levels.
- **Better Generalization**: The adaptive fusion mechanism provides robustness against variations in illumination, occlusion, and background complexity, which is critical for real-world robotic harvesting and sorting systems.
- **Near Real-Time Performance**: Despite the additional computations introduced by ASFF, the YOLOv7-ASFF model maintains inference efficiency suitable for robotic arm integration and other edge-computing scenarios in smart agriculture.

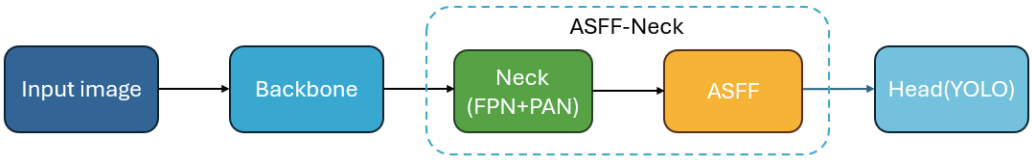

Fig. 4. (Color online) Illustration of YOLOv7-ASFF model structure.

3. Hardware Architecture

3.1 Experimental environment

As illustrated in Figs. 5 and 6, the experimental setup developed in this study consists of a robotic arm, a camera, a fruit recognition zone, and six designated fruit placement areas. Initially, a fruit is placed within the recognition zone. The camera captures an image of the object, and the YOLOv4 algorithm is applied to identify the fruit and its location. The detection results are then transmitted via serial communication to the robotic arm, which executes the corresponding pick-and-place operation. A pneumatic compressor is used to drive the air-powered gripper, enabling the arm to grasp and relocate the object to its designated location.

In this study, the robotic arm system is operated using NVIDIA Jetson Nano, a compact and energy-efficient computing platform designed for edge AI applications. Jetson Nano is equipped with a quad-core ARM Cortex-A57 CPU and a 128-core Maxwell GPU, offering sufficient computational performance to support real-time image processing and inference tasks. It includes 4 GB of LPDDR4 memory and is compatible with major deep learning frameworks, such as TensorFlow, PyTorch, and MXNet. The integrated JetPack SDK, which combines the Ubuntu operating system with CUDA, cuDNN, and TensorRT, provides a comprehensive development environment tailored for embedded AI applications. Owing to its low power consumption (5–10 W) and small form factor, Jetson Nano is particularly well suited to use in robotics, unmanned systems, and intelligent vision-based control.

The robotic arm utilized in this system was manufactured by Shenzhen Yuanhang Robotics Technology Co., Ltd., and features six degrees of freedom (DOF), a 62.9 cm reach, and a total weight of 1.9 kg. The arm achieves a global positioning accuracy of approximately 1 mm and has a maximum power consumption of 198 W. The end effector is an air-driven gripper constructed from aluminum alloy, which ensures a favorable balance between structural strength and low weight. This design contributes to enhanced stability and precision during high-speed movements. In addition, the gripper includes flexible, high-resilience fingers made from wear-resistant materials, which allow it to conform to objects of various shapes and surfaces. The compliant structure promotes even pressure distribution during grasping, thereby reducing the risk of object slippage or damage.

Visual input is captured by a wide-angle camera with a resolution of 1920 × 1080 pixels (progressive scan), operating at 60 frames per second and offering a 120° field of view. The camera is mounted approximately 30 cm above the workspace and angled 30° downward to

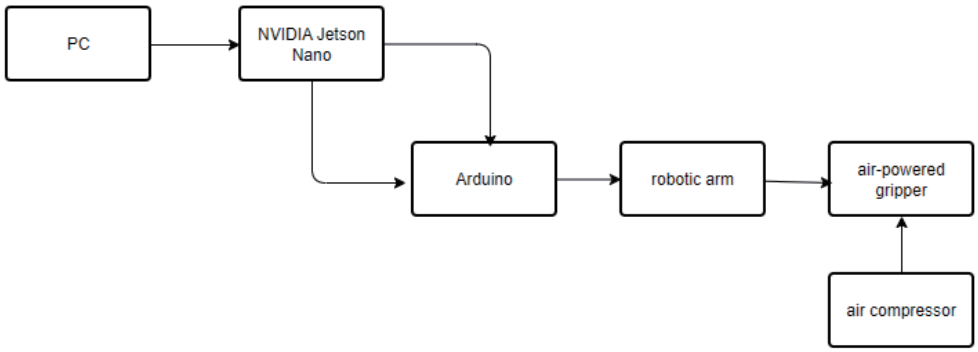

Fig. 5. Schematic of the experimental environment.

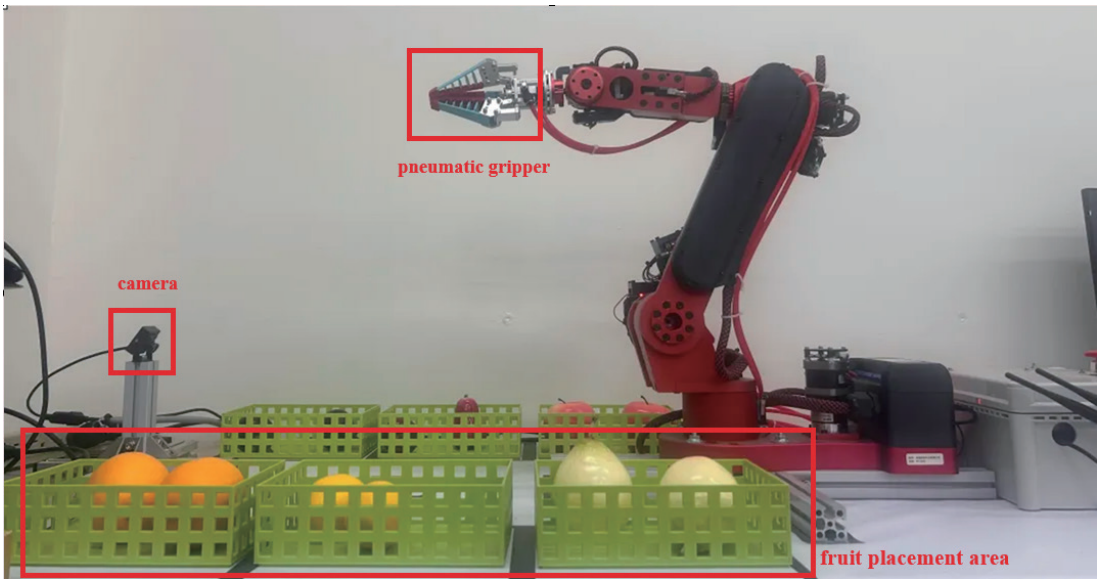

Fig. 6. (Color online) Robotic arm used in the system.

optimize the capture of the robotic arm's operating region. This setup facilitates a clear observation of the detection zone, improving object recognition and localization accuracy. The high frame rate further enhances the responsiveness of the control system, supporting real-time detection and tracking based on the YOLO algorithm. This configuration ensures that the robotic system maintains operational efficiency in dynamic and time-sensitive environments.

3.2 Denavit–Hartenberg method

The Denavit–Hartenberg (D–H) convention is a mathematical method commonly used to establish the kinematic model of a robotic arm. Proposed by Jacques Denavit and Richard S. Hartenberg in 1955, this method simplifies the description of the spatial relationship between

adjacent joints. While the relative position between two joints typically requires six parameters—three translational and three rotational—the D–H method reduces this to just four parameters.

As shown in Fig. 7, the four D–H parameters for the j-th joint are b_j, α_j, c_j, and θ_j, where j denotes the joint index. The definitions of these parameters are as follows.

- b_j is the distance between points P_j and P'_{j-1}.
- α_j is the angle of rotation from z_{j-1} to z_j, with counterclockwise rotation around x_j being positive.
- c_j is the distance between points P_j and P'_{j-1}.
- θ_j is the angle of rotation from x_{j-1} to x_j with counterclockwise rotation around z_{j-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 $(j + 1)$th axis can be represented by T_j^{-1}, T_j^{j-1}. Here, T_j^{-1}, T_j^{j-1} denotes the transformation matrix that converts the coordinates from the $(j - 1)$th axis to the jth axis, as shown in Eq. (1).

Using the D–H convention, the transformation from the coordinate frame of the $(j + 1)$th joint to that of the jth joint can be expressed as a homogeneous transformation matrix T_j^{j-1}, as defined in Eq. (1). This transformation is composed of a sequence of elementary operations: rotation about the z_{j-1}-axis by θ_j, translation along the z_{j-1}-axis by c_j, translation along the x_j-axis by b_j, and finally, rotation about the x_i-axis by α_i:

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & \cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

After calculating the transformation matrices between each pair of adjacent joints, the overall transformation from the base frame to the end-effector frame can be obtained by sequentially multiplying the individual transformations. This yields the complete forward kinematic model of the robotic arm.

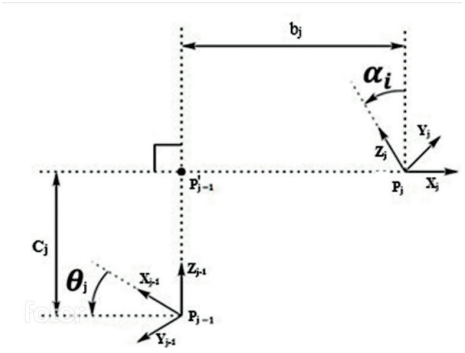

Fig. 7. Parameter explanation diagram for the D–H method.

3.3 Inverse kinematics (IK)

IK is a mathematical approach used to determine the joint angles of a robotic arm when the desired position and orientation of the end effector are known. Given a target pose for the end effector, IK allows for the computation of the corresponding joint configurations that will move the arm precisely to the intended location. Since robotic arms typically consist of multiple joints, where the motion of one joint affects the posture of others, solving the IK problem requires the careful consideration of the robot's geometric structure and motion constraints.

For a 6-DOF robotic manipulator, IK often yields multiple solutions. This is due to the redundancy in joint DOFs, which allows the robot to reach the same end-effector pose using different joint configurations.

On the basis of the D–H convention, the relative transformation between adjacent joints is expressed using a series of homogeneous transformation matrices denoted as T_i^{i-1}. These matrices describe the coordinate transformation from the (i-1)-th joint frame to the i-th joint frame. For a 6-DOF robotic arm, if the desired position and orientation of the end effector are known, the overall transformation matrix T_0^6 can be used to derive the joint variables through inverse kinematic analysis.

The overall transformation from the base frame to the end-effector frame is given by

$$T_0^6 = T_5^6 \times T_4^5 \times T_3^4 \times T_2^3 \times T_1^2 \times T_0^1 . \tag{2}$$

Since each transformation matrix comprises both a rotation matrix and a translation vector, it is possible to extract the individual joint angles and displacements by applying inverse operations to these matrices. Ultimately, by solving the IK equations, the desired joint configurations can be obtained. This enables the robotic arm to accurately perform predefined tasks, achieving high-precision target positioning and object manipulation.

3.4 Forward kinematics (FK)

FK is the process of determining the position and orientation of a robotic arm's end effector on the basis of the known joint angles and link parameters. In contrast to IK, where joint variables are computed from a desired end-effector pose, FK calculates the resulting pose when the joint configurations are already specified. This technique is essential in robot control and path planning, as it allows for the precise determination of the end effector's absolute location in space.

The core principle of forward kinematics lies in a sequence of geometric transformations that describe the spatial relationships between adjacent links and joints. Typically, the D–H convention is adopted to model these relationships mathematically. Using the D–H parameterization, each joint can be represented by a standard homogeneous transformation matrix T_i^{i-1}, which defines the transformation from one joint coordinate frame to the next.

By successively multiplying these transformation matrices, the overall transformation from the robot base to the end effector can be obtained. For a 6-DOF robotic arm, the FK can be expressed as

$$T_0^1 \times T_1^2 \times T_2^3 \times T_3^4 \times T_4^5 \times T_5^6 = T_0^6 . \quad (3)$$

Here, T_0^6 represents the final homogeneous transformation matrix from the base frame to the end-effector frame, and each T_i^{i-1} is computed from the joint's rotation and translation parameters. These matrices contain both rotational and translational components, enabling a complete description of the robot's pose.

In practical applications, FK is often used to simulate the motion trajectory of the robotic arm, verify whether the arm can reach a target position, or serve as a reference output in closed-loop control systems to ensure that the robot performs as expected. Since FK calculations are unidirectional and deterministic, they are generally more straightforward than IK calculations and do not suffer from issues such as multiple solutions. As a result, FK is widely applied in robotic path planning and control algorithms.

4. Experimental Methods and Analysis

4.1 Integration process

(1) Baseline Selection: We adopt YOLOv7 as the base architecture because of its performance and modularity.
(2) ASFF Placement: The ASFF module is integrated into the neck portion of the YOLOv7 architecture, replacing or augmenting the traditional PANet- or FPN-based feature fusion layers. Specifically, ASFF receives feature maps from different stages (e.g., P3, P4, and P5) and fuses them adaptively.
(3) Channel and Spatial Alignment: Feature maps from each scale are resized to the same spatial dimensions and channel depths before being passed to the ASFF block.
(4) Adaptive Weighting: Within the ASFF module, spatial attention weights are learned during training to emphasize more informative regions at each scale. This allows the model to dynamically adjust its focus during inference.
(5) Detection Head: The fused feature map is then passed to the YOLOv7 detection head, where object classification and bounding box regression are performed.
(6) Training Strategy: The model is trained end-to-end on a customized fruit dataset using standard YOLO loss functions [including objectness (i.e., the confidence score representing the probability that an object exists in the bounding box), class probability, and bounding box regression losses], with additional regularization to stabilize the learning of attention weights in ASFF.

4.2 Experimental procedure

To evaluate the accuracy and robustness of the object recognition model adopted in this study under different angles and object orientations, the experimental design deliberately incorporates various real-world challenges. In addition to variations in the appearance, direction, and placement of the objects, the environment was configured with more realistic and complex

background elements—such as nonuniform surface textures, natural lighting variations, shadows, reflective surfaces, and other visual noise—to simulate the perceptual disturbances commonly encountered in agricultural, logistics, and processing environments.

As illustrated in Fig. 8, two representative types of experimental environments were considered: a simple background [Fig. 8(b)] and a complex background [Fig. 8(a)]. The simple background corresponds to a controlled environment with a uniform surface and minimal interference, allowing the system to be tested under idealized conditions. In contrast, the complex background represents an orchard-like environment, where fruits are embedded within dense foliage and branches, and are affected by natural lighting variation, occlusion, and irregular textures. These contrasting settings enable a systematic evaluation of the system's robustness by exposing the model to both laboratory-style conditions and realistic agricultural scenarios.

The experiment is conducted in two stages. In the first stage, in a simple background setting, each fruit is placed in six distinct orientations—top view, side view, bottom view, inclined, and asymmetrical positions. Real-time detection and classification are then performed by the models to compare the recognition performance characteristics of YOLOv7 and YOLOv7 integrated with the ASFF module (YOLOv7-ASFF). In the second stage, the same experiment is repeated with a visually complex background to investigate whether the performance gap between YOLOv7 and YOLOv7-ASFF becomes more significant in a challenging environment. This two-stage evaluation aims to verify whether integrating the ASFF module effectively enhances YOLOv7's recognition performance, particularly under complex visual conditions.

4.3 Experimental results

To quantitatively compare the performance characteristics of different object detection models, YOLOv4, YOLOv7, and YOLOv7 integrated with the ASFF module (YOLOv7-ASFF) were evaluated in a simple background environment. The evaluation metrics used in this study include mean average precision at an IoU threshold of 0.5 (mAP@0.5), Precision, Recall, and F1 score.

(a) (b)

Fig. 8. (Color online) Representative images of experimental environments: (a) complex and (b) simple background.

mAP@0.5 reflects the model's ability to correctly localize and classify objects under a relatively lenient criterion. Precision measures how many of the model's positive predictions are correct (i.e., low false positive rate), while Recall indicates how many of the actual targets are successfully detected (i.e., low false negative rate). F1 score, the harmonic mean of Precision and Recall, provides a balanced measure of the model's overall detection performance.

As shown in Table 1, YOLOv7-ASFF outperformed both YOLOv7 and YOLOv4 across all metrics. Its mAP@0.5 reached 0.93, indicating excellent localization and classification capability. Compared with YOLOv7 (mAP@0.5 = 0.89), the integration of the ASFF module led to improvements in Recall (0.91 vs 0.86), highlighting better generalization to varying object poses. YOLOv4 exhibited the lowest performance across the board (e.g., mAP@0.5 = 0.82, Recall = 0.80), reflecting its relatively outdated architecture. Overall, the results demonstrate that the addition of ASFF effectively enhances YOLOv7's detection accuracy and robustness, even under relatively simple visual conditions.

To further examine model robustness under realistic field conditions, the same evaluation was conducted using more visually complex backgrounds. These environments included an orchard setting with dense foliage and variable lighting, as well as a structured conveyor platform commonly used in post-harvest processing. These backgrounds introduced moderate visual disturbances such as nonuniform illumination, natural occlusion, and reflective surfaces, simulating common challenges in agricultural and logistics applications. The corresponding performance metrics are presented in Table 2.

Overall, all models experienced some degree of performance degradation owing to the increased visual complexity. However, YOLOv7-ASFF maintained relatively stable results, indicating superior adaptability to realistic field conditions. Specifically, YOLOv7-ASFF achieved an mAP@0.5 of 0.89 and an F1 score of 0.89, demonstrating high accuracy and consistency. In contrast, YOLOv4 exhibited a notable drop in performance (mAP@0.5 = 0.72; Recall = 0.68), suggesting limited robustness. YOLOv7 showed moderate decreases across several metrics, with Recall dropping to 0.79. These findings confirm that the ASFF module significantly improves YOLOv7's detection performance under realistic operating scenarios, making it more suitable for practical deployment.

Table 1
Performance metrics under simple background conditions.

Metric	YOLOv4	YOLOv7	YOLOv7-ASFF
mAP@0.5	0.82	0.89	0.93
Precision	0.88	0.92	0.95
Recall	0.80	0.86	0.91
F1 score	0.84	0.89	0.93

Table 2
Performance metrics under complex background conditions.

Metric	YOLOv4	YOLOv7	YOLOv7-ASFF
mAP@0.5	0.72	0.83	0.89
Precision	0.78	0.86	0.91
Recall	0.68	0.79	0.87
F1 score	0.72	0.82	0.89

4.4 Results of grasping experiment

Our experimental design involved testing the grasping performance of the proposed YOLOv7-ASFF model using six different types of fruit: apple, orange, pear, mangosteen, lemon, and wax apple. The experiments were carried out under two scenarios: a simple background with a uniform surface and a complex background with multiple fruits and distractors to simulate real-world conditions. Performance was evaluated in terms of grasping success rate, average grasping time, and failure cases, which were further categorized into misdetection, missed detection, pose deviation, and slippage. For each condition, 30 grasping trials were conducted and the results of YOLOv7-ASFF were compared against those of the baseline YOLOv7 model.

For each fruit type, a fixed placement location was designated, and the corresponding return coordinates were programmed into the robotic arm control system, with the end effector referenced to the base as the origin, as shown in Table 3. This configuration ensures that after each grasping operation, the fruit is accurately returned to its assigned category, thereby maintaining consistency in sorting outcomes and experimental repeatability. Furthermore, this approach allows the control system to efficiently coordinate the grasping and returning motions, facilitating subsequent performance analysis and process optimization.

As illustrated in Fig. 9, the experimental process of fruit recognition and manipulation by the robotic arm is demonstrated in four sequential stages. In the initial state [Fig. 9(a)], the robotic arm is positioned at its standby location, awaiting detection results. A camera-based vision sensor identifies the fruit type and designates the placement area accordingly. Once the fruit is positioned in the designated zone [Fig. 9(b)], the robotic arm receives the control command and proceeds to grasp the target object [Fig. 9(c)]. Subsequently, the robotic arm executes the returning motion to relocate the fruit back to its original category bin [Fig. 9(d)]. This sequence validates the integration of the vision sensor and robotic manipulation system, ensuring accurate object recognition, grasping, and sorting functionality under real-world conditions. The final experimental results are shown in Table 4.

4.5 Analysis of experimental results

The experimental results presented in Table 4 clearly demonstrate the effectiveness of integrating the ASFF module into the YOLOv7 framework. Compared with the baseline YOLOv7, the YOLOv7-ASFF model achieved significantly higher grasp success rate in both

Table 3
Fruit-specific placement and return coordinates.

Fruit	*X*	*Y*	*Z*
Apple	122.4	319.2	268
Orange	171.2	−313.8	268
Pear	122.4	−319.2	268
Mangosteen	171.2	313.8	268
Lemon	146.8	325.9	268
Wax apple	146.8	−325.9	268

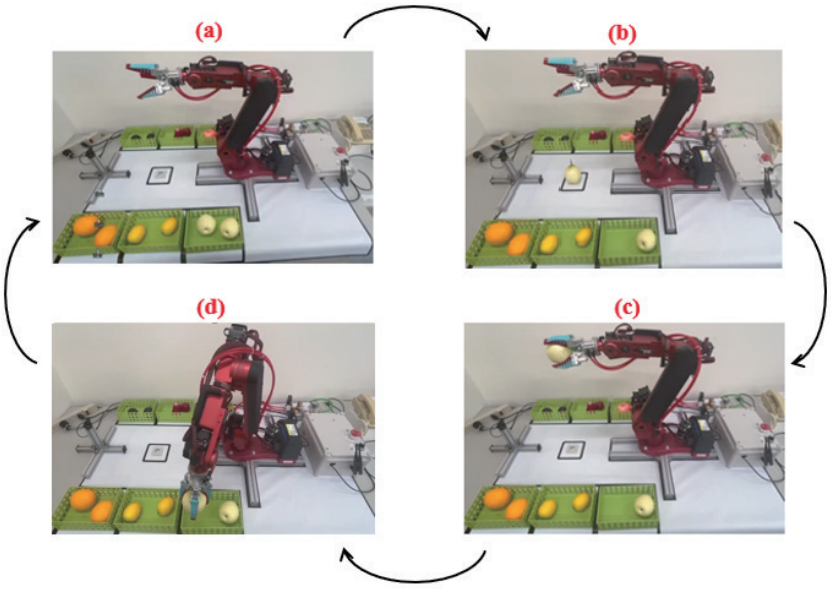

Fig. 9. (Color online) Fruit grasping process.

Table 4
Quantitative results.

Metric	Background	YOLOv7	YOLOv7-ASFF
Grasp success rate (%)	Simple	82.0	92.0
	Complex	68.0	85.0
Failure cases (per 30 trials)	Simple	5	2
	Complex	9	4

simple and complex backgrounds, with improvements from 82.0 to 92.0% and from 68.0 to 85.0%, respectively. In addition, the number of failure cases per 30 trials was markedly reduced, particularly in the complex background scenario, where the number of failures decreased from 9 to 4. These reductions were mainly attributed to fewer instances of misdetection and pose deviation, indicating that the ASFF module enhanced the stability and robustness of fruit localization under challenging conditions.

The analysis results confirm that the improved detection accuracy provided by YOLOv7-ASFF directly contributes to higher reliability in robotic grasping. Under simple background conditions, the model achieved nearly flawless performance with only minimal errors, while under realistic complex environments characterized by clutter and distractors, the system maintained stable and consistent grasping outcomes. This robustness is crucial for agricultural automation, where environmental variations often challenge vision-based systems.

5. Conclusions

We presented an interactive teaching platform that integrates real-time fruit detection and robotic grasping using the YOLOv7-ASFF algorithm and a vision-based sensing system. The

results of experimental evaluation demonstrated that the proposed model achieved a mean average precision of 94.6% and a grasping success rate of 93%, confirming both the accuracy of fruit recognition and the reliability of robotic manipulation. By combining deep learning-based visual perception, adaptive grasping strategies, and modular hardware design, the system not only demonstrated the effectiveness of sensor–algorithm integration but also provided a practical framework for exploring intelligent automation.

Acknowledgments

This research is partially supported by National Science and Technology Council, Taiwan, under contract: NSTC 114-2221-E-224-050, 113-2221-E-224-037, and 113-2622-E-224-017, and IRIS "Intelligent Recognition Industry Service Research Center" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

References

1 J. Li, M. Xue, M. Zhang, J. Yin, Y. Liu, X. Qiao, D. Zheng, and Z. Li: Agronomy **13** (2023) 1901. https://doi.org/10.3390/agronomy13071901
2 Y. Liu, B. Liu, and Y. Li: Remote Sens. **14** (2022) 3498.
3 H. Wang, J. Liu, X. Gao, L. Zhang, and C. Wang: IET Image Process. **17** (2023) 2653. https://doi.org/10.1049/ipr2.13304
4 K. Liu, Y. Qi, G. Xu, and J. Li: IET Image Process. **18** (2024) 13024. https://doi.org/10.1049/ipr2.13024
5 K. Zhao, R. Lu, S. Wang, X. Yang, Q. Li, and J. Fan: Front. Neurorobot. **17** (2023) 1170163. https://doi.org/10.3389/fnbot.2023.1170163
6 G. Liu, Y. Zhang, J. Liu, D. Liu, C. Chen, Y. Li, X. Zhang, and P. L. Touko Mbouembe: Front. Plant Sci. **15** (2024) 1452821. https://doi.org/10.3389/fpls.2024.1452821
7 A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao: arXiv (2020) 2004.10934. https://doi.org/10.48550/arXiv.2004.10934
8 Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (2023) 7464–7473. https://doi.org/10.1109/CVPR52729.2023.00764

About the Authors

Chun-Chieh Wang is a contract-based professor at the Department of Electrical Engineering of National Yunlin University of Science and Technology. His areas of research interest include robotics, image detection, electromechanical integration, innovative inventions, long-term care aids, and the application of control theory. He is now a permanent member of the Chinese Automatic Control Society (CACS) and the Taiwan Society of Robotics (TSR). He is also a member of the Robot Artificial Life Society. (jasonccw@yuntech.edu.tw)

Chung-Wen Hung received his Ph.D. degree in electrical engineering from National Taiwan University in 2006. Currently, he is a professor at National Yunlin University of Science and Technology. His research interests include IoT, IIoT, power electronics, motor control, and AI application.
(wnhung@yuntech.edu.tw)

Sun-Jing Yan is a Master's student at the Department of Electrical Engineering, National Yunlin University of Science and Technology. His focus is on algorithm research, with his current work being centered on optimizing algorithms and their applications, with the aim of providing innovative solutions in relevant fields. (M11212100@yuntech.edu.tw)

Chian-Cheng Ho received his Ph.D. in electrical engineering from National Chung Cheng University in 2000. He was a visiting Ph.D. student at UCLA in 2000 and worked at ChungHwa Telecom Labs until 2005. Since then, he has been an associate professor at National Yunlin University of Science and Technology. His research focuses on embedded processor firmware, real-time operating systems, and visual robotics applications.
(futureho@yuntech.edu.tw)