

Article

Deep Learning-Based Oyster Packaging System

Ruihua Zhang, Xujun Chen, Zhengzhong Wan, Meng Wang and Xinqing Xiao * 

College of Engineering, China Agricultural University, Beijing 100083, China; zhangruihua1020@163.com (R.Z.); a1227727909@163.com (X.C.); w1005681290@163.com (Z.W.); wangm.coe@cau.edu.cn (M.W.)

* Correspondence: author. xxqjd@cau.edu.cn

Abstract: With the deepening understanding of the nutritional value of oysters by consumers, oysters as high-quality seafood are gradually entering the market. Raw edible oyster production lines mainly rely on manual sorting and packaging, which hinders the improvement of oyster packaging efficiency and quality, and it is easy to cause secondary oyster pollution and cross-contamination, which results in the waste of oysters. To enhance the production efficiency, technical level, and hygiene safety of the raw aquatic products production line, this study proposes and constructs a deep learning-based oyster packaging system. The system achieves intelligence and automation of the oyster packaging production line by integrating the deep learning algorithm, machine vision technology, and mechanical arm control technology. The oyster visual perception model is established by deep learning object detection techniques to realize fast and real-time detection of oysters. Using a simple online real-time tracking (SORT) algorithm, the grasping position of the oyster can be predicted, which enables dynamic grasping. Utilizing mechanical arm control technology, an automatic oyster packaging production line was designed and constructed to realize the automated grasping and packaging of raw edible oysters, which improves the efficiency and quality of oyster packaging. System tests showed that the absolute error in oyster pose estimation was less than 7 mm, which allowed the mechanical claw to consistently grasp and transport oysters. The static grasping and packing of a single oyster took about 7.8 s, and the success rate of grasping was 94.44%. The success rate of grasping under different transportation speeds was above 68%.

Keywords: oyster; deep learning; machine vision; grasping; packaging



Citation: Zhang, R.; Chen, X.; Wan, Z.; Wang, M.; Xiao, X. Deep Learning-Based Oyster Packaging System. *Appl. Sci.* **2023**, *13*, 13105. <https://doi.org/10.3390/app132413105>

Academic Editor: Andrea Prati

Received: 21 November 2023

Revised: 3 December 2023

Accepted: 7 December 2023

Published: 8 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Oyster is one of the largest cultivated shellfish in the world [1,2]. It not only has delicious meat but also provides unique health effects and medicinal value, making it a seafood treasure with high nutritional value [3]. With a deepening understanding of the nutritional value of oysters, oysters have gradually become high-quality and high-end seafood. To cater to the diversified demands of consumers and the market, raw edible oysters are positioned as a premium product entering the market [4]. At the present stage, the raw edible oyster processing and production line relies heavily on manual sorting and packaging, with a low degree of automation, which hinders the improvement of the efficiency and quality of oyster packaging. Packaging of sterilized oysters, as the final stage of the oyster production line, has very strict hygiene standards [5]. However, relying on manual operations not only is time-consuming and inefficient but also easily leads to secondary contamination and cross-contamination of oysters, which reduces the shelf life of raw edible oysters and leads to the wastage of oysters [6].

In the current upgrade and transformation of production lines for raw edible aquatic products, the introduction of intelligent and automated production lines has become a sign of smart agriculture, aiming to enhance production efficiency, technological standards, and hygiene safety [7]. With the increasing attention to food safety and hygiene, the demand for aseptic and uncontaminated production environments has become increasingly important. Mechanical arms are widely utilized in the agricultural production field

due to their advantages in automated production [8], precise operation [9], and strong adaptability. Mechanical arms can be applied in aseptic and nonpolluting unmanned production plants to ensure hygiene and safety in the food production process, reduce the risk of cross-contamination, improve production efficiency, and ensure product quality and reliability [10]. The introduction of machine vision technology can make the mechanical arm more autonomous and intelligent, which improves the efficiency and quality of the production line and reduces manual errors and waste [11,12]. Object detection is an important part of the machine vision field, enabling the mechanical arm to perform tasks such as target localization, object recognition, and attitude estimation based on the acquired visual information, thus accomplishing smarter and more flexible operations [13,14]. With the development of computer hardware and software, the application of deep learning techniques in the field of detection and grasping has been widely studied and explored [15–17]. A target detection method based on convolutional neural networks has achieved high-precision object detection in complex environments [12]. Deep learning-based algorithms enable robots to improve their grasping success rate through trial-and-error learning [17]. The visual recognition model based on deep learning has higher robustness and environmental adaptability [18] and has been diversified for applications in industry, agriculture, medicine, and other fields [19–21].

Therefore, this study combines deep learning algorithms, machine vision techniques, and mechanical arm control techniques to design an automated packaging system for linking raw edible oyster sterilization production lines. The system avoids damage and contamination of oysters due to improper manual operation and improves the efficiency and quality of the oyster packing line. The scheme of the oyster packing system is designed based on four aspects—application layer, functional layer, control layer, and hardware layer—and the software and hardware aspects of the system are designed and analyzed. Using deep learning object detection techniques to build an oyster visual perception model improves the accuracy and reliability of oyster identification and localization on the production line. The model is pruned and optimized not only to satisfy the detection accuracy [22] but also to significantly reduce the computational complexity of the model and reduce the latency which enables accurate and efficient oyster detection. Using the simple online and real-time tracking (SORT) algorithm [20], the grasping position of the oyster can be predicted, which enables dynamic grasping. This study proposes a visual touch line counting method for accurately counting the number of oysters on the production line.

A deep learning-based oyster packaging production line was designed and constructed, and an experimental plan was developed for system testing and functional verification. An oyster image dataset was created and used to train an oyster detection model. The model was pruned and optimized, and the results showed that the oyster detection model could balance speed and accuracy when the sparse rate was at 0.001 and the channel pruning ratio was 80%, which resulted in the best detection performance of the model. The number of parameters and computation of the model after pruning decreased by 86.4% and 61.3%, and the detection speed improved by 25.80%, which enabled the oyster intelligent perception model to accurately and efficiently realize the tracking and counting of oysters. The overall performance test results indicated that the absolute error of the oyster grasping position was no more than 7 mm, and the oyster could be grasped stably. The grasping and packing of a single oyster took about 7.8 s, and the success rate of static grasping was 94.44%. The success rate of grasping at different conveyor speeds was over 68%, which validated the effectiveness of the dynamic grasping and packaging function of the system.

The system can be applied to the packaging line of raw edible oysters to improve the efficiency and production capacity of the oyster production line through automated gripping and packaging, thus lowering labor costs and reducing errors and waste. At the same time, the system avoids cross-contamination caused by manual operation, which ensures the hygiene and safety of the production process of raw edible oysters, extending the shelf life of raw edible oysters and reducing losses.

2. Oyster Packaging System

2.1. Systematic Architecture Design

Intelligent, accurate, and sterile oyster packaging systems are a requirement for automated oyster production lines. To realize the automated grasping and packaging of the oyster production line, the oyster packaging system designed in this study mainly consists of four parts: application layer, function layer, control layer, and hardware layer (Figure 1).

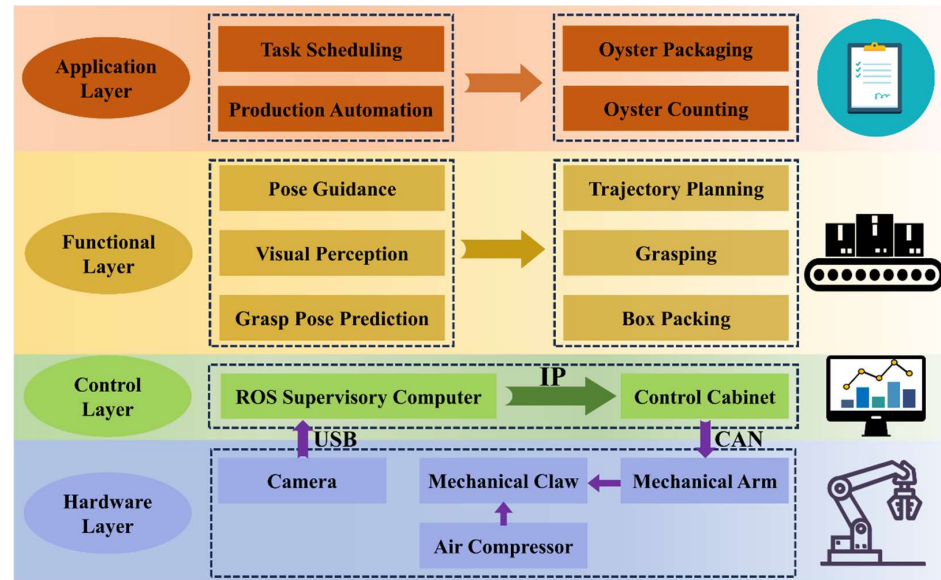


Figure 1. Intelligent machine vision model for oyster packaging.

The application layer identifies the production application aspects and task requirements of the oyster packaging system. This system realizes the grasping and counting function of oysters by linking with the oyster sterilizing production line, which realizes the automation of the oyster packaging production line. The functional layer predicts the oyster grasping position by a visual perception model, guides the position of the mechanical arm, and plans the trajectory for the oyster grasping to achieve the grasping and transfer of the oyster. The core of the functional layer is to establish the oyster intelligent perception model by deep learning machine vision algorithm, to obtain the position information of oysters being transported on the conveyor belt, and to realize the visual perception, tracking prediction, and counting function of oysters. By combining the mechanical arm control technology, the acquired oyster position information was mapped between the camera coordinate system and the robotic arm coordinate system, which was used to plan the running trajectory of the mechanical arm to grasp the oysters. The control layer consists of the robot operating system (ROS) and the control cabinet, which is mainly used to establish communication, coordinate, and send control commands for packaging tasks. Integrated deep learning algorithms, machine vision technology, and mechanical arm control technology in the ROS upper computer are used for the implementation of software algorithms and the transmission of control data. The control cabinet controls the operation of the mechanical arm and the opening and closing of the mechanical claw and plans the trajectory of the mechanical arm. Trajectory interpolation optimization was performed on the control cabinet, which together enabled control of the hardware layer. The hardware layer is based on the working characteristics of the oyster packaging production line; the relevant hardware of the packaging system is selected and constructed, including the mechanical arm, mechanical claw, and image acquisition equipment. The camera transfers the acquired image information to the ROS upper computer via the universal serial bus (USB). The mechanical arm receives motion commands from the control cabinet via the controller area network (CAN) for position adjustment.

2.2. Oyster Packaging System Design

Based on the requirements analysis and architectural design of the oyster packaging system, the system hardware was selected. The hardware of the system offered the whole product line a good basis for intelligent control. As shown in Figure 2, the deep learning-based oyster packaging system proposed in this study consists of a mechanical arm control module, an actuator module, and a visual perception module. The position of oysters on the conveyor belt is sensed by a visual sensor, and the mechanical arm is controlled under the ROS system to grasp the oysters and accurately place them in the freshness box to realize the automated packaging. The system can be integrated with an oyster production line to realize automated oyster grasping and packaging, which provides technical references for the construction of smart agriculture and unmanned workshops.

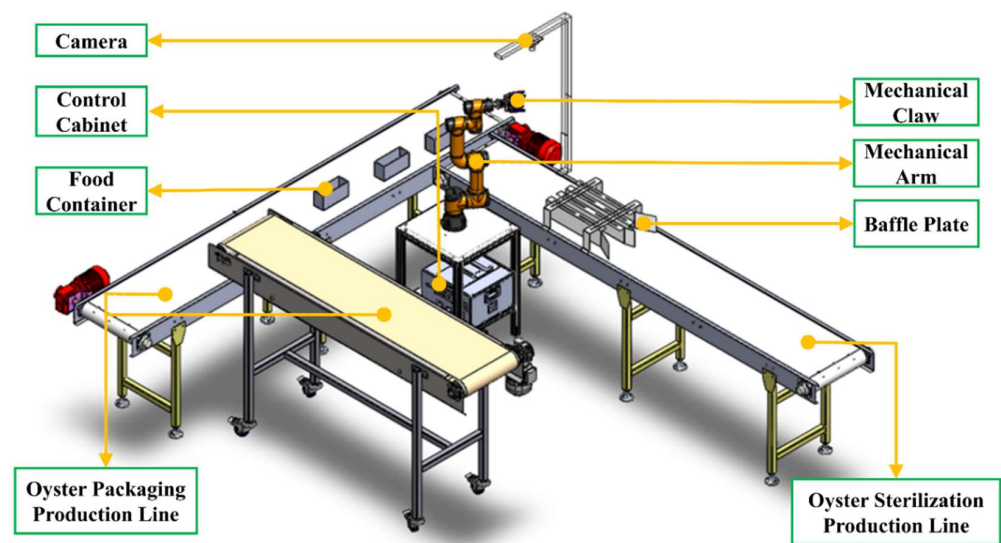


Figure 2. Design schematic of the oyster packaging system.

The mechanical arm control module consists of the mechanical arm and the control cabinet. The mechanical arm controller communicates with the upper computer via Ethernet and controls the motor at each joint to rotate at a certain angle according to the instructions issued by the upper computer to realize the motion control of the mechanical arm. The visual perception module mainly plays the role of perception and localization, and the image acquisition device is one of the important components of the visual localization module, which completes the transmission of image information in the ROS system. The image acquisition device is a USB camera, which is vertical to the conveyor belt and fixed on the bracket. The camera is a fixed-focus camera with a monitoring angle of 45 degrees, a resolution of 1920×1080 , and a frame rate of 30 frames/s. It uses a complementary metal-oxide-semiconductor (CMOS) sensor chip, and the data transmission between the camera and the upper computer is carried out via USB. The camera has the advantages of distortion-free lens, sensitive response, easy installation, and low price to meet the practical usage requirements of this system. The visual perception module can get the pixel coordinates of the oyster and convert the pixel coordinate system to the coordinate system of the mechanical arm, making the mechanical arm control the mechanical claw to grasp the oysters. The camera needs to continuously monitor and identify oysters on the conveyor belt, while the mechanical arm needs to reciprocate between the conveyor belt and the oyster preservation box. Therefore, this system installs the camera above the production line, which simplifies the complicated hand-eye calibration and camera calibration. It also enables the camera to monitor the production line in real time for greater efficiency. The position of the camera in this mounting method is relatively stable and is not easily affected by vibration or movement of the arm, which allows for more stable image information to be obtained. Due to the irregular shape of oysters, the end implementation module adopts

a soft gripper that performs the grasping task by means of pneumatic control. The soft gripper can firmly grasp and transfer oysters, which significantly improves the success rate of grasping. The system controls the mechanical claw to grasp oysters by moving the mechanical arm, and if the density of oysters scattered randomly is too high, two neighboring oysters too close to each other will cause interference during the grasping process. Therefore, by optimizing the layout of the production line and designing a baffle plate to guide various shapes of oysters on the conveyor belt in the same direction, the oyster grasp is not disturbed at this time—the smallest outer rectangle is close to the level and the best point of grasping is the center point of the target detection frame. Using this method, there is no need to consider the oyster grasping angle, and at the same time, the complexity of the model calculation is reduced, which enables faster oyster grasping. The main operation process of the oyster packaging line is as follows: the conveyor belt transports the sterilized oysters, which are adjusted by the baffle plate to a uniform position, ensuring that each oyster has a noninterfering gripping space. After the oyster passes through the baffle plate of adjustment and enters the field of view of the camera, the oyster perception module obtains the position information of the oyster and uploads it to the upper computer. The oyster perception model detects oysters within the field of view of the camera in real time. The upper computer collects the position information of the oyster, plans the optimal position for oyster grasping, and transmits the oyster grasping position information and the trajectory planning to the control cabinet. Under the ROS system, the mechanical arm moves according to the trajectory and grasps the oysters from the initial position to the grasping point, then transfers the oysters to the packaging box to complete the packaging task once.

This oyster packaging system is designed based on deep learning to realize the automated production of oyster packaging links. Intelligent machine vision perception algorithm, target tracking algorithm, and mechanical arm motion control technology were used to realize the intelligent perception and automated grasping and packing of oysters. The oyster production line was constructed in the actual production environment to test the feasibility of the system functions and validate the real-time performance and robustness of the system.

2.3. Deep Learning-Based Oyster Detection Model

To achieve intelligent grasping and packaging of oysters by the mechanical arm on the production line, it is necessary to construct a visual perception model to detect the oysters on the conveyor belt in real time and to realize oyster tracking, prediction of oyster grasping position, and oyster counting. Since the oyster visual perception model needs to be able to perform well on devices with average arithmetic and performance and, at the same time, to track and detect oysters moving on the conveyor belt in real time, the intelligent perception counting model should have the following characteristics:

- Low latency: the vision algorithm needs to be able to obtain the movement information of oysters in real time within the field of view of the camera.
- High detection accuracy and robustness: able to accurately recognize and locate oysters without leakage under different speeds of conveyor belts.
- Accurate tracking: able to realize the optimal prediction of oyster grabbing position and counting function.

The oyster perception model is constructed for real-time detection of oysters on the production line and obtaining the location information, so the recognition and localization of oysters is the core of the intelligent perception model. The rapid and accurate recognition and acquisition of oyster position information from the vision sensor are the prerequisites for the mechanical arm to be able to accurately complete the oyster grasping task. Traditional perception algorithms perform target detection and localization through traditional image processing algorithms such as binarization, threshold segmentation, and edge detection [23]. In the actual unmanned workshop production, the traditional image processing shortcomings such as poor real time, poor robustness, and the need to be

constantly adjusted for the detection of the object—oyster is an extremely irregular, black-and-white mixed shellfish—the traditional image processing algorithms are likely to lead to the target-missed detection. Deep learning systems, which can automatically discover features and patterns within a given dataset, simplify tasks such as pattern recognition and detection [24]. Therefore, the deep learning-based object detection algorithm makes it more generalizable by extracting more abstract and higher dimensional features of the oyster in the image and performs better in complex scenes.

In this study, oyster detection weights were trained as an oyster detection model based on the single-stage object detection model You Only Look Once version 5 (YOLOv5) [25,26]. Single-stage object detection does not require pre-generation of candidate frames, direct convolution of the image to extract features, classification, regression, and the output of the predicted location and class of the target [27]. Also known as regression-based object detection, the YOLO series is a typical representative of the algorithm based on regression detection. The YOLO series algorithms have been iteratively improved for a variety of scenarios due to their excellent detection speed and accuracy. The streamlined architecture and ease of deployment of YOLOv5 contribute to its outstanding performance in real-time object detection tasks. It strikes a fine balance between speed and precision, exhibits the capability to handle diverse environmental conditions, and demonstrates adaptability to various object classes. Its advanced performance across diverse benchmark datasets underscores its robustness in varied scenarios. The oyster intelligent perception model is applied to the moving production line, so there are high requirements for the computing speed and accuracy of the real-time detection model. Considering the detection scenario on the oyster production line, this study chose YOLOv5, which has the shallowest depth and the smallest width as the oyster detection model. The network structure of the YOLOv5 model is shown in Figure 3 and consists of the backbone, neck, and prediction.

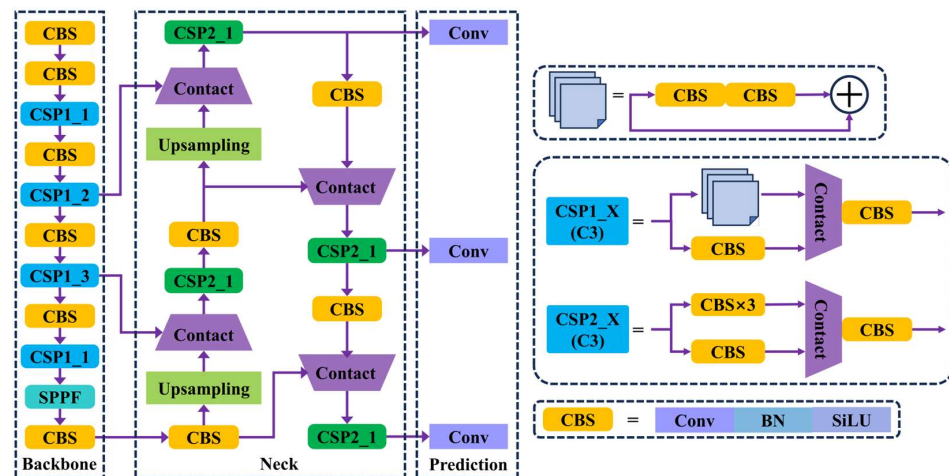


Figure 3. Deep learning-based oyster detection model.

The backbone section is mainly for oyster feature extraction. Focus was replaced with an equivalent Conv for better exportability, and the C3 structure was mainly used in the backbone feature extraction network. The C3 module is based on the cross-stage partial architecture for learning residual features from the input image. The structure is divided into two paths, one path goes through three convolutional layers and multiple bottlenecks, while the other path goes through only one convolutional layer, and finally, the two paths are connected. There is also a spatial pyramid pooling (SPP) module at the end of the feature extraction, which uses a cascade of multiple small-size pooling kernels instead of a single large-size pooling kernel in the SPP module, resulting in the preservation of the original functionality. This approach fuses the feature maps of different sensory fields and further improves the running speed part in the case of enriching the expressive ability of the feature maps.

The main purpose of the neck part of the model is to better utilize the oyster features extracted from the backbone for feature fusion and enhancement. Instead of blindly deepening the network, the residuals are no longer added to the C3 module. The neck part adopts a path aggregation network (PAN) structure, which improves the receptive field of the network and makes it more robust in oyster detection tasks by fusing the bottom- and top-level features and expanding the fused features by upsampling and downsampling the features. The prediction section of the model optimizes the predicted output by constructing multiple loss functions and NMS operations. The loss function is constructed to evaluate the difference between the true value and the prediction of the network. The loss function of this model consists of bounding box regression localization loss ($L_{box_{loss}}$), object classification loss ($L_{cls_{loss}}$), and confidence loss ($L_{con_{loss}}$). In target detection tasks, the regression loss typically uses the intersection over union (IOU) to calculate and label the coordinate distance between the true frame and the prediction frame of the network. In certain scenarios, there may be multiple possible ways for objects with the same IOU to intersect, making it difficult to describe how the ground truth boxes and predicted boxes intersect. Therefore, YOLOv5 adopts generalized intersection over union ($GIOU$) to describe the localization loss of the bounding box. The IOU and $GIOU$ are calculated as shown in Equations (1) and (2).

$$IOU = \frac{A \cap B}{A \cup B} \quad (1)$$

$$GIOU = \frac{A \cap B}{A \cup B} - \frac{|C - (A \cup B)|}{|C|} \quad (2)$$

where A represents the labeled true frame, B represents the predicted frame of the model, and C represents the minimally convex closed frame of A and B .

The $L_{cls_{loss}}$ evaluates the accuracy of the predicted category labels with respect to the true labels using binary cross entropy, which is shown in Equations (3) and (4).

$$L_{cls_{loss}} = \frac{\sum_{i \in pos} \sum_{j=cls} (O_{ij} \ln(\hat{c}_{ij} + (1 - O_{ij}) \ln(1 - \hat{c}_{ij})))}{N_{pos}} \quad (3)$$

$$\hat{c}_{ij} = \text{Sigmoid}(C_{ij}) \quad (4)$$

where O_{ij} represents whether the i -th prediction frame contains target j , C_{ij} represents the target prediction probability, $\hat{c}_{ij}$ represents the prediction confidence of C_{ij} calculated by the Sigmoid function, and N_{pos} represents the number of positive samples.

$L_{con_{loss}}$ is also calculated using binary cross entropy as shown in Equations (5) and (6).

$$L_{con_{loss}} = \frac{\sum_i O_i \ln(\hat{c}_i + (1 - O_i) \ln(1 - \hat{c}_i))}{N} \quad (5)$$

$$\hat{c}_i = \text{Sigmoid}(c_i) \quad (6)$$

where O_i represents the IOU of the prediction frame and the true frame, C_i represents the prediction frame, $\hat{c}_i$ represents the prediction confidence of C_i computed by the Sigmoid function, and N represents the total number of samples.

In the detection model, the three components are combined in a weighted sum to obtain the total loss function L_t , which is calculated as shown in Equation (7). The ADAM optimizer is introduced to optimize the network parameters to minimize the loss function and converge quickly while performing model training.

$$L_t = L_{box_{loss}} + L_{cls_{loss}} + L_{con_{loss}} \quad (7)$$

2.4. Pruning Optimization of Oyster Detection Model

The oyster perception model is embedded in a mobile platform that recognizes and counts oysters on the production line. Considering the limited memory and computational resources, there is a lightweight requirement for the detection model to reduce the detection latency and minimize the grasping error. Therefore, the oyster perception model was pruned and optimized to crop the model structure to have less impact on the output, have less computation and weight size, and be able to maintain the detection accuracy and increase the computational speed on low-computing-power devices. Pruning optimization can streamline the network and improve the iterative efficiency of the oyster tracking algorithm while satisfying the detection accuracy.

In this study, the optimized oyster perception model was used for practical test applications in oyster production lines. The oyster perception model is based on a deep learning single-stage target detection algorithm, which constructs a fast and accurate lightweight detection model by judging the importance of the channel and performing a pruning scheme. Combined with the tracking algorithm, it accomplishes the counting of oysters on the production line and the prediction of the grabbing position. The lightweight model can ensure better accuracy performance while continuously optimizing the network structure, reducing the amount of redundant calculations, and balancing the detection speed and accuracy.

Direct pruning of the model causes a significant drop in accuracy, so a criterion is needed to determine which channels in the model are less important, analyzing the dataset and model features together. This study performs channel pruning on the *BN* layer in the CNN and performs *L1* regular sparsification on the scaling factor γ of the output of the *BN* layer so that γ is close to zero, and the value of γ is used as a criterion to measure the importance of the channel. The *BN* layer is part of the conv module and mainly plays a role in speeding up the network training and convergence, doing bulk normalization of the convolution results. The convolutional and *BN* layers were fused in the YOLOv5 model of this study, and the *BN* layer was pruned without calling the fusion function during training and saving. The pruning reduces the complexity of operations on matrices, which in turn achieves the effect of streamlining the model and improving the speed of reasoning. The *BN* layer is calculated as shown in Equation (8).

$$Y_{bn} = BN_{out} = \frac{\gamma(x_{conv} - \mu)}{\sqrt{\sigma^2 + \varepsilon}} + \beta \quad (8)$$

where x_{conv} is the input to the convolutional layer, Y_{bn} is the corresponding output on that *BN* layer, μ and σ are the corresponding mean and variance on that layer, ε is a positive parameter that is not zero, β and γ are the normalization parameters of the *BN* layer, β is the bias compensation, and γ is the scale factor.

To sparsify the parameter γ while avoiding the overfitting phenomenon, it is first *L1* regularized. The regular term of parameter γ is summed with the loss function as the improved loss function for joint training, as shown in Equation (9). This allows the sparsification operation to determine the importance of the channel in preparation for subsequent pruning.

$$L = \sum_{(x,y)} l(f(x,w),y) + \lambda \sum_{\gamma \in \tau} g(s) \quad (9)$$

The first term of Equation (9) is the loss function for normal training of the convolutional neural network, where x is the training input, y is the training output, w is the parameter in training, and y is the training output. The second term is the added regularization constraint. where the function $g(s) = |s|$, and λ is the regularization factor, which is the sparsity rate.

Based on the evaluation criterion of channel importance, this study proposes a pruning optimization method for the oyster detection model, which includes four steps: base

training, L1 sparsity training, channel pruning, and model fine-tuning. The specific model optimization implementation process is shown in Figure 4.

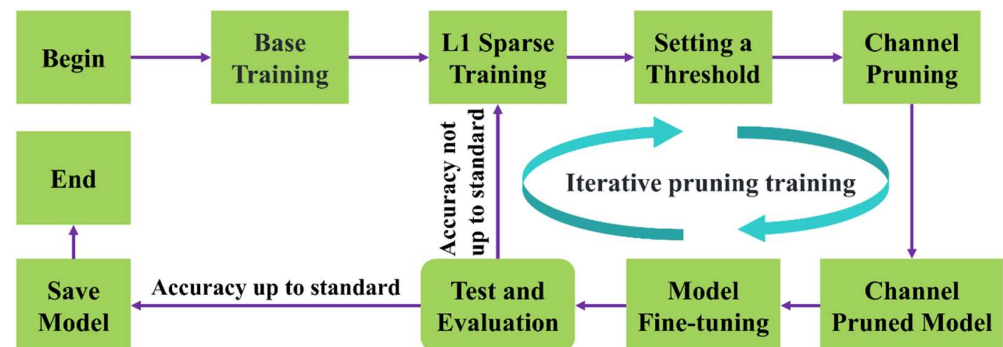


Figure 4. Pruning optimization of the oyster detection model.

The pruned and optimized model needs to continuously adjust the parameters according to the experimental results [28] and, after several iterative tests, obtain an oyster sensing model that can achieve the expected results in both detection speed and accuracy. The performance of the oyster detection model was tested under different pruning ratios to determine the optimal pruning ratio to ensure the efficiency and functionality of the perceptual model. We tested the performance of the original oyster detection model YOLOv5_O, the pruned oyster detection model YOLOv5_PO, and the oyster detection model YOLOv5_MO with the replacement of the backbone network on the test set. The results are shown in Table 1. The pruned oyster detection model was compressed to 3.99 MB in size with a 1.9% accuracy reduction. The number of parameters and computation decreased by 86.4% and 61.3%, and the speed of model detection increased by 25.80%. The significant reduction in model complexity in exchange for less accuracy loss is more favorable to the application of YOLOv5 in reality.

Table 1. Different oyster detection models.

Model	Model Size (MB)	Computational Speed (FPS)	MAP (%)	FLOPs (G)	Params (M)
YOLOv5_O	27.9	29	90.1	16.3	7,063,542
YOLOv5_MO	8.97	37	84.7	9.4	4,436,516
YOLOv5_PO	3.99	39	88.2	6.3	959,511

The test compares the performance of the model with different pruning ratios. The test results are shown in Figure 5. The pruned and optimized oyster perception model YOLOv5_PO is not only less complex than the model YOLOv5_MO that introduces a lightweight structure but is also better than the latter in terms of detection accuracy. The pruned and optimized oyster perception model not only reduces the model complexity but also maintains the same recognition ability as the original oyster detection model in real detection.

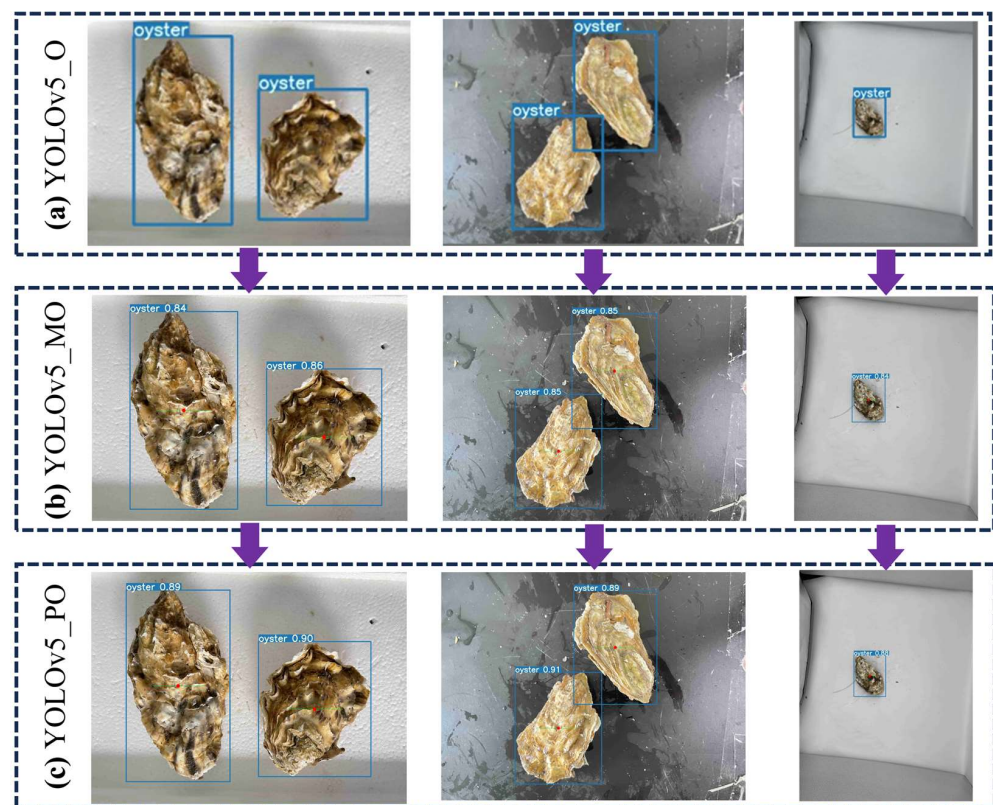


Figure 5. Performance evaluation of different oyster detection models.

2.5. Machine Vision-Based Oyster Counting Model

Oysters to be caught on the production line are in motion, so to catch the oysters dynamically, real-time tracking of the oysters is required to accomplish the grasping. The system employs a streamlined oyster detection model as the detector, combined with the SORT multi-target tracking algorithm, to achieve the counting of the number of oysters on the conveyor belt [29] and, at the same time, to track the oysters on the conveyor belt and predict the oyster grasping position, and, thus, to complete the algorithmic design of the dynamic interception of the grasping. Based on the movement speed and acceleration of the conveyor combined with the oyster movement slip error, the optimal grasping position in the direction of oyster movement is predicted. The SORT algorithm is a detection-based tracking algorithm and the performance of the detector plays an important role in the tracking effectiveness. The oyster tracking algorithm is designed using the improved deep learning oyster detection model as a detector, which has high accuracy and real-time performance for oyster tracking and detection. The flow of SORT conveyor oyster tracking in conjunction with the oyster detection model is shown in Figure 6.

The principle of the oyster tracking and detection model is shown in Figure 7. The oyster detection model was used to detect oysters in the camera field of view that were moving on the conveyor belt. The position of the oysters in the $n - 1$ th frame was detected (the two frames and their oyster grasping centroids in Figure 7a). Each blue detection window is a detector, and the position information of this picture is fed into the Kalman filter for predicting the information in the n th frame. The oyster detection model was used for the n th frame image to detect the positional information of the oysters (two orange windows in Figure 7b). The IOU between the oyster prediction frame and the oyster detection frame in the n th frame was calculated by Equation (1), and the cost matrix was built based on the IOU. Then the Hungarian algorithm [30] was used to determine the optimal result for the oyster association (two green windows in Figure 7b). Trackers that have not been matched beyond a certain number of frames were removed and new detection frames were added to the tracker to achieve the oyster tracking process.

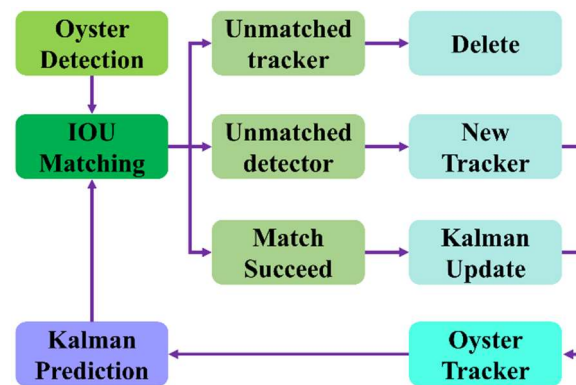


Figure 6. Machine vision-based oyster tracking and counting model.

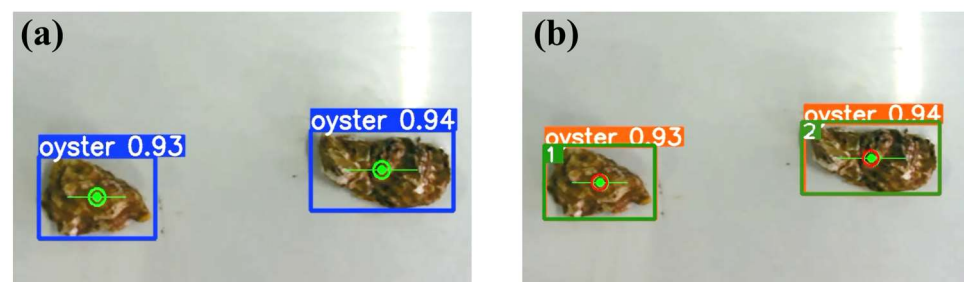


Figure 7. Schematic of oyster tracking detection model. (a) The detection image of the $(n - 1)$ th frame; (b) The detection image of the n th frame.

The automated packaging line for oysters needs to not only accurately grasp the oysters but also count the oysters on the conveyor belt to reduce the workload and error rate of manual inspection. Based on the multi-target tracking algorithm to realize the visual counting of oysters, the specific idea is that the conveyor belt movement direction must be from right to left, and a virtual yellow straight line must be marked in the camera field of view for judgment. It is important to determine whether the line connecting the oyster center point position predicted by the target tracking model and the oyster center point position detected by the target detection model intersects with the virtual yellow straight line and, also, to determine whether the direction of the vector composed of the two center points is the same as the direction of the conveyor belt. If both conditions are satisfied, it can be judged that the oyster passed through the virtual straight line and the count is increased by one. The SORT multi-target tracking algorithm was used to solve the problem of repeated counting during real-time detection by optimally matching the targets in neighboring frames, which avoids incorrect counting when the conveyor is transporting multiple oysters across the infrared counting sensor at the same time. The oyster tracking and counting model was tested on the actual oyster packing production line, and the counting effects of the model were observed on the upper computer, which is shown in Figure 8. The oyster tracking and counting model achieves better tracking results for multiple oysters, and whenever an oyster is passing through the virtual yellow line, the oyster is counted in real time, and the number of oysters counted is increased by one.

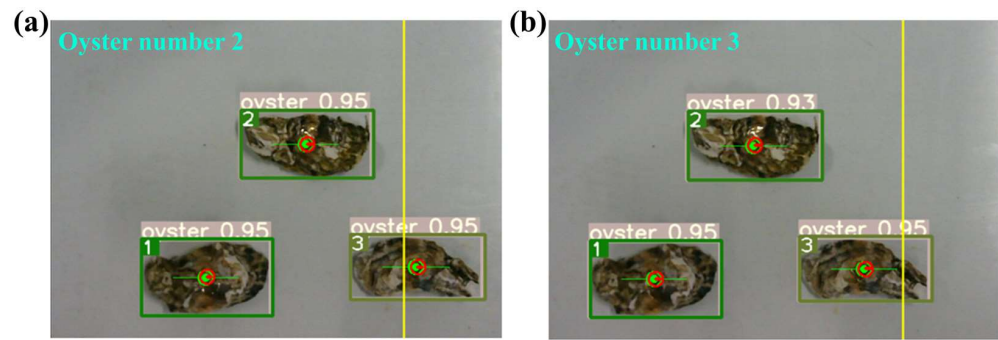


Figure 8. Schematic of oyster tracking count model. (a) Two oysters passed over the yellow line and the oyster count is 2. (b) The third oyster passed over the yellow line and the oyster count is increased by one.

2.6. Dataset Building and Model Training

The oyster dataset contains 500 high-resolution oyster images taken by mobile phones. The dataset was expanded to 1500 images by employing random flipping, blurring, random brightness, cropping, panning, and mirroring of the images (Figure 9a). This approach enhances the robustness of the oyster dataset and prevents overfitting of the oyster detection model. The dataset was manually labeled using LabelImg and saved as XML files (Figure 9b) and randomly divided into a training set and a test set in the ratio of 80% and 20%.

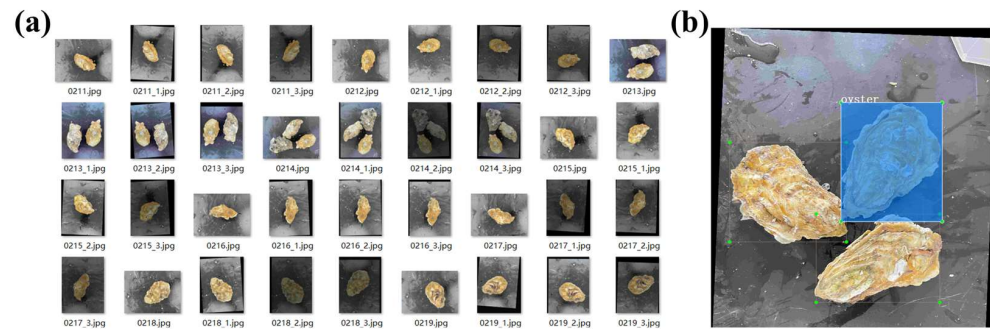


Figure 9. Oyster image dataset. (a) Expanded oyster image dataset; (b) Labeling the oyster dataset.

The evaluation metrics of a target detection model measure the detection performance of the model and guide the optimization of the model. Commonly used evaluation metrics mainly include the detection accuracy and inference speed of the model. The main metrics for model detection accuracy assessment are *Average Recall*, *Average Precision*, and mean *Average Precision* (*mAP*) across all classes. The *Average Recall* and *Average Precision* are calculated as shown in Equations (10) and (11). The *mAP* is the most comprehensive and objective index for evaluating the model accuracy in the field of target detection, and its calculation is shown in Equation (12).

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$mAP = \sum_{n=1}^n p(r_{n+1})(r_{n+1} - r_n) \quad (12)$$

where *TP* denotes the number of positive samples predicted correctly. *FP* denotes the number of negative samples predicted correctly. *FN* denotes the number of nondetected

targets. r_{n+1} denotes the threshold for recall. n denotes the number of recall thresholds. $p(r_{n+1})$ denotes the precision. $r_{n+1} - r_n$ denotes the recall.

The metrics for evaluating the speed of model inference include the model parameters, the floating-point operations (FLOPs), and the detection speed. The model parameter number parameter is the sum of all the parameters of the model, which reflects the spatial complexity of the algorithm and is calculated as shown in Equation (13). The number of floating-point operations FLOPs is the number of floating-point operations in the model, which reflects the time complexity of the algorithm and is calculated as shown in Equation (14).

$$para = c_{in} \times c_{out} \times k^2 \quad (13)$$

$$FLOPs = 2 \times c_{in} \times c_{out} \times k^2 \times M^2 \quad (14)$$

where k is the height of the convolution kernel (width and height are the same, both are k); c_{in} is the number of channels of the input image; $count$ is the number of channels of the output image; and M denotes the size of the output image (image length and width are the same, both are M).

The model was trained and tested with an NVIDIA GeForce GTX1050 GPU (NVIDIA Corporation, Santa Clara, CA, USA) computing platform with 2 GB of video memory. Python 3.9, PyTorch 1.8, and CUDA 11.1 were chosen as the framework for running the model. The initial learning rate was set to the default of 0.01, batch size was set to 4, the input image size was 640×640 , workers were set to 2, and the initial number of model epochs was 200, and it was set to 100 epochs for sparse training. The mosaic data enhancement strategy was used for training, where four oyster images were spliced in a randomized combination and cropped to increase the number of data samples. The variations of the classification loss (cls_{loss}), bounding box regression loss (box_{loss}), and objectness loss (obj_{loss}) during the model training process are shown in Figure 10. The model loss decreases noticeably during the first 40 epochs of training, and shows no significant fluctuation at around 140 epochs, indicating that the model has converged.

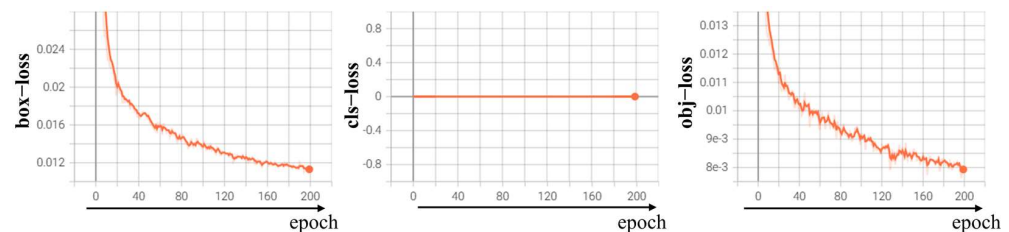


Figure 10. Loss function variation image for oyster detection model.

3. Experiments and Results

After designing and selecting each module, the communication between each module was established and integrated into a complete oyster packaging system based on deep learning, and the physical diagram of the system after the hardware connection of each module was completed as shown in Figure 11. The oyster automated packaging system constructs an oyster detection model based on the deep learning target detection algorithm and the lightweight technology of pruning and composes an oyster intelligent perception model with the oyster tracking and counting model to realize the intelligent perception and counting tasks for oysters on the production line.

Dynamic grasping tests were conducted on a single oyster on the automated production line. After the pre-planning was completed, the oyster grasping tests were conducted on the actual production line, and the whole grasping test process is shown in Figure 12. Oysters on the production line were detected with the deep-learning oyster detection model, and then, the oysters were quickly tracked and localized by the tracking and counting model. The test results showed that the model can accurately recognize and track

oysters when they are moving, and Kalman filtering can accurately predict oyster location information. The control cabinet guided the mechanical arm to reach the predicted grasping position and achieved accurate grasping and smooth transferring of oysters, which completed the oyster packaging task.

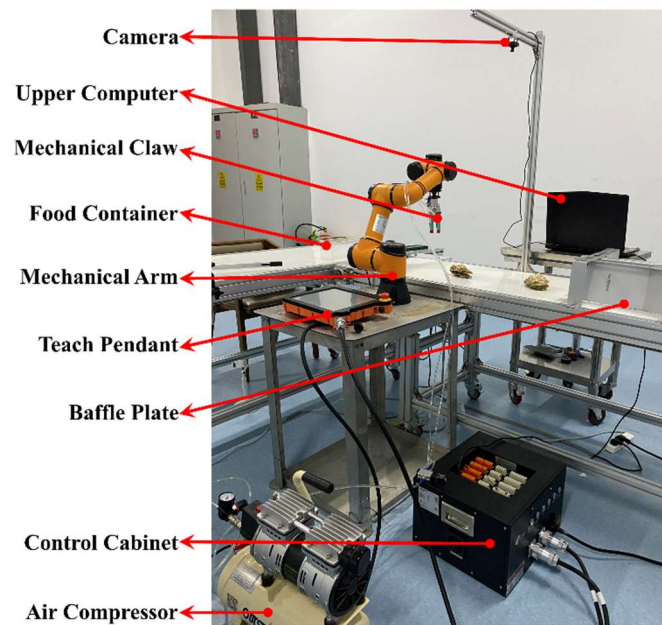


Figure 11. Components of the oyster packaging system.

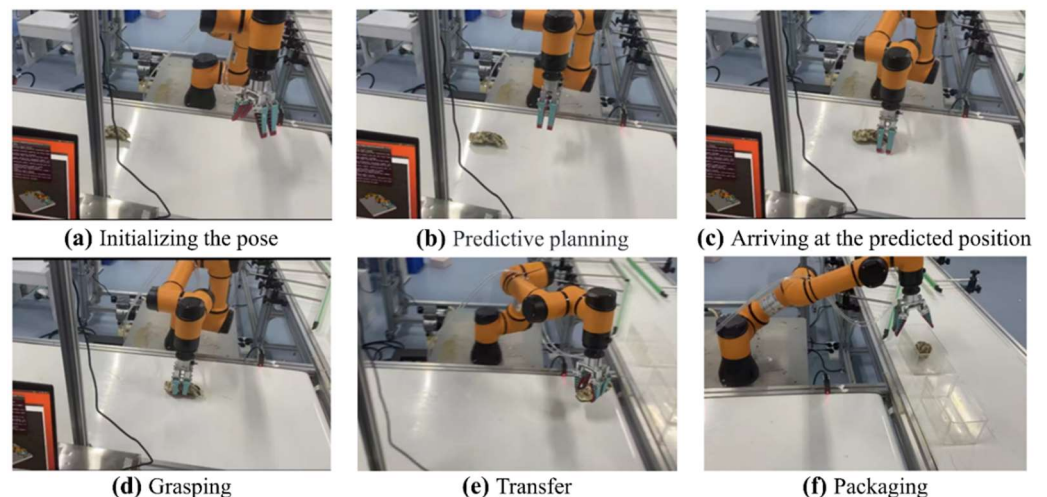


Figure 12. Flowchart of the oyster grasping process.

The accuracy of oyster pose estimation was tested with the system. Oysters were randomly placed on the conveyor belt, ensuring that the camera was fixed and the relative position to the mechanical arm remained unchanged. The coordinates of the oysters relative to the mechanical arm, the coordinates of the oysters detected by the recognition and localization algorithm, and the transformed coordinates by the mapping matrix were recorded. The results of the six sets of tests are shown in Table 2. The maximum opening and closing distance of the mechanical claw used in this study is 120 mm, and the absolute error of the oyster position estimation is not more than 7 mm, which is within the permissible error range of the mechanical claw.

Table 2. Results of oyster pose estimation.

Group	Actual Coordinates (x, y)	Localized Coordinates (x, y)	Transformed Coordinates (x, y)	Error (mm)	Absolute Error (mm)
1	(−0.1676, −0.5029)	(301, 246)	(−0.1688, −0.5013)	(1.2, 1.6)	2.00
2	(−0.2441, −0.5707)	(400, 155)	(−0.2460, −0.5657)	(1.9, 5.0)	5.35
3	(−0.2262, −0.5501)	(379, 169)	(−0.2297, −0.5559)	(3.5, 5.8)	6.77
4	(−0.3095, −0.3732)	(500, 420)	(−0.3148, −0.3743)	(5.3, 1.1)	5.41
5	(−0.1705, −0.3867)	(312, 407)	(−0.1730, −0.3855)	(2.5, 1.2)	2.77
6	(−0.2536, −0.5212)	(414, 224)	(−0.2548, −0.5160)	(1.2, 5.2)	5.33

The system was tested for oyster dynamic and static grasping. Six sets of static grasp tests were conducted for oysters, with six oysters randomly placed in each set. The observations were made whether the oyster would be adjusted to a better grasping position after passing through the baffle plate for position adjustment and whether the conveyor belt stopped running when the oyster passed through the infrared sensor. When an oyster reached the infrared sensor sensing range, the conveyor stopped running, the mechanical claw grasped the oyster, which triggered the infrared sensor, and then the conveyor continued to run. The number of successful oyster grasps and the time taken for the six groups of oyster grasps were recorded. The results of the experiments are shown in Table 3. In the six rounds of grasping experiments conducted, there were only two grasping failures, and in the two grasping failures, it was the oyster that slipped during grasping. Comprehensively analyzing the results of the six rounds of grasping experiments, the average success rate of grasping was 94.44%. The static grasping experiments were conducted on oysters in six rounds, for a total of 36 grasps, and the total time taken for grasping in each round was recorded. The average time taken for static grasping and packing of a single oyster was calculated to be about 7.8 s.

Table 3. The static grasping results of oysters.

Group	Total	Success Count	Success Rate (%)	Duration (s)
1	6	6	100	46.3
2	6	6	100	48.5
3	6	5	83.33	46.1
4	6	5	83.33	45.9
5	6	6	100	47.2
6	6	6	100	47.6

For the oyster dynamic grasping test, an oyster was placed on the top of the conveyor belt, and the time delay generated by the mechanical arm movement for grasping was recorded. Twenty-five grasping tests were performed at different belt speeds, and the number of successful grasps was recorded. The Kalman filter in the oyster visual perception model can accurately predict the state information of the oyster. The tests have shown that only ten frames of iterative updating are needed to get accurate position information for oyster position information. The mechanical claw was chosen to allow for some deviation in the oyster grasping position so that the whole system had a high grasping success rate. The success rate of the system in grasping oysters at three conveyor speeds was tested in the range of 2–7 cm/s speeds, 25 times at each speed, and the results of the grasping experiments are shown in Table 4.

Table 4. The dynamic grasping results of oysters.

Conveyor Speed (cm/s)	Total	Success Count	Failure Count	Success Rate (%)
2	25	24	1	96
5	25	21	4	84
7	25	17	8	68

The grasping success rate of the system is more than 84% when the running speed of the conveyor belt is less than 5 cm/s. The success rate of oyster grasping decreased when the running speed was higher than 5 cm/s. This was due to the limited camera field of view, which led to abnormal trajectory planning when the oyster passed quickly, causing the prediction position of the oyster grasping by the system to exceed the workspace.

4. Conclusions

This study combines the actual needs of the raw oyster production line, from the application layer, the functional layer, the control layer, and the hardware layer to designing the scheme of the oyster automated cartoning system and building the hardware of the system. An oyster automated packaging system was designed and constructed by using a vision detection algorithm and target tracking algorithm, combined with ROS mechanical arm control technology, to realize the automated grasping and packaging of fine raw oysters, to ensure the efficiency and quality of packaging oysters. A YOLOv5-SORT oyster intelligent perception model was constructed based on a deep learning target detection algorithm, which realizes real-time oyster detection, oyster tracking and counting, and the oyster grasping position prediction. By employing the lightweight technique of pruning, the complexity of the model was reduced, and the network was streamlined to improve the computational efficiency of the oyster detection algorithm under the premise of satisfying the detection accuracy. The accuracy and speed of target detection were greatly improved, with higher robustness and strong environmental adaptability. During static grasping, the mechanical arm was able to complete oyster grasping according to the formulated grasping sequence strategy, with an average grasping success rate of 94.44% and an average time consumed for each grasping task of 7.8 s. The dynamic gripping test of the oyster was carried out at three speeds with the conveyor running to verify the function of dynamic grasping of the mechanical arm. The system was tested to have a reasonable structural layout, work efficiently, and a high success rate of automated gripping and packaging.

The deep learning-based automated oyster packaging system designed in this study improves the efficiency and production capacity of the production line, reduces labor costs, and cuts down on errors and waste through automation. It also avoids cross-contamination caused by manual operation and ensures the hygiene and safety of the production process of raw edible oysters, which extends the shelf life of raw edible oysters and reduces losses. The intelligent perception model established in this system has a high degree of convergence, which can add more visual perception models of other aquatic products to realize the intelligent detection of a wider range of raw edible aquatic products and be integrated into the automated packaging production line of other raw edible aquatic products.

Author Contributions: Conceptualization, R.Z. and X.X.; Methodology, Z.W.; Software, R.Z., X.C. and M.W.; Validation, X.C.; Investigation, X.C.; Data curation, Z.W.; Writing—original draft, R.Z.; Writing—review & editing, M.W.; Supervision, X.X.; Funding acquisition, X.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: This research is supported by the 2115 talent development program of China Agricultural University.

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

References

1. Botta, R.; Asche, F.; Borsum, J.S.; Camp, E.V. A review of global oyster aquaculture production and consumption. *Mar. Pol.* **2020**, *117*, 103952. [\[CrossRef\]](#)
2. Mizuta, D.D.; Wikfors, G.H. Seeking the perfect oyster shell: A brief review of current knowledge. *Rev. Aquac.* **2019**, *11*, 586–602. [\[CrossRef\]](#)
3. Negara, B.F.S.P.; Mohibbullah, M.; Sohn, J.; Kim, J.; Choi, J. Nutritional value and potential bioactivities of Pacific oyster (*Crassostrea gigas*). *Int. J. Food Sci. Technol.* **2022**, *57*, 5732–5749. [\[CrossRef\]](#)
4. Felici, A.; Vittori, S.; Meligrana, M.C.T.; Roncarati, A. Quality traits of raw and cooked cupped oysters. *Eur. Food Res. Technol.* **2020**, *246*, 349–353. [\[CrossRef\]](#)
5. Li, Y.; Zhang, L.; He, Y.; Zhang, X.; Liu, X. Intelligent pulsed ultraviolet c radiation sterilization system: A cleaner solution of raw ready-to-eat aquatic products processing. *J. Clean. Prod.* **2023**, *427*, 139281. [\[CrossRef\]](#)
6. Liu, P.; Zhang, L.; Li, Y.; Feng, H.; Zhang, X.; Zhang, M. rGO-PDMS Flexible Sensors Enabled Survival Decision System for Live Oysters. *Sensors* **2023**, *23*, 1308. [\[CrossRef\]](#)
7. Li, C.E.; Tang, Y.; Zou, X.; Zhang, P.; Lin, J.; Lian, G.; Pan, Y. A Novel Agricultural Machinery Intelligent Design System Based on Integrating Image Processing and Knowledge Reasoning. *Appl. Sci.* **2022**, *12*, 7900. [\[CrossRef\]](#)
8. Bechar, A.; Vigneault, C. Agricultural robots for field operations: Concepts and components. *Biosyst. Eng.* **2016**, *149*, 94–111. [\[CrossRef\]](#)
9. Caldera, S.; Rassau, A.; Chai, D. Review of Deep Learning Methods in Robotic Grasp Detection. *Multimodal Technol. Interact.* **2018**, *2*, 57. [\[CrossRef\]](#)
10. Paradkar, V.; Raheman, H.; Rahul, K. Development of a metering mechanism with serial robotic arm for handling paper pot seedlings in a vegetable transplanter. *Artif. Intell. Agric.* **2021**, *5*, 52–63. [\[CrossRef\]](#)
11. Baduge, S.K.; Thilakarathna, S.; Perera, J.S.; Arashpour, M.; Sharafi, P.; Teodosio, B.; Shringi, A.; Mendis, P. Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. *Autom. Constr.* **2022**, *141*, 104440. [\[CrossRef\]](#)
12. Lv, Z.; Chen, T.; Cai, Z.; Chen, Z. Machine Learning-Based Garbage Detection and 3D Spatial Localization for Intelligent Robotic Grasp. *Appl. Sci.* **2023**, *13*, 10018. [\[CrossRef\]](#)
13. Cinal, M.; Sioma, A.; Lentz, B. The Quality Control System of Planks Using Machine Vision. *Appl. Sci.* **2023**, *13*, 9187. [\[CrossRef\]](#)
14. Benbarrad, T.; Salhaoui, M.; Kenitar, S.B.; Arioua, M. Intelligent Machine Vision Model for Defective Product Inspection Based on Machine Learning. *J. Sens. Actuat. Netw.* **2021**, *10*, 7. [\[CrossRef\]](#)
15. Feng, X.; Jiang, Y.; Yang, X.; Du, M.; Li, X. Computer vision algorithms and hardware implementations: A survey. *Integr. VLSI J.* **2019**, *69*, 309–320. [\[CrossRef\]](#)
16. Ku, Y.; Yang, J.; Fang, H.; Xiao, W.; Zhuang, J. Deep learning of grasping detection for a robot used in sorting construction and demolition waste. *J. Mater. Cycles Waste Manag.* **2021**, *23*, 84–95. [\[CrossRef\]](#)
17. Ribeiro, E.G.; de Queiroz Mendes, R.; Grassi, V. Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation. *Robot. Auton. Syst.* **2021**, *139*, 103757. [\[CrossRef\]](#)
18. Song, Y.; Gao, L.; Li, X.; Shen, W. A novel robotic grasp detection method based on region proposal networks. *Robot. Comput. Integr. Manuf.* **2020**, *65*, 101963. [\[CrossRef\]](#)
19. Zhu, L.; Spachos, P.; Pensini, E.; Plataniotis, K.N. Deep learning and machine vision for food processing: A survey. *Curr. Res. Food Sci.* **2021**, *4*, 233–249. [\[CrossRef\]](#)
20. Wu, H.; Du, C.; Ji, Z.; Gao, M.; He, Z. SORT-YM: An Algorithm of Multi-Object Tracking with YOLOv4-Tiny and Motion Prediction. *Electronics* **2021**, *10*, 2319. [\[CrossRef\]](#)
21. Nguyen, T.; Nguyen, H.; Bui, N.; Bui, T.; Vu, V.; Duong, H.; Hoang, H. Vision-Based System for Black Rubber Roller Surface Inspection. *Appl. Sci.* **2023**, *13*, 8999. [\[CrossRef\]](#)
22. Petchrompo, S.; Coit, D.W.; Brintrup, A.; Wannakrairot, A.; Parlikad, A.K. A review of Pareto pruning methods for multi-objective optimization. *Comput. Ind. Eng.* **2022**, *167*, 108022. [\[CrossRef\]](#)
23. Zou, X. A Review of Object Detection Techniques. In Proceedings of the 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), Xiangtan, China, 10–11 August 2019.
24. Joshi, S.A.; Bongale, A.M.; Olsson, P.O.; Urolagin, S.; Dharrao, D.; Bongale, A. Enhanced Pre-Trained Xception Model Transfer Learned for Breast Cancer Detection. *Computation* **2023**, *11*, 59. [\[CrossRef\]](#)
25. Yao, J.; Qi, J.; Zhang, J.; Shao, H.; Yang, J.; Li, X. A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5. *Electronics* **2021**, *10*, 1711. [\[CrossRef\]](#)

26. Kim, J.; Kim, N.; Park, Y.W.; Won, C.S. Object Detection and Classification Based on YOLO-V5 with Improved Maritime Dataset. *J. Mar. Sci. Eng.* **2022**, *10*, 377. [[CrossRef](#)]
27. Liang, H.; Tong, Y.; Zhang, Q. Spatial Alignment for Unsupervised Domain Adaptive Single-Stage Object Detection. *Sensors* **2022**, *22*, 3253. [[CrossRef](#)] [[PubMed](#)]
28. Shen, L.; Su, J.; He, R.; Song, L.; Huang, R.; Fang, Y.; Song, Y.; Su, B. Real-time tracking and counting of grape clusters in the field based on channel pruning with YOLOv5s. *Comput. Electron. Agric.* **2023**, *206*, 107662. [[CrossRef](#)]
29. Chen, Y.; Wu, B.; Luo, G.; Chen, X.; Liu, J. Multi-target tracking algorithm based on YOLO+DeepSORT. *J. Phys. Conf. Ser.* **2022**, *2414*, 012018. [[CrossRef](#)]
30. Hamuda, E.; Mc Ginley, B.; Glavin, M.; Jones, E. Improved image processing-based crop detection using Kalman filtering and the Hungarian algorithm. *Comput. Electron. Agric.* **2018**, *148*, 37–44. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.