smart cities

MDPI

Article

Real-Time Recognition and Localization of Apples for Robotic Picking Based on Structural Light and Deep Learning

Quan Zhang [1,2] and Wen-Hao Su [1,*]

[1] College of Engineering, China Agricultural University, 17 Qinghua East Road, Haidian, Beijing 100083, China
[2] School of Mechanical Engineering, Tongji University, 4800 Caoan Road, Jiading, Shanghai 201800, China
* Correspondence: wenhao.su@cau.edu.cn

Abstract: The apple is a delicious fruit with high nutritional value that is widely grown around the world. Apples are traditionally picked by hand, which is very inefficient. The development of advanced fruit-picking robots has great potential to replace manual labor. A major prerequisite for a robot to successfully pick fruits the accurate identification and positioning of the target fruit. The active laser vision systems based on structured algorithms can achieve higher recognition rates by quickly capturing the three-dimensional information of objects. This study proposes to combine the laser active vision system with the YOLOv5 neural network model to recognize and locate apples on trees. The method obtained accurate two-dimensional pixel coordinates, which, when combined with the active laser vision system, can be converted into three-dimensional world coordinates for apple recognition and positioning. On this basis, we built a picking robot platform equipped with this visual recognition system, and carried out a robot picking experiment. The experimental findings showcase the efficacy of the neural network recognition algorithm proposed in this study, which achieves a precision rate of 94%, an average precision mAP% of 92.86%, and a spatial localization accuracy of approximately 4 mm for the visual system. The implementation of this control method in simulated harvesting operations shows the promise of more precise and successful fruit positioning. In summary, the integration of the YOLOv5 neural network model with an active laser vision system presents a novel and effective approach for the accurate identification and positioning of apples. The achieved precision and spatial accuracy indicate the potential for enhanced fruit-harvesting operations, marking a significant step towards the automation of fruit-picking processes.

Keywords: agricultural automation; fruit detection; artificial intelligence; neural networks; YOLOv5; laser vision system; three-dimensional recognition; precision agriculture

Citation: Zhang, Q.; Su, W.-H. Real-Time Recognition and Localization of Apples for Robotic Picking Based on Structural Light and Deep Learning. *Smart Cities* **2023**, *6*, 3393–3410. https://doi.org/10.3390/smartcities6060150

Academic Editor: Pierluigi Siano

Received: 26 October 2023
Revised: 23 November 2023
Accepted: 30 November 2023
Published: 4 December 2023

1. Introduction

Apples are rich in vitamins C and E [1], offering a wealth of nutritional value with a low fat content and high carbohydrates. Their delightful, sweet taste has made them a favorite among consumers. They stand as one of the world's most extensively cultivated, highest-yielding, and globally traded fruits. However, in current practical production, apple harvesting remains largely reliant on manual labor, which can impact both efficiency and quality. There is an urgent need for automated picking robots due to the high demand for labor during harvest seasons [2]. The rising cost of manual harvesting, driven by the aging population and decreasing agricultural workforce, underscores the necessity for cost-effective alternatives. Harvesting robots, operating continuously, offer heightened efficiency and lower costs.

Harvesting robots comprise two primary subsystems: the vision system and the actuator system [3]. The vision system guides the robot's actuators in detecting and localizing apples on trees [4]. Target localization stands as a critical aspect of apple-harvesting robots. In recent years, researchers have delved deep into utilizing machine vision for target localization. Depending on the method of obtaining depth information,

three main categories emerge: binocular stereo vision, structured light, and time-of-flight [5]. Binocular stereo vision is sensitive to ambient lighting and is unsuitable for monotone and textureless scenes. Its high computational complexity and baseline limitations also constrain the measurement range, and these drawbacks should be acknowledged [6]. Time-of-flight technology may encounter measurement errors and failures under external interference and high illumination conditions [7]. Scattered structured-light technology, distinguished by its compact size, low resource consumption, active measurement, high precision, and resolution, has garnered substantial attention [8]. With a smaller camera baseline and lower resource requirements, it exhibits potential for widespread application [9].

Feng et al. designed a structured-light vision system for a tomato-harvesting robot. As demonstrated by the field test results, the measurement error for the fruit radius is less than 5 mm, the center distance error between the fruit and camera is less than 7 mm, and the single-axis coordinate error is less than 5.6 mm [10]. Jimenez et al., in their developed citrus-harvesting robot, implemented a laser-based active vision system, achieving an accuracy of approximately 10 mm in three-dimensional fruit positioning, with an estimated average error in fruit radius of under 5 mm. This system solves the challenging task of identifying target fruits in unstructured operational environments [11]. Setting up an orchard is a crucial aspect and a focal point in our agricultural automation efforts, emphasizing the integration of agricultural machinery with agricultural technology. An efficient orchard layout is essential for the successful implementation of robotic harvesting. This involves strategically positioning fruit trees and regularly pruning them to optimize their suitability for robotic harvesting tasks. Considering the influence of complex natural conditions and equipment costs, the structured-light localization method was chosen for this experiment.

The recognition of target fruits is a crucial component of harvesting robot technology. In response to the challenges of target fruit recognition, domestic and international researchers have proposed various methods. Initially, single-feature analysis methods were employed, but they proved to be inaccurate and unstable. These methods primarily relied on color features to determine whether a fruit is a target, but they suffered from drawbacks such as low recognition accuracy, limited robustness, and poor adaptability. Building upon single-feature analysis, researchers introduced multi-feature fusion approaches (color, geometric shape, texture) to enhance recognition success rates. Fusing information from these different types of data can improve the successful identification of target fruits. In addition to multi-feature fusion analysis methods, approaches based on neural networks have also proven effective for target fruit recognition. Compared to these traditional detection methods, deep learning techniques exhibit a more promising performance in the field of object detection. They offer higher accuracy, surpassing conventional image-processing approaches [12]. The study by Koirala et al. [13] indicates that deep learning algorithms have been recommended for fruit tree detection. Sa et al. [14] employed the Faster R-CNN (Faster Regional Convolutional Neural Network) algorithm [15] to detect multi-colored (green, red, yellow) pepper fruits. Chen et al. [16] used fully convolutional networks [17] to count fruits in apple and orange orchards. Bargoti and Underwood [18] utilized Faster R-CNN and transfer learning to estimate yield in apple, mango, and almond orchards. Gao et al. employed Faster R-CNN (Regional Convolutional Neural Network) for the detection of apples, achieving an average precision (AP) of 0.879 [19]. Xiao et al. utilized a backpropagation neural network to train an apple color recognition model, effectively identifying apples on fruit trees [20]. Fu et al. used ZFNet to detect apples from segmented RGB images, achieving an AP of 0.805 [21]. Neural networks can learn from extensive data to extract features for tasks such as classification or regression automatically. The YOLO (You Only Look Once) algorithm is a typical example of this approach [6]. In 2019, Wang et al. [22] introduced a mango fruit detection method based on deep learning algorithms. This method utilizes a deep learning algorithm based on the YOLO model to identify target fruits in each frame of an image. The experimental results indicate that the algorithm can accurately identify fruit targets when processing tracking videos. These findings suggest that by applying deep learning technology, detecting and localizing fruits is feasible.

This study employs an improved structured-light localization method based on YOLOv5. Building upon the original neural network model, optimizations were made specifically for apple detection. After obtaining the 2D coordinates of the targets, these coordinates were transformed and input into the calibrated structured-light camera's world coordinate system to derive the 3D world coordinates of the apples. The performance of the model was assessed using the mean average precision (mAP) metric. The standard deviation of depth and localization precision were calculated to evaluate the accuracy of apple localization.

2. Materials and Methods

2.1. Image Acquisition

The apple orchard image data used in this study were obtained from the Guoku Orchard in Changping District, Beijing, China. To simulate the actual harvesting process, the images were collected during the apple-harvesting season in mid to late October. The spacing between the apple trees is typically between 3 m and 5 m. The formula for calculating the number of plants per hectare is 10,000/(plant spacing × row spacing) (unit: meters). Due to the uneven distribution of row spacing in the orchard images collected for this experiment, precise figures cannot be provided. However, based on an average spacing of 4 m between rows and trees, the number of apple trees per hectare is estimated to be around 625. Factors such as soil fertility limitations and actual orchard usage may result in the actual plant density per hectare being lower than this calculated value. Regarding the thickness of the plant canopy, it varies for different tree forms, mainly ranging from 0.5 to 2.0 m within the canopy. A single-lens reflex (SLR) camera, Sony α6000L, Brand: SONY, Japan, Origin: Wuxi, China, equipped with a fixed macro lens, was used for image acquisition. The camera operated in automatic mode, adjusting the appropriate capture parameters, including the white balance, ISO speed, and exposure time. Multiple samples were taken at different times, under various weather and lighting conditions, in the apple orchard to capture a large number of images of mature apples of different types. A total of 1000 images were selected as the research materials based on image quality and apple distribution. The development process of the apple picking robot in this study is shown in Figure 1.

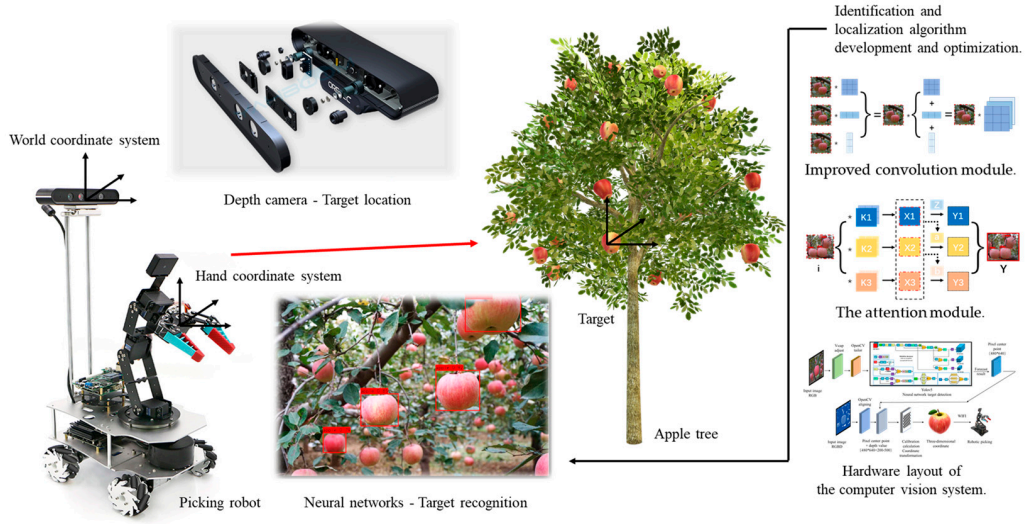


Figure 1. This study is centered around the development of an apple-picking robot. We have engineered an enhanced YOLOv5 object detection model and pioneered an innovative computer vision localization technique. By integrating a depth camera ranging module, we embedded depth measurement functionality into the neural network model. The fusion of this depth measurement functionality with the object detection capability achieves the seamless integration of apple recognition and three-dimensional spatial positioning. The * symbol in the graph represents the convolution operation.

2.2. Deep Learning Model

Deep learning, which has gained significant breakthroughs in the field of machine learning, is based on the construction of multi-layer artificial neural networks. It possesses powerful learning capabilities and computational performance, with its key advantage being the ability to learn features automatically (automatic feature learning). Among them, YOLOv5 is a high-precision and high-speed object detection model that can process 140 frames per second. Compared to the YOLOv4 model, YOLOv5 reduces the training weights by nearly 90%, making it highly suitable for real-time object detection deployment on small devices [20]. Therefore, this study adopts YOLOv5 as the foundation, combining it with other advanced neural network model modules in the field of deep learning. It optimizes these models based on the specific requirements of the apple-picking robot to further improve the recognition performance of apple targets and construct an apple recognition network for the apple-picking robot.

YOLOv5 is a popular object detection model that can be divided into four architectures, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, based on the number of feature extraction modules and convolutional kernel sizes [23]. When selecting the optimal object detection model, multiple factors such as speed, accuracy, user-friendliness, and developer experience need to be considered. YOLOv5, being more user-friendly, has gained favor among many developers. Despite being an older version, YOLOv5 has achieved widespread adoption in the community. Its popularity can be attributed to factors such as user familiarity, extensive documentation, and a robust user base, making it a reliable and well-supported choice for various applications, significantly aiding in model optimization. Considering the real-time requirement and lightweight network structure for apple recognition in this study, the YOLOv5m architecture was chosen as the base network. The network model was then optimized and upgraded to meet this specific requirements of apple recognition in this study. The YOLOv5m model mainly consists of the backbone, neck, and detect networks. Based on the YOLOv5m architecture, this study made improvements and designs tailored to the requirements of the apple-picking robot, considering real-time performance and lightweight demands. By optimizing the model's architecture and parameter settings, the accuracy of apple recognition for the apple-picking robot was improved.

The model training was conducted using the PyTorch deep learning framework in Python 3.8. The configuration included an NVIDIA GeForce GTX 1650, PyTorch 1.10.1, NumPy 1.21.2, CUDA Toolkit 11.3.1, and other relevant libraries. The labeled dataset was divided into training and test parts and placed in their respective folders. The number of epochs, which determines the training iterations, was set. The model was then trained on either the GPU of the local computer or a cloud-based GPU provided by Google. Throughout the training process, metrics such as recall, accuracy, precision, and average precision were evaluated, and adjustments were made to hyperparameters based on the actual performance. Eventually, experimentation and parameter tuning allowed us to obtain a relatively ideal and stable set of training parameter weights. The loss function converged, and the average precision was relatively high.

The entire model training and validation process was performed on a computer with an Intel(R) Core(TM) i5-9300H CPU @ 2.40 GHz processor, 16 GB RAM, and a 64-bit Windows 10 operating system. The training speed was optimized using the graphics processing unit (GPU) of the device, which was an NVIDIA GeForce GTX 1650 with 8 GB of dedicated memory.

2.3. Dataset Labeling and Preparation

The Python 3.8 OpenCV module was used to label the dataset. The apple images were in RGB format. A portion of the images was selected, and their RGB values were recorded, normalized, and used to calculate color feature indices.

In this study, a total of 1000 images with a resolution of 6000 $\times$ 4000 pixels were selected from the previously mentioned image data as experimental materials for apple segmentation. These images were further filtered, compressed, and cropped, resulting in

722 apple orchard images for feature extraction and segmentation experiments. Manual labeling was performed on the apple regions in the high-throughput apple images from the orchard to detect target apple regions. A total of 3569 apples were labeled, with approximately 3 to 7 apples per image. Then, 648 randomly selected images were used as the training set, and 72 images were used as the validation set (with a 9:1 ratio) for model training and validation. Additionally, 50 images were randomly selected as a test set to evaluate the training results of the model. For model construction in the apple target detection task, some modeling parameters used in this study are as follows: pretrained: True; batch_size: 20; max_epoch: 350; Init_lr: 0.001; min_lr: 0.0001; optimizer: Adam; weight_decay: 0; warmup_lr_ratio: 0.1; no_aug_iter_ratio: 0.3; lr_decay_type: cos; Number of classes: 2.

All of the image labeling and processing mentioned above were performed using the Baidu Paddle EasyDL AI platform. In this study, the platform was primarily utilized to assist in annotating the target apples in the images. The process involved two steps. The first step was to annotate the apple regions in the images (apple dataset) by selecting and marking them. The second step was to upload the labeled images to the platform for training. The platform autonomously labeled additional images based on the learned patterns, and human evaluation and adjustment were performed to calibrate the annotations. This process could be repeated to improve the segmentation accuracy until the training requirements were met.

Various data augmentation methods were employed in this study to supplement and expand the image dataset, facilitating better model fitting and computation. The "resize-image" module was used to generate new orchard apple images by applying operations such as rotation, flipping, translation, and scaling, thereby increasing the number of images in the dataset. The "place-image" module handled the dataset by performing cutout operations, replicating image content, and swapping, aiming to prevent overfitting and address the issue of imbalanced samples. In a large dataset, images with disproportionately large or small apple pixel ratios could lead to imbalanced positive and negative samples during training and result in overfitting to the dominant samples. The "distort-image" module modified image parameters such as the brightness, contrast, saturation, and hue to enhance the model's robustness and generalization capability, reducing the influence of environmental factors on the images and making the model less sensitive to environmental changes. The collected image dataset was mainly captured in the same scene type, so the images had some noticeable and similar features. These data augmentation operations improved the model's fitting and computational abilities, enriched the image information, and helped to enhance model performance.

2.4. Model Optimization

2.4.1. Replacement of Convolution Kernel with Convolution Kernel Group

Since apple images are captured in complex outdoor natural scenes with the partial occlusion of apples, leaves, and branches, YOLOv5 struggles to extract clear apple features in such complex backgrounds. The backbone network is improved by replacing the convolution kernel with a convolution kernel group to address this issue. The convolution kernel group consists of three parallel convolution kernels that perform convolutions on the input image with the same stride, producing feature maps of the same size and channels. The corresponding feature maps are summed to obtain the output feature map, as shown in Figure 2. This improvement enhances the network's ability to extract apple features, reduces the influence of complex backgrounds, and improves the accuracy of object detection.

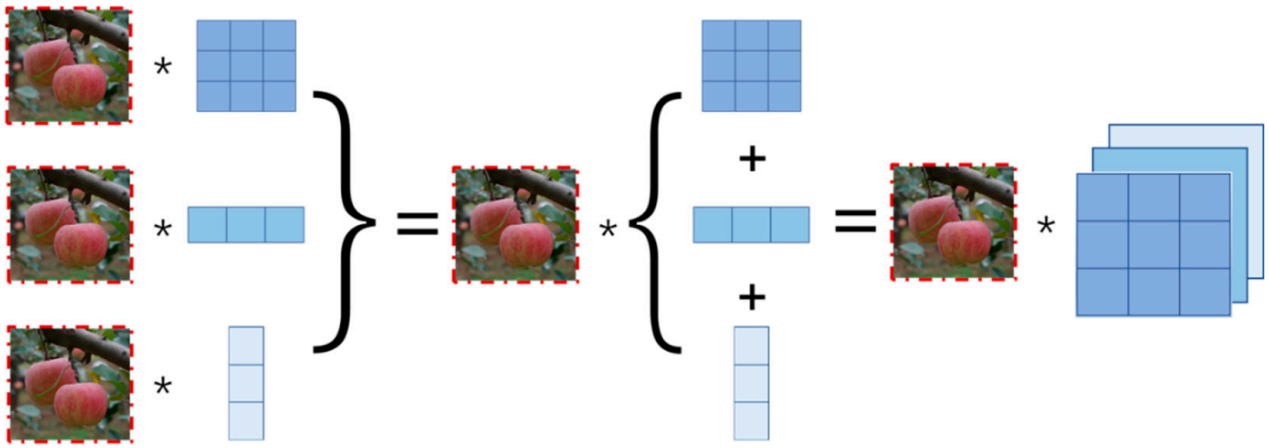


Figure 2. Improved convolution module. The convolution kernel is replaced by a group of convolution kernels, and the corresponding feature graph is summed to obtain the output feature graph. The * symbol in the graph represents the convolution operation.

2.4.2. Addition of Attention Module

An attention module is added to the YOLOv5m network, consisting of three parts, segmentation, fusion, and selection, as shown in Figure 3. In the segmentation part, the input feature map (i) is convolved with three different convolution kernels (K1, K2, K3) to generate feature maps (X1, X2, and X3). In the fusion part, the segmented feature maps (X1, X2, and X3) are combined and processed to obtain matrices (a and b). In the selection part, the feature maps X2 and X3 are weighted and selected based on matrices a and b, while the feature map X1 is weighted using the output of a fully connected layer (z), resulting in feature maps Y1, Y2, and Y3. Finally, the feature maps (Y1, Y2, and Y3) are combined to obtain the output feature map of the attention module (Y). By adding the attention module, this research effectively extracts global information about apples, reduces the impact of small and non-uniformly shaped apples, and enhances the performance of the YOLOv5 detection algorithm.

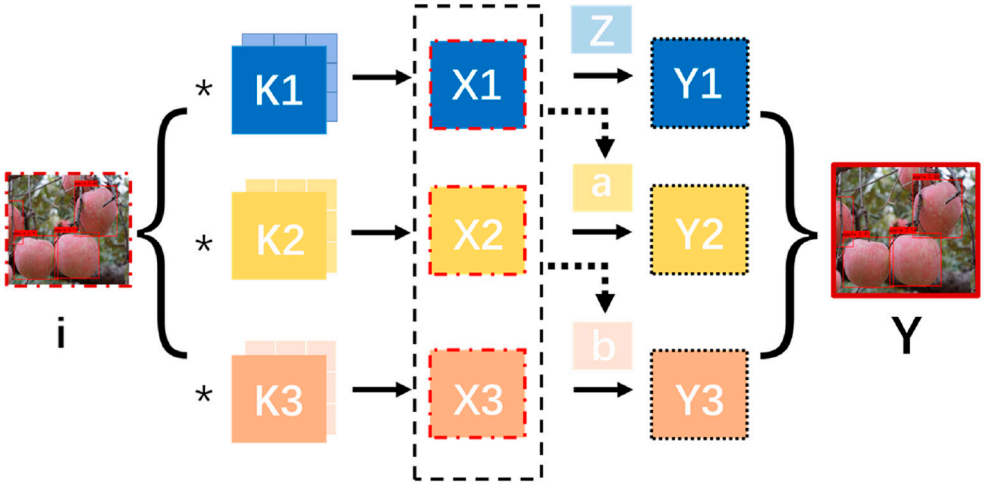


Figure 3. The attention module, consists of three parts: segmentation, fusion, and selection. K1, K2, K3: three different convolution kernels; X1, X2, and X3: feature maps; a and b: matrices; z: the output of a fully connected layer; Y1, Y2, and Y3: feature maps. The * symbol in the graph represents the convolution operation.

2.4.3. Improved Initial Anchor Box Sizes

YOLOv5m uses three initial detection anchor box sizes for each multi-scale detection layer to identify small, medium, and large objects, better addressing the recognition requirements of different-sized objects. However, for apple tree images obtained by the robotic vision system, apples located in the distance rows of the image and far away from the picking robot are not considered valid targets. Therefore, this research modifies the initial anchor box sizes in the YOLOv5m network to accurately identify fruit targets within the close picking range. The modified anchor box sizes are set as 60×70, 45×90, 85×65; 60×122, 130×90, and 120×240. Experimental tests show that the improved anchor box sizes can better identify small and medium-sized objects, thus improving the accuracy of object detection.

2.4.4. Optimization of Object Detection Model Based on Transfer Learning

To address the slow convergence and overfitting issues of apple fruit recognition and object detection algorithms under limited sample conditions, this research adopts transfer learning based on deep learning models [24] to transfer existing knowledge structures from different auxiliary domains, reducing the impact of insufficient apple fruit datasets. To utilize existing domain knowledge for apple fruit recognition tasks, considering the similarity in recognition features between multi-object apple images and single-object apple images, this research uses the VOC2012 dataset as the source domain and treats the single-object apple image dataset as the auxiliary domain for knowledge transfer. The improved YOLOv5m model is trained on the source domain dataset to obtain the source domain knowledge model [25]. The auxiliary domain knowledge model is trained on the single-object apple image dataset and then loaded into the multi-object apple image recognition task for training, achieving parameter transfer.

Experimental tests demonstrate that transferring knowledge from the multi-object apple image dataset as the auxiliary domain can effectively improve the accuracy and convergence speed of apple recognition tasks while alleviating overfitting phenomena.

2.5. Detection Performance

Detection performance in this study was evaluated using the mean average precision (mAP). Among them, the AP, as shown in formula (4), was calculated based on precision and recall, which are defined in Equations (1) and (2), respectively.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{3}$$

Predictions are categorized into four scenarios: true positive (TP), where the actual state and the predicted state are both positive, indicating a correct prediction; false negative (FN), where the predicted state is negative while the actual state is positive, signifying a prediction error; false positive (FP), where the predicted state is positive while the actual state is negative, indicating another type of prediction error; and true negative (TN), where both the predicted and actual states are negative, demonstrating a correct negative prediction. Precision represents the proportion of true positive predictions among the predictions labeled as positive. It can be perceived as the model's ability to accurately identify positive instances among its predictions. Conversely, recall is the ratio of true positive predictions to the total number of actual positive samples in the dataset. It reflects the model's capacity to detect instances of the target type within the dataset. Accuracy, as shown in formula (3), is determined based on the ratio of true positive predictions and true negative predictions to the total number of samples. It provides an overall indication of the

model's ability to predict both positive and negative cases across the entire dataset. The mAP (mean average precision) is calculated as the average of individual average precision (*AP*) values. It serves as a primary evaluation metric in object detection algorithms. Object detection models are often evaluated based on the dual metrics of speed and accuracy (mAP). A higher mAP value signifies the superior performance of the object detection model on the given dataset, indicating more effective object detection.

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \tag{4}$$

The function $p_{interp}(r)$ represents the smoothed precision–recall (PR) curve, where r denotes the recall. The PR curve is a graphical representation of a classifier's performance, with recall on the horizontal axis and precision on the vertical axis. The number of positive samples is denoted as n, r_i represents the recall of the ith positive sample, and P_i represents the precision of the ith positive sample. The calculation of $p_{interp}(r)$ involves smoothing the PR curve, where, for each point on the curve, the precision value is taken as the maximum precision value to its right. AP_i signifies the average precision for the ith category, and K denotes the total number of categories. Let r_1, r_2, …, r_n represent the recall values corresponding to the first interpolated point of each precision–recall curve segment, sorted in ascending order. Given a total of K categories, where $K > 1$, the formula for calculating the mAP is as follows:

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{5}$$

2.6. Principle of Structure Light Localization

Structured-light technology is a three-dimensional measurement technique that employs infrared lasers or other light sources as illumination. By projecting specific encoded or random patterns onto the object and subsequently decoding the patterns, the positional and depth information of the object is extracted. Through an analysis of pattern deformations, distances from every point on the object's surface to the camera can be calculated, thereby generating a three-dimensional point cloud or model.

As shown in Figure 4, a laser of a specific wavelength, after being encoded through a chip, is projected onto the object's surface. The camera, equipped with a filter, captures the reflected light. The filter restricts the camera from receiving only that specific wavelength of light. The chip then processes the encoded image received to perform decoding operations, yielding the depth data of the object. Depth information for various points on the object's surface can be obtained by comparing the offsets in the same direction.

This study ultimately chose the Astra S IR structured-light depth camera from Orbbec. Scope of work: 0.4–2 m; Field Angle: H58.4°–V45.5°; Data interface: Usb2.0; Support system: Android/Linux/Windows7/Windows10; Power dissipation < 2.4 w; dimension: 164.85 × 48.25 × 40 mm^3; Operating temperature: 10–40 °C. Due to factors such as inherent sensor noise, variations in ambient lighting, and uncertainties in-depth image data processing, there often arise instances of missing depth information. This phenomenon is particularly pronounced in regions such as object edges. To address this, bilateral filtering was applied to the depth images, as shown in Figure 5. This technique eliminates noise and preserves the details and edges of the depth images, thereby achieving improved image processing outcomes.

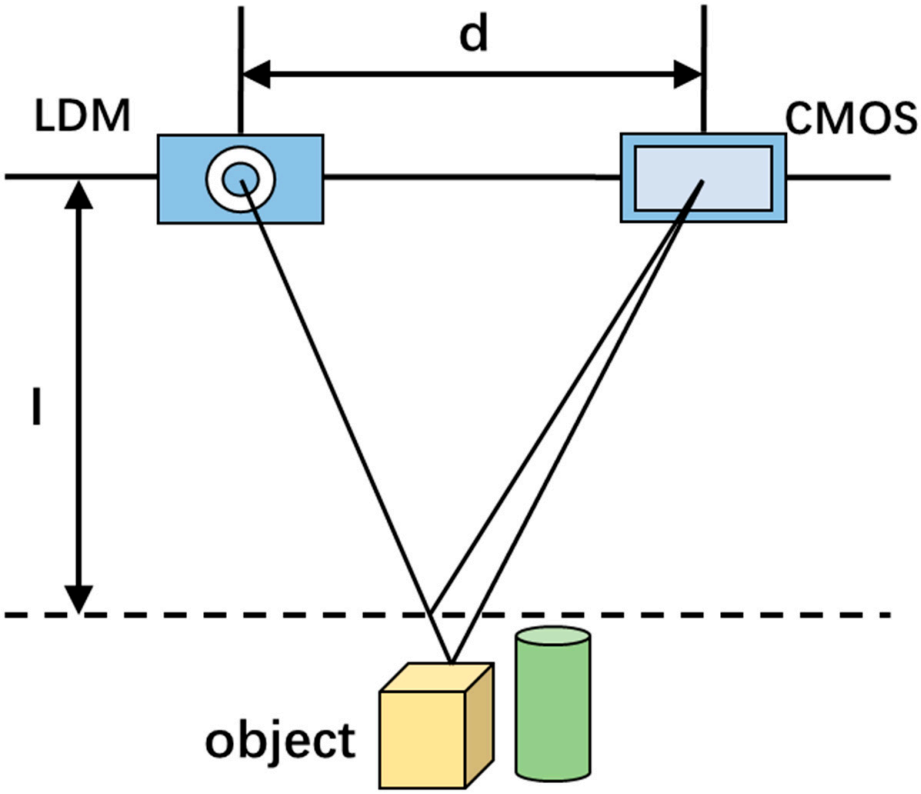


Figure 4. Schematic diagram of structural light. LDM: Laser Designator Module; CMOS: Complementary Metal Oxide Semiconductor; d: Base Line; l: Distance from the reference plane to the camera.

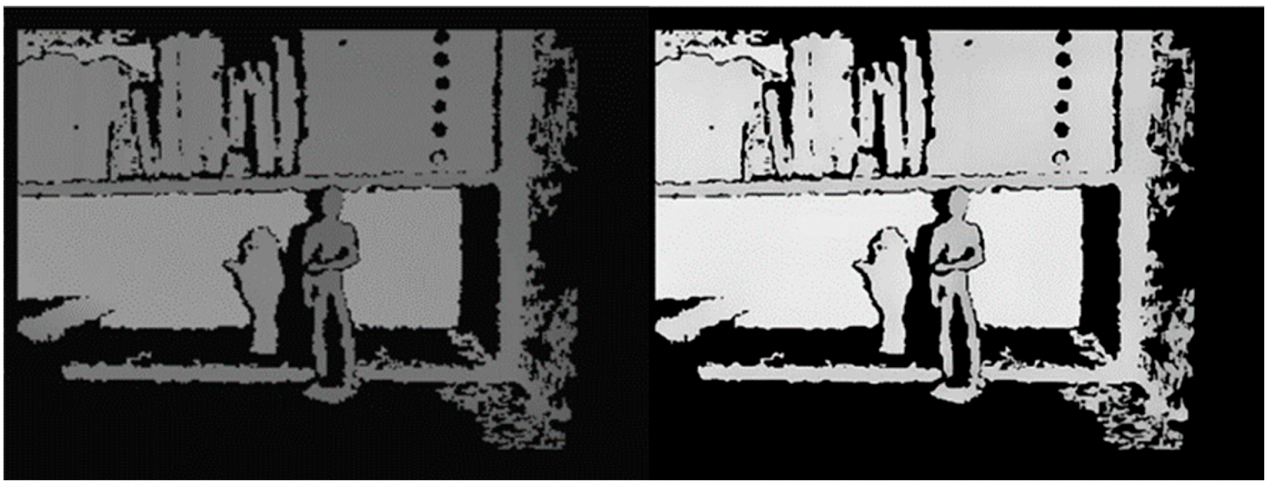


Figure 5. Before (**left**) and after (**right**) comparison of bilateral filtering.

2.7. Localization

2.7.1. Software Structure and Hardware Layout of Computer Vision System

The computer vision system in this study consists primarily of neural network object detection and depth algorithm spatial localization, as shown in Figure 6. And the hardware components and device communication of the target detection and localization system, as shown in Figure 7. RGB color images and RGBD depth images are acquired using an industrial camera and a depth camera, respectively. After the neural network model recognizes the objects and obtains the 2D image coordinates, these coordinates are transformed via a predefined callback function and applied to the environment point cloud obtained from the depth algorithm. This process allows us to obtain 3D spatial coordinates in the

camera coordinate system. After performing coordinate calculations and transformations, the coordinates are converted into 3D spatial coordinates in the robotic arm coordinate system. Subsequently, the coordinate information is wirelessly transmitted via WiFi to the robotic arm for picking.

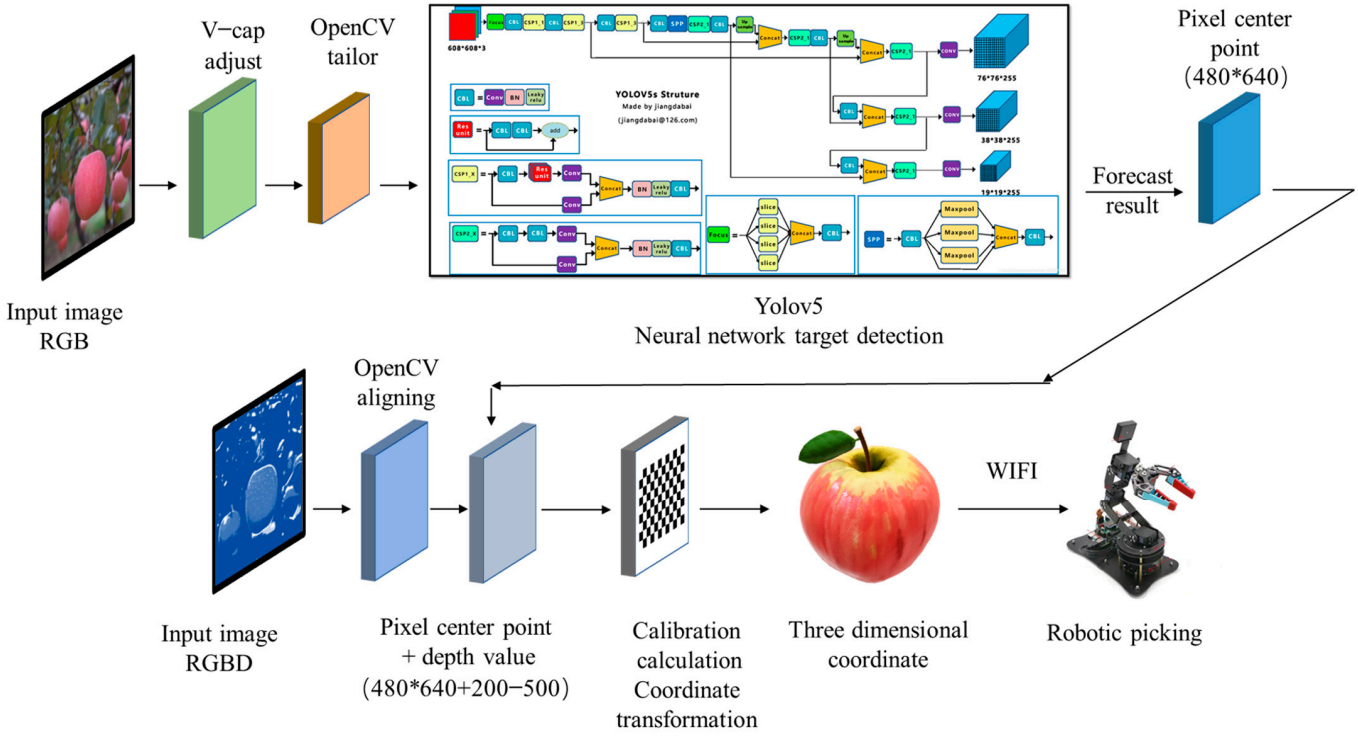


Figure 6. Software architecture of a computer vision system.

The core of this coordinate transformation lies in hand–eye calibration, a commonly used technique in the field of robotic vision. It is employed to achieve precise positioning and grasping of objects by a robotic arm. The goal is to convert the three-dimensional coordinates of a target detected in the camera coordinate system to the three-dimensional coordinates in the robotic arm's base coordinate system. This study adopts the "eye-in-hand" hand–eye calibration approach, selecting the center of the robotic arm's end effector as a reference point for calibration. Throughout the motion of the robotic arm, the camera coordinate system and the robotic arm's base coordinate system remain fixed, and their relative positions remain unchanged, resulting in a constant transformation matrix. Assuming the robotic arm's base coordinate system is denoted as {Base} and the camera coordinate system as {Camera}, if the coordinates of several fixed points (P) in both systems are known, the transformation matrix corresponding to the two coordinate systems can be obtained using the coordinate transformation formula. In this study, the "eye-in-hand" hand–eye calibration is employed, and the calibration process is carried out following the tutorial of the Moveit-easy_handeye package in the robot operating system (ROS). The calibration procedure involves launching relevant nodes for the robotic arm, Realsense camera, Aruco marker detection, and the easy_handeye node. The parameter "eye_to_hand" is set to true, indicating that the camera is positioned on the hand.

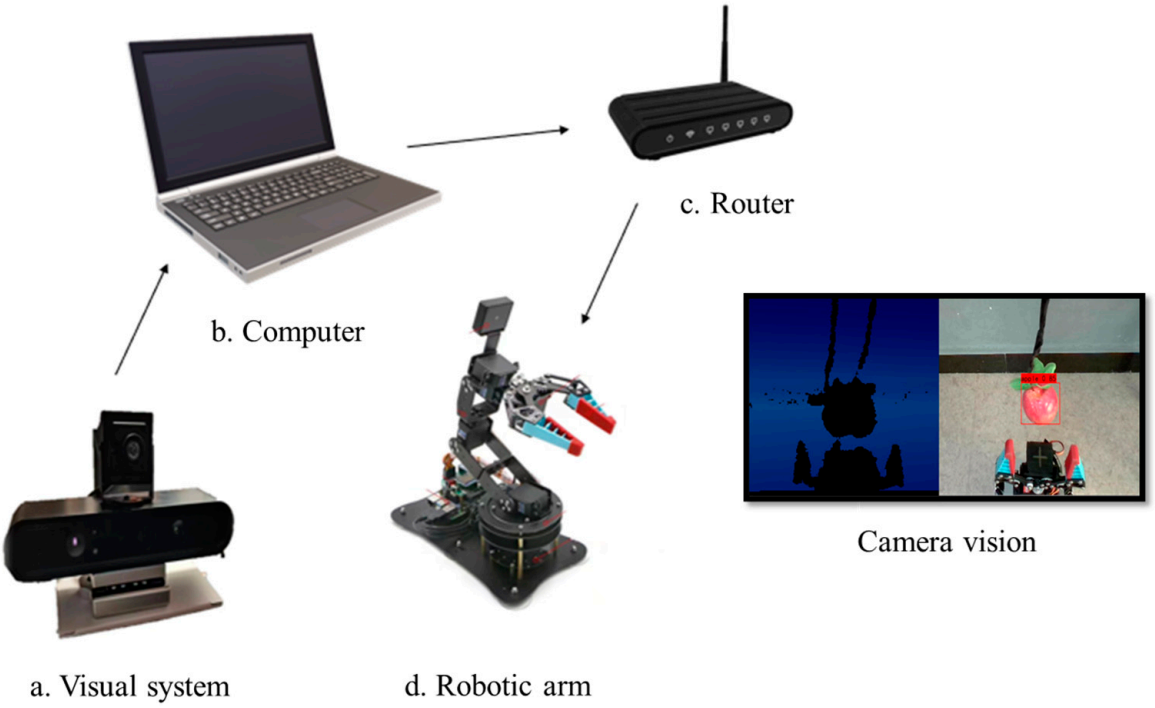


Figure 7. The hardware components and device communication of the target detection and localization system. The target detection and localization system in this study is mainly composed of neural network object detection and depth algorithm spatial positioning. RGB color image and RGBD depth image are obtained with an industrial camera and depth camera, respectively. After identifying the target and obtaining the two-dimensional image coordinates, the neural network model converts the coordinates into the environment point cloud image obtained by the depth algorithm through the callback function set, thereby obtaining the three-dimensional space coordinates in the camera coordinate system and converting the coordinates into the three-dimensional space coordinates in the robot arm coordinate system through calculation. It is then transmitted wirelessly via WiFi to send the coordinate information to the robotic arm for picking. The robotic arm used for apple picking is the A1501 robotic arm from WHEELTEC (Dongguan) Co., Ltd. (Room 814, Building 9, No. 1 Xuefu Road, Songshan Lake Park, Dongguan City, Guangdong Province, China). The servo motors employed are the high-torque digital servo motors from the S20 series, with dimensions of 275 × 160 × 483.34 mm^3. The weight of the robotic arm itself is 1.6 kg, and it is capable of grasping objects within the weight range of 200 to 300 g. The main control board utilized is the STM32_F103C8T6, and the operating system is ubuntu18.04+melodic. The external communication interface supports communication with an external controller through either CAN or serial communication.

2.7.2. Simulation Experiment Process for Computer Vision-Based Apple-Harvesting Robot Control

The simulation experiment is conducted to validate the proposed control method for apple harvesting. Models of ripe apples are suspended in the laboratory environment. The robot follows the control to perform apple harvesting. The process may include the following steps, as shown in Figure 8.

Experimental validation: The harvesting robot conducts simulated picking experiments following the control process. During the experiment, metrics such as the harvesting success rate, picking speed, and accuracy are recorded and evaluated to validate the feasibility and effectiveness of the proposed method in apple-harvesting tasks.

Through this simulation experiment, the performance and feasibility of the computer vision-based apple-harvesting robot control method can be assessed. Based on the experimental results, further optimization and improvement of the control method can be pursued to enhance the harvesting efficiency and accuracy of the robot.

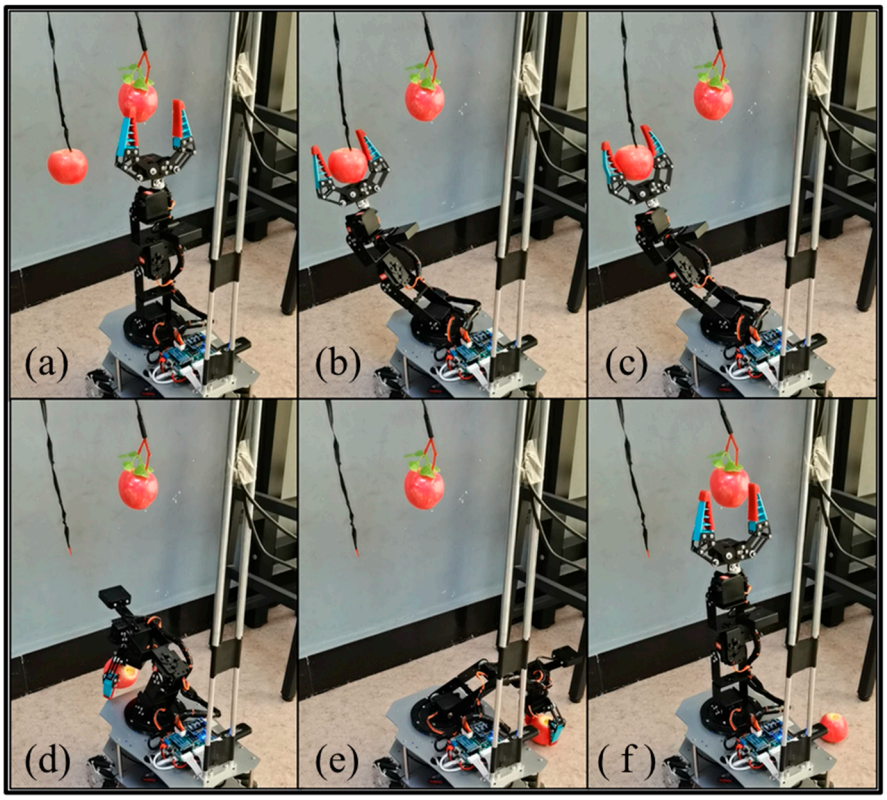


Figure 8. (**a**) Image acquisition: the robot uses a camera or sensors equipped with a computer vision system to capture images of the environment. (**b**) Object detection: through the application of object detection algorithms, the robot identifies and locates apple targets in the captured images. (**c**) Path planning: based on the target's location information, the robot performs path planning to determine the optimal harvesting path. (**d**) Motion control: the robot controls its movement according to path planning results, maneuvering towards the target position. (**e**) Harvesting operation: the robot executes the harvesting operation using a robotic arm or other harvesting devices to pick the apples. (**f**) Result assessment: the robot assesses the success of the picking operation and records the harvesting results.

2.8. Evaluation of Localization Accuracy

Different measurement distances, namely 100 mm, 200 mm, 300 mm, and 400 mm, were selected to test the spatial localization accuracy. Ten random test points were set for each distance, and the actual spatial coordinates of the test points (relative to the camera coordinates) were known. The depth camera was used to measure the spatial coordinates of the ten test points in each distance group. The computer vision errors were calculated for each axis, and the norm of the errors for each axis was calculated using Equation (6) to represent the computer vision error [26].

$$\| (X_e, Y_e, Z_e) \| = \sqrt{(X_e^2 + Y_e^2 + Z_e^2)} \tag{6}$$

In Equation (6), X_e, Y_e, and Z_e represent the average values of the positioning errors in the X-, Y-, and Z-axis directions, respectively, for each distance group. They can be calculated using Equation (7):

$$\begin{aligned} X_e &= \frac{\sum_{i=1}^{10} |e_{xj}|}{10} \\ Y_e &= \frac{\sum_{i=1}^{10} |e_{yj}|}{10} \\ Z_e &= \frac{\sum_{i=1}^{10} |e_{zi}|}{10} \end{aligned} \tag{7}$$

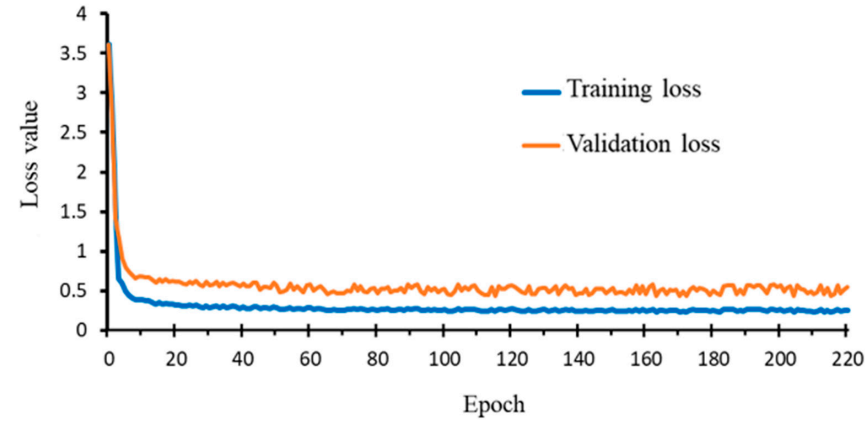
Here, e_{xj}, e_{yj}, and e_{zi} represent the positioning errors in each axis for each test sample.

3. Results

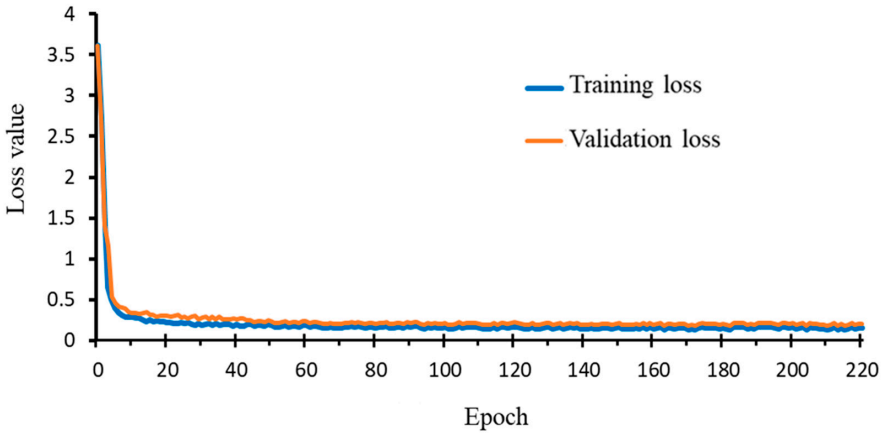
3.1. Training Assessment and Performance of the Network

The training loss curves of the YOLOv5 convergence are depicted in Figure 8. The horizontal axis "Epoch" represents the number of times the training dataset passes through the neural network and returns during the training process. The vertical axis "Loss value" indicates the discrepancy between the model's output and the actual values. The loss function serves as a measure of the quality of model predictions, where smaller loss values indicate the closer proximity of the model output to the ground truth. With increasing iterations, the loss values gradually decrease.

Figure 9a shows that before model optimization, the model overfits the validation set, leading to suboptimal performance. As shown in Figure 9b, after model optimization, the training and testing loss curves exhibit minimal fluctuations and align well, indicating that the model performs well on both the training and validation sets. After multiple epochs, YOLOv5 stabilizes at a loss value of around 0.30. The converging loss curve signifies YOLOv5's effectiveness in accurately identifying objects and performing apple image detection.



(**a**)



(**b**)

Figure 9. Training and validation loss curves: (**a**) before model optimization; (**b**) after model optimization. After model optimization, the training and testing loss curves exhibit minimal fluctuations and align well, indicating that the model performs well on both the training and validation sets. There are three main losses during training: rectangular frame loss (lossrect), confidence loss (lossobj), and classification loss (lossclc). The loss function of the yolov5 network is defined as Loss = a × lossobj + b × lossrect + c × lossclc. In other words, the total loss is the weighted sum of the three losses, and the confidence loss is usually the largest weight, followed by the rectangular frame loss and the classification loss.

3.2. Comparison between before and after Optimization

By introducing convolutional kernel groups and attention modules into the YOLOv5m network and applying transfer learning, significant performance improvements have been achieved in the object detection task. The convolutional kernel groups allow the model to extract apple features better and improve its ability to handle occluded apples in complex backgrounds. The introduction of attention modules further focuses on apple features, helping to reduce the impact of small-sized and irregularly shaped apples and thus improving the recall rate.

Using transfer learning, existing domain knowledge structures from auxiliary domains (such as the VOC2012 dataset) are transferred to the apple image recognition task. This approach improves model convergence and prevents overfitting. Through transfer learning, the model can learn general feature representations from the auxiliary domain and apply them to multi-object apple image recognition tasks, thereby enhancing the model's detection accuracy and recall rate.

The results of the ablation test are shown in Table 1. Data augmentation and preprocessing can improve network generalization and detection effects by a small margin. By using the convolution kernel set to enhance the feature extraction of apples, the recall rate is significantly improved, which is due to the complex background of the dataset and the large morphological changes in apples, and the convolution kernel set and attention module can effectively suppress their influence. The general attention module focuses on the whole of an apple, reduces the class imbalance between the foreground and the background, and reduces false detection. Therefore, the proposed method has fewer instances of false detection and missing detection and obtains good detection results. The test results show that the optimization method in this study improves the detection accuracy, recall rate, and mAP of the model, and the average accuracy of the method is increased by 5.75 percentage points compared with the baseline network.

Table 1. The results of ablation experiments.

Baseline Network	Augmentation	Preprocessing	Convolution Kernel Group	Attention Block	Recall	Precision	mAP/%
YOLOv5m					0.89	0.84	87.11
YOLOv5m	√				0.89	0.86	88.16
YOLOv5m	√	√			0.90	0.87	89.56
YOLOv5m	√	√	√		0.92	0.91	90.87
YOLOv5m	√	√	√	√	0.95	0.94	92.86

('√' indicates that this method is used).

3.3. Structured-Light Positioning Evaluation

The results of spatial positioning error for four sets of test distances are presented in Table 2. In the X and Y planes of four different depths, the errors in the X and Y directions fluctuate around 1 mm, which indicates that the determination of two-dimensional pixel coordinates based on a neural network is accurate and reliable, and the improved YOLOv5 model can adapt to the changes in apple size at different depths. In the Z direction, as the depth increases, the error reaches the minimum at 800 mm, about 3 mm. When the plane of the target is closer to the laser vision system, the error is larger, which is caused by the distance between the laser transmitter and the signal receiver. The experimental findings of this study indicate that the precision of apple-harvesting robot control based on computer vision is influenced by both the accuracy of the computer vision and the distance measurement of the depth camera. The primary contributor to spatial positioning error is observed in the Z-axis direction, specifically stemming from depth measurement inaccuracies of the depth camera. Within the operational range of the apple-harvesting robot, the error in computer vision-based apple recognition and localization is maintained at approximately 4 mm, which satisfies the requirements of practical applications.

Table 2. The results of spatial positioning.

Distance	Axial Measurement Error			Spatial Positioning Error
(mm)	X_e (mm)	Y_e (mm)	Z_e (mm)	(mm)
200	1.1	1.2	4.4	4.69
400	1.2	1.0	3.7	4.02
600	1.0	1.3	3.5	3.87
800	1.2	1.2	3.2	3.62

4. Discussion

Compared with the binocular positioning method based on the traditional image algorithm, the active laser positioning method based on deep learning has higher detection accuracy. Jiao et al. found the maximum value of the calculated minimum distance from the inner point to the edge. Finally, the radius of the apple was obtained by finding the minimum distance from the center to the edge. In this study, the maximum error of the apple center reached 23.21 mm [27]. Li et al. used the Faster R-CNN to detect binocular images of apples. Color difference and the color difference ratio were used to segment the detected apple in the boundary frame quickly, and the three-dimensional coordinates of the feature points were calculated. Finally, the average standard deviation of the positioning results of 76 datasets was 51 mm [28]. Chen et al. built a fruit recognition model based on a deep convolutional network, and spatially located the centroid of the fruit according to the local point cloud information on the fruit's surface [29]. Kang et al. introduced a vision perception and localization strategy based on LiDAR–camera fusion, and used a one-stage instance segmentation network to perform fruit localization [30]. Comparison with other methods in this study are presented in Table 3

Table 3. Comparison with other methods in this study.

References	Method	Data	Error
Jiao et al. [27]	Convert the color space to the lab color space for K-means segmentation. Obtain the contour of apples through morphological processing. Utilize an algorithm to calculate the distance between two points to find the center.	150 images	7.41–23.21 mm
Li et al. [28]	Detect binocular images using the Faster R-CNN. Utilize color difference and color contrast to quickly segment the detected apples in the boundary frames, and compute the three-dimensional coordinates of feature points.	76 datasets	51 mm
Chen et al. [29]	Built a fruit recognition model based on deep convolutional network, and spatially located the centroid of the fruit according to the local point cloud information on the fruit surface.	452 fruits in the crown of 10 fruit trees	15 mm
Kang et al. [30]	Introduced a vision perception and localization strategy based on LiDAR–camera fusion, and used a one-stage instance segmentation network to perform fruit localization.	Apple trees at 0.5 m, 1.2 m, and 1.8 m	0.5 m:2.5 mm; 1.2 m:2.3 mm; 1.8 m:2.8 mm
This study	Rely on a neural network model to detect targets by mapping two-dimensional pixel coordinates from images to a laser point cloud, thereby determining the position of apples.	40 test points	4 mm

In contrast to the above localization methods, the laser localization method in this study relies on neural network models to detect targets. The position of the apple was determined by substituting the 2D pixel coordinates in the image into the laser point cloud. The principle of this method is simple and the equipment cost is low. The laser localization method in this study relies on a neural network model to detect targets by mapping

the two-dimensional pixel coordinates from images into the laser point cloud, thereby determining the position of the apples. This method utilizes a neural network model that has been optimized through training. The recognition of targets is simpler compared to feature-matching algorithms with the same accuracy, as it does not require complex matching computations. For localization, the approach leverages point cloud information from laser sensors, which can be directly incorporated into coordinates, eliminating the need for processes like stereo vision disparity calculation. This simplification in localization enhances precision. In terms of cost, depth cameras equipped with laser sensors may be slightly more expensive compared to other types of cameras at a similar level. However, when considering the overall system cost, laser depth cameras do not require an additional higher-performance processor. In contrast, for instance, stereo cameras, due to their substantial initial data volume, necessitate a more powerful processor to avoid sluggishness in image computations.

Innovation in the apple spatial localization method using a laser depth camera involved: The use of a neural network model for object detection, incorporating two-dimensional image coordinates into the depth map from a laser depth camera, and employing a coordinate transformation algorithm for the spatial localization of target fruits constitute innovative aspects. In the experiments, the computer vision system demonstrated a spatial localization precision of approximately 4 mm, enabling the guidance of the picking gripper to achieve accurate positioning within the working space.

To propel the advancement of harvesting robots in sync with contemporary trends, avenues for innovation can be explored across various dimensions: Human–robot co-design emerges as a pivotal trajectory in the evolution of agricultural robotics. Especially in realms like target recognition, efficient picking, and remote control, the synergy between humans and robots through co-design possesses the potential to profoundly amplify the operational efficiency of harvesting robots. The integration with agricultural techniques assumes a crucial role, wherein the seamless fusion of robots with cutting-edge agricultural practices, encompassing standardized cultivation methods and orchard management, emerges as a decisive factor influencing the efficacy of harvesting robots. The confluence of robotic technology and agricultural techniques bears the capacity to exponentially augment their productivity in agricultural operations. Leveraging online deep learning object detection platforms proves indispensable. These platforms serve as reservoirs for real-time data garnered from harvesting robots, facilitating the creation of augmented datasets. Through a continuum of updates and iterative learning, the precision of target detection for harvesting robots can be significantly elevated. Visual servo mechanisms coupled with feedback control emerge as a strategy to heighten the precision of robotic grippers. This approach entails harnessing visual sensors to gauge and regulate the position and orientation of the robotic gripper, thus honing its control accuracy and stability.

In addressing false positives occurring outside the harvesting zone, besides incorporating laser ranging for assessment, this study also highlights the optimization of the initial anchor box size within the target recognition algorithm to reduce the detection of non-harvestable targets. Following this optimization, smaller targets, typically located beyond the canopy, are excluded from recognition as harvestable targets. When working in a specific position and completing the harvesting of all viable targets within that range, the harvesting robot can be relocated to the next position, facilitating more suitable harvesting for other targets. Due to constraints in experimental conditions, no research has been conducted on selective harvesting. The study solely assumes that all fruits within the working range of the robotic arm need to be harvested. Further investigation is required to develop a selective harvesting strategy for apples. Additionally, algorithm optimization is needed to address challenges such as fruit overlap and obstruction by branches and leaves. Adverse weather conditions also need to be considered. At present, we cannot guarantee with absolute certainty that the algorithm will function in all conditions or environments, as our current experiments have certain limitations. It is known that the algorithm can successfully perform target recognition and localization under clear weather conditions

when apples are mature and meet harvesting criteria. However, under overcast skies or low-visibility conditions, especially when apples are not sufficiently mature, and their color and size are not distinct, improvements tailored to specific conditions are required. This may involve adjustments to contrast between targets and the environment, as well as threshold modifications for target size, to enhance performance.

5. Conclusions

This study focuses on the development of an improved structured-light-based object detection and localization method using the YOLOv5 neural network model for an apple-harvesting robot. The integration of apple recognition and three-dimensional spatial localization was achieved. Experimental results reveal that the enhanced model achieved a recognition rate of 92.86%, marking an increase of 5.75 percentage points in average precision compared to the original model. During harvesting experiments, the average spatial localization accuracy for 40 test points was approximately 4 mm. This experimental result demonstrates that employing the localization method proposed in this study can achieve accurate robotic harvesting operations under simulated conditions.

Author Contributions: Conceptualization, W.-H.S.; methodology, Q.Z. and W.-H.S.; investigation, Q.Z.; resources, W.-H.S.; writing—original draft preparation, Q.Z.; writing—review and editing, Q.Z. and W.-H.S.; visualization, Q.Z. and W.-H.S.; supervision, W.-H.S.; project administration, W.-H.S.; funding acquisition, W.-H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 32101610.

Data Availability Statement: Data will be available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Koutsos, A.; Tuohy, K.M.; Lovegrove, J.A. Apples and Cardiovascular Health—Is the Gut Microbiota a Core Consideration? *Nutrients* **2015**, *7*, 3959–3998. [CrossRef]
2. Chu, P.; Li, Z.; Lammers, K.; Lu, R.; Liu, X. Deep Learning-Based Apple Detection Using a Suppression Mask R-CNN. *Pattern Recognit. Lett.* **2021**, *147*, 206–211. [CrossRef]
3. Zhang, Z.; Igathinathane, C.; Li, J.; Cen, H.; Lu, Y.; Flores, P. Technology Progress in Mechanical Harvest of Fresh Market Apples. *Comput. Electron. Agric.* **2020**, *175*, 105606. [CrossRef]
4. Kang, H.; Chen, C. Fast Implementation of Real-Time Fruit Detection in Apple Orchards Using Deep Learning. *Comput. Electron. Agric.* **2020**, *168*, 105108. [CrossRef]
5. Robin, C.; Lacroix, S. Multi-Robot Target Detection and Tracking: Taxonomy and Survey. *Auton Robot* **2016**, *40*, 729–760. [CrossRef]
6. Andriyanov, N.; Khasanshin, I.; Utkin, D.; Gataullin, T.; Ignar, S.; Shumaev, V.; Soloviev, V. Intelligent System for Estimation of the Spatial Position of Apples Based on YOLOv3 and Real Sense Depth Camera D415. *Symmetry* **2022**, *14*, 148. [CrossRef]
7. Lu, C.; Song, Y.; Wu, Y.; Yang, M. 3D Information Acquisition and Error Analysis Based on TOF Computational Imaging. *Hongwai Yu Jiguang Gongcheng/Infrared Laser Eng.* **2018**, *47*, 1041004. [CrossRef]
8. Li, F.; Li, Q.; Zhang, T.; Niu, Y.; Shi, G. Depth Acquisition with the Combination of Structured Light and Deep Learning Stereo Matching. *Signal Process. Image Commun.* **2019**, *75*, 111–117. [CrossRef]
9. Fu, L.; Gao, F.; Wu, J.; Li, R.; Karkee, M.; Zhang, Q. Application of Consumer RGB-D Cameras for Fruit Detection and Localization in Field: A Critical Review. *Comput. Electron. Agric.* **2020**, *177*, 105687. [CrossRef]
10. Qingchun, F.; Wei, C.; Jianjun, Z.; Xiu, W. Design of Structured-Light Vision System for Tomato Harvesting Robot. *Int. J. Agric. Biol. Eng.* **2014**, *7*, 19–26. [CrossRef]
11. Jiménez, A.R.; Ceres, R.; Pons, J.L. A Vision System Based on a Laser Range-Finder Applied to Robotic Fruit Harvesting. *Mach. Vis. Appl.* **2000**, *11*, 321–329. [CrossRef]
12. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
13. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep Learning—Method Overview and Review of Use for Fruit Detection and Yield Estimation. *Comput. Electron. Agric.* **2019**, *162*, 219–234. [CrossRef]
14. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. DeepFruits: A Fruit Detection System Using Deep Neural Networks. *Sensors* **2016**, *16*, 1222. [CrossRef] [PubMed]
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]

16. Chen, S.W.; Shivakumar, S.S.; Dcunha, S.; Das, J.; Okon, E.; Qu, C.; Taylor, C.J.; Kumar, V. Counting Apples and Oranges with Deep Learning: A Data-Driven Approach. *IEEE Robot. Autom. Lett.* **2017**, *2*, 781–788. [CrossRef]
17. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 431–440.
18. Bargoti, S.; Underwood, J. Deep Fruit Detection in Orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3626–3633.
19. Gao, F.; Fu, L.; Zhang, X.; Majeed, Y.; Li, R.; Karkee, M.; Zhang, Q. Multi-Class Fruit-on-Plant Detection for Apple in SNAP System Using Faster R-CNN. *Comput. Electron. Agric.* **2020**, *176*, 105634. [CrossRef]
20. Changyi, X.; Lihua, Z.; Minzan, L.; Yuan, C.; Chunyan, M. Apple Detection from Apple Tree Image Based on BP Neural Network and Hough Transform. *Int. J. Agric. Biol. Eng.* **2015**, *8*, 46–53. [CrossRef]
21. Fu, L.; Majeed, Y.; Zhang, X.; Karkee, M.; Zhang, Q. Faster R–CNN–Based Apple Detection in Dense-Foliage Fruiting-Wall Trees Using RGB and Depth Features for Robotic Harvesting. *Biosyst. Eng.* **2020**, *197*, 245–256. [CrossRef]
22. Wang, Z.; Walsh, K.; Koirala, A. Mango Fruit Load Estimation Using a Video Based MangoYOLO—Kalman Filter—Hungarian Algorithm Method. *Sensors* **2019**, *19*, 2742. [CrossRef]
23. Ji, W.; Pan, Y.; Xu, B.; Wang, J. A Real-Time Apple Targets Detection Method for Picking Robot Based on ShufflenetV2-YOLOX. *Agriculture* **2022**, *12*, 856. [CrossRef]
24. Gao, Y.; Mosalam, K.M. Deep Transfer Learning for Image-Based Structural Damage Recognition. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 748–768. [CrossRef]
25. Zhang, S.; Xie, M. Real-Time Recognition and Localization Based on Improved YOLOv5s for Robot's Picking Clustered Fruits of Chilies. *Sensors* **2023**, *23*, 3408. [CrossRef] [PubMed]
26. Adamides, G.; Katsanos, C.; Constantinou, I.; Christou, G.; Xenos, M.; Hadzilacos, T.; Edan, Y. Design and Development of a Semi-Autonomous Agricultural Vineyard Sprayer: Human–Robot Interaction Aspects. *J. Field Robot.* **2017**, *34*, 1407–1426. [CrossRef]
27. Jiao, Y.; Luo, R.; Li, Q.; Deng, X.; Yin, X.; Ruan, C.; Jia, W. Detection and Localization of Overlapped Fruits Application in an Apple Harvesting Robot. *Electronics* **2020**, *9*, 1023. [CrossRef]
28. Li, T.; Fang, W.; Zhao, G.; Gao, F.; Wu, Z.; Li, R.; Fu, L.; Dhupia, J. An Improved Binocular Localization Method for Apple Based on Fruit Detection Using Deep Learning. *Inf. Process. Agric.* **2023**, *10*, 276–287. [CrossRef]
29. Feng, Q.; Zhao, C.; Li, T.; Chen, L.; Guo, X.; Xie, F.; Xiong, Z.; Chen, K.; Liu, C.; Yan, T. Design and Test of a Four-Arm Apple Harvesting Robot. *Trans. Chin. Soc. Agric. Eng. (Trans. CSAE)* **2023**, *39*, 25–33. [CrossRef]
30. Kang, H.; Wang, X.; Chen, C. Accurate Fruit Localisation Using High Resolution LiDAR-Camera Fusion and Instance Segmentation. *Comput. Electron. Agric.* **2022**, *203*, 107450. [CrossRef]