Article

Lightweight Tennis Ball Detection Algorithm Based on Robomaster EP

Yuan Zhao [1], Ling Lu [1], Wu Yang [1,*], Qizheng Li [1], Xiujie Zhang [1]

[1] School of Computer Science and Engineering, Chongqing University of Technology, ChongQing 400054, China
* Correspondence:yw@cqut.edu.cn;zy1181759905@163.com

Abstract: To address the problems of poor recognition, low detection accuracy, a large number of model parameters and computation, complex network structure, and unfavorable portability to embedded devices in traditional tennis ball detection algorithms, this study proposes a lightweight tennis ball detection algorithm YOLOv5s-Z based on Robomater EP. The main work is as follows: Firstly, constructing lightweight G-Backbone and G-Neck network layers to reduce the number of parameters and computation of the network structure. Secondly, the convolutional coordinate attention is incorporated in G-Backbnone to embed the location information into the channel attention, which makes the network obtain the location information in a larger area through multiple convolutions and enhances the expression ability of the mobile network learning features. In addition, the Concat module in the original feature fusion is modified into a weighted bi-directional feature pyramid W-BiFPN with settable learning weights to improve the feature fusion capability and achieve efficient weighted feature fusion and bi-directional cross-scale connectivity. The EIOU loss is introduced to split the influence factor of aspect ratio and calculate the length and width of the target frame and anchor frame respectively, combined with Focal-EIOU Loss to solve the problem of imbalance between difficult and easy samples. The activation function Meta-ACON is introduced to achieve an adaptive selection of whether to activate the neurons and improve the detection accuracy. Finally, the experimental results show that compared with the original algorithm, the YOLOv5s-Z algorithm reduces the number of parameters and computation by 42% and 44%, the model size by 39%, and 2% improvement in average accuracy mean value, which verifies the effectiveness of the improved algorithm and the light weight of the model to meet the deployment requirements of embedded devices, and adapts Robomaster EP for accurate detection and real-time recognition of tennis balls.

Keywords: Tennis ball detection algorithm; Lightweight; Convolutional coordinate attention; Feature fusion; Loss function; Activation function

1. Introduction

Despite the popularity of tennis, the experience of the sport suffers from picking up the large number of tennis balls on the court. To solve the problem of low efficiency of manual ball pickup, many intelligent tennis ball pickup devices have emerged on the market, but they are mainly human-driven and mainly rely on the downward pressure action of the arm to pick up the tennis ball, which needs to repeat the arm lifting action several times during the pickup process.

With the rapid development of the robotics industry, service robots are becoming increasingly popular. The International Federation of Robotics (IFR) defines service robot [1] as a robot that works semi-autonomously or fully autonomously, and service robots are divided into two categories according to their uses: home service robots and professional service robots. Service robots for the tennis field are very rare in China, and most of them do not have models based on real environments. However, the world's first tennis AI ball-picking robot, Tennibot [2], has emerged abroad, using computer vision and artificial intelligence technology to automatically locate, detect and catch tennis balls, but the detection and recognition efficiency is not high. Therefore, service robots for the tennis field are in greater demand and have a good market value, broad market prospect, and certain practical significance.

Most of the traditional tennis ball detection algorithms are based on image processing methods [3], which mainly use computer vision or image sensors to process the image, range through digital image processing techniques [4], and apply target-ranging correlation algorithms to identify tennis balls. OpenCV-based image recognition firstly preprocesses the image, extracts features using correlation algorithms, sets different radius thresholds to screen the target, and finally determines whether it is a tennis ball based on the contour features, the commonly used method is based on the Hoff circle transform to detect tennis balls. Compared with manual ball pickup, the detection efficiency of the image-processing-based method is improved, but it is affected by different detection scenes, different time periods, and tennis ball occlusion. For example, different detection scenes and different time periods produce a certain degree of interference in identifying tennis balls, changes in the light lead to difficulty in extracting some target features, and the phenomenon of tennis ball occlusion leads to missed and false detection. In addition, the traditional tennis ball detection algorithm has a large number of model parameters, a large computational volume, Low detection accuracy, and a complex model structure, which is not conducive to porting to embedded devices.

For the above problems, we propose a lightweight tennis ball detection algorithm YOLOv5s-Z based on Robomaster EP [5] to achieve accurate detection and recognition of tennis balls. Robomaster EP is a programming-oriented educational robot from DJI with powerful scalability and programmability to perform a variety of complex tasks. Robomaster EP uses a High-performance servo as the main drive component, with a small side gap, high torque, and high repeat positioning accuracy. A parallel robotic arm is assembled on the top of the servo, a monocular camera is assembled on the top of the arm for real-time display and image transmission, and a mechanical claw is assembled at the end of the arm, which is used in conjunction with the mechanical claw. The infrared depth sensor is assembled on the top, based on TOF [6] (Time of Flight) principle, the sensor emits modulated near-infrared light, and after encountering the reflection of the object, the sensor calculates the distance to the object by calculating the time difference or phase difference between the light emission and reflection to achieve intelligent obstacle avoidance [7] and environment perception [8].Robomaster EP adapts to lightweight models, facilitating the porting and implementation of algorithms, calling algorithms through interfaces to achieve real-time detection and recognition, with high computational performance.

The main work and contributions of this study are as follows: Firstly, the lightweight backbone network G-Backbone and G-Neck network layers are proposed to reduce the number of parameters and computation of the model and build a lightweight model. Secondly, the convolutional coordinate attention mechanism is proposed to embed the location information into the channel attention, so that the network can obtain information of a larger area and enhance the expression ability of mobile network learning features. In addition, the Concat [9] module in the original feature fusion is modified into a weighted bidirectional feature pyramid W-BiFPN with settable learning weights to enhance the feature fusion capability and achieve efficient weighted feature fusion and bidirectional cross-scale connectivity. The EIOU Loss [10] is introduced to split the influence factors of aspect ratio to calculate the length and width the of target and anchor frames separately to solve the problem of imbalance between difficult and easy samples, the activation function Meta-ACON [11] is introduced to achieve an adaptive selection of whether to activate neurons and improve the detection accuracy. Finally, The experimental results show that compared with the YOLOv5s algorithm, the YOLOv5s-Z algorithm reduces the number of parameters and computation by 42% and 44%, the model size by 39%, and 2% improvement in average accuracy mean value, which verifies the effectiveness of the improved algorithm and the lightweight of the model to meet the deployment requirements of embedded devices and can be used for Robomaster EP accurate detection and real-time recognition.

Meaning that the proposed tennis ball detection algorithm has practical significance and future prospects.

2. Related Work and Motivation

2.1. Tennis ball detection algorithm

Tennis ball detection algorithms are mainly divided into methods of machine learning, image processing and deep learning. Ball objects generally have only two features, color and shape, and machine learning methods are relatively complex and not applicable to tennis ball recognition. Image processing mainly uses computer vision or image sensor methods, mostly based on OpenCV image recognition [12], which recognizes tennis balls by their color and shape features, firstly preprocessing the image, then extracting image features, and finally applying OpenCV-related algorithms for recognition. Image processing-based methods are susceptible to environmental and equipment influences, resulting in difficult extraction of image features and low detection accuracy and recognition efficiency. Compared with machine learning and image processing methods, deep learning-based tennis ball detection algorithms can achieve accurate detection and recognition of tennis balls. Gu et al. [13] proposed a model based on AlexNet [14] and SSD [15] for tennis ball recognition, using deep learning to divide the image recognition into two steps, using the AlexNet network model to test whether there is a tennis ball in the image, if there is a tennis ball, using SSD network model to locate the tennis ball, otherwise using AlexNet to continue checking the next image. Gu et al. [16] proposed a deep learning-based tennis ball collection robot using YOLO [17] model for tennis ball recognition and implemented it on an Nvidia Gitzo TX1 board.

2.2. Target detection algorithms

Traditional target detection algorithm, the main steps are: first select the candidate region on the image, then perform feature extraction, and finally use the classifier for classification. There are disadvantages such as low detection accuracy, high computational cost, and poor robustness.Deep learning-based target detection algorithms rely on a large amount of data to obtain feature information through autonomous learning and model training of convolutional neural networks [18] to achieve high-precision and high-efficiency detection and recognition. There are two main categories of deep learning-based target detection algorithms. The first category is the two-stage target detection algorithm based on region recommendation, the main steps: first generate candidate regions, and then classify the candidate regions. The representative algorithms are R-CNN [19], Fast R-CNN [20], Faster R-CNN [21], Mask R-CNN [22], etc. The main idea is to extract the image features by a convolutional neural network, use the region extraction network to give the candidate frame of the image target to be detected, and use the detection head with convolution to classify the target in the candidate frame to complete the detection. The accuracy is high, but the speed is slow. Another category is the one-stage method target detection algorithm based on the regression idea, that is, end-to-end, single-stage detection of objects, for a picture using only one convolutional neural network prediction to get the category probability and position coordinates of different targets, which greatly improves the detection speed and the operation speed of the algorithm to meet the demand of real-time detection, the representative algorithms are RetinaNet [23] YOLO, SSD, etc. The main idea of the algorithm is to extract features directly in the network, transform the target localization into a border regression problem, and complete the localization and classification tasks at once, which is faster but less accurate.

2.3. YOLOv5 Algorithm

The network structure of the YOLOv5s algorithm is shown in Figure 1. YOLOv5 is a single-stage target detection algorithm with good performance at present, based on the idea of regression, which reduces the target detection problem to a regression problem. The network model consists of the Input, Backbone, Neck network layer, and Output. The main role of the Input is to pre-process the input image, which mainly include mosaic data enhancement, adaptive anchor frame calculation, adaptive image scaling. The main role of Backbone is to extract the image features, which mainly include Focus, Conv, C3, SPPF Modules. The main role of the Neck network layer is to fuse the information of different network layers in the Backbone to further improve the detection of the network. The main role of the Neck network layer is to fuse the information of different network layers in the Backbone to further improve the detection capability of the network. It mainly adopts the structure of FPN+PAN to realize the transfer of target feature information of different sizes, strengthen the network feature fusion capability, and solve the multi-scale problem. The main role of the Output is to predict targets of different sizes on different feature maps, generate bounding boxes and predict categories, which mainly includes the calculation of loss functions and non-maximum suppression operations.

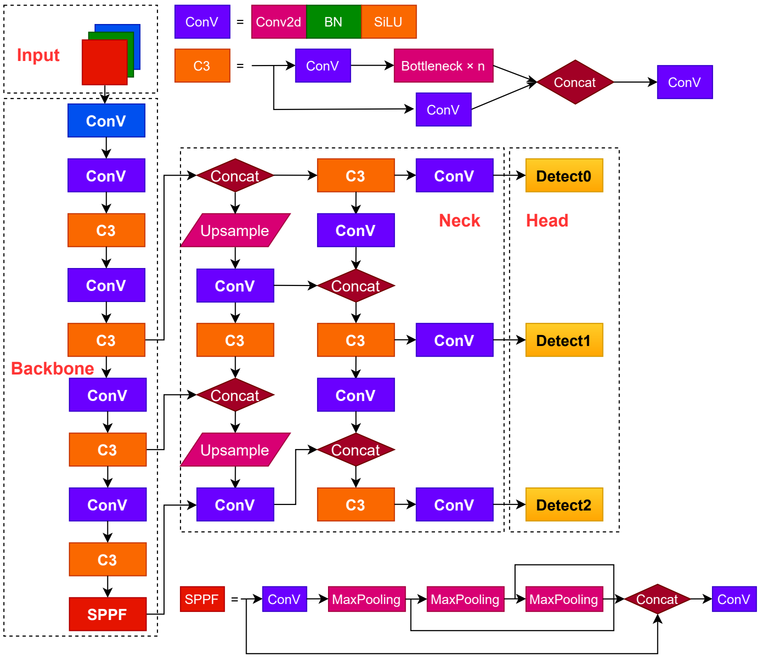

Figure 1. YOLOv5s network structure.

2.4. Lightweight neural networks

Since the introduction of AlexNet, neural networks have been widely used in image classification, image segmentation, target detection, and other fields. Due to the limitation of storage space and power consumption, it is difficult to store and compute neural networks on embedded devices. With the iterative update of embedded devices and the diversified development of model application scenarios, traditional neural network models are gradually replaced due to a large number of parameters and computation and complex network structure, and lightweight neural network structures emerge. In recent years, many excellent lightweight neural network structures have emerged. For example, EfficientNet [24] uses a model composite scaling method and AutoML [25] technique to scale up the convolutional neural network in a more structured way using a simple and efficient composite coefficient. SqueezeNet [26] uses a different convolutional approach from the traditional one by proposing a Fire module. MobileNet [27] builds a network based on depth-separable convolution, splitting the standard convolution into a depth convolution and a point convolution, and controlling the width and resolution size of the model by two hyperparameters. ShuffleNet [28] uses grouped convolution to reduce the number of parameters and uses channel shuffling to achieve information exchange between different groups. GhostNet [29] proposes a new basic unit of the neural network, the Ghost module, that uses inexpensive operations to generate as many feature maps as possible at a small cost, without changing the size of the output map or the channel size, reducing the number of parameters and computation of the model.

2.5. Motivation

To propose a lightweight tennis ball detection algorithm based on Robomaster EP for tennis ball detection and recognition, we have two main considerations. Firstly, Robomaster EP is a programming-oriented educational engineering robot from DJI with powerful scalability and programmability, as shown in Figure 2. A parallel robotic arm is used instead of the gimbal structure installed in the middle of the chassis, and the graphical transmission system is retained. A monocular camera is assembled on the top of the robotic arm for real-time display and image transmission, and a mechanical jaw is assembled at the end of the robotic arm At the end of the robot arm is equipped with a mechanical jaw, the robot arm, and the mechanical jaw cooperates to perform more complex tasks. The Robomaster EP is different from other tennis ball picking devices, with more comprehensive performance and the following advantages: (1) small size, a total weight of 3.3kg, and easy to carry. (2) Large storage space, with a microSD card, supporting up to 64GB. (3) Large range of movement of the robot arm and the opening and closing distance of the robot claw, the robot arm, and the robot claw work together to move more flexibly and more efficiently. (4) broader detection range, the top assembled infrared depth sensor detection range of 0.1-10 meters. Mechanical arm top assembly monocular camera, real-time display images. (5) Strong scalability and programmability, better compatibility, easy to call and deploy algorithms. (6) Large battery capacity, long standby time, and endurance can reach one hour. Secondly, the detection performance is maintained stable under real detection scenarios. The real scenario of tennis ball detection differs from other detection scenarios in that the number of tennis balls on the tennis court is often high and there is partial tennis ball occlusion, and it is also easily affected by weather and light. The image processing method using OpenCV leads to difficult feature extraction, low detection accuracy, and poor recognition effect. The deep learning-based tennis ball detection algorithm can effectively avoid the above phenomena, and according to the subsequent experimental results, it shows better detection performance and more robustness in different detection scenes or different periods of the same scene.

According to existing research, inspired by certain, this paper substantially reduces the number of parameters and computation by constructing a lightweight network structure, incorporates convolutional coordinate attention in Backbone to enhance the ability of the backbone network to feel the field and capture the location information, incorporates W-BiFPN in Neck to enhance the feature fusion ability and improve the detection speed, and introduces EIOU Loss and Meta- ACON to enhance the detection accuracy. As a result, the proposed lightweight tennis ball detection algorithm based on Robomaster EP is adapted to Robomaster EP to meet the deployment requirements of embedded devices for accurate detection and real-time identification and to promote the development of deep learning-based tennis ball detection algorithms, which has certain market prospects and practical significance.

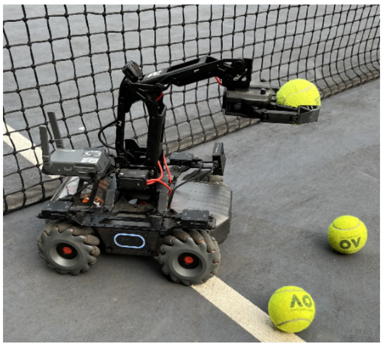

Figure 2. Robomaster EP.

3. YOLOv5s-Z algorithm

This study proposes a lightweight tennis ball detection algorithm YOLOv5s-Z based on Robomaster EP based on the YOLOv5s algorithm v6.1 version for improvement and optimization, the network structure is shown in Figure 3, which mainly consists of the input, lightweight G-Backbone and G-Neck network layers and the output.

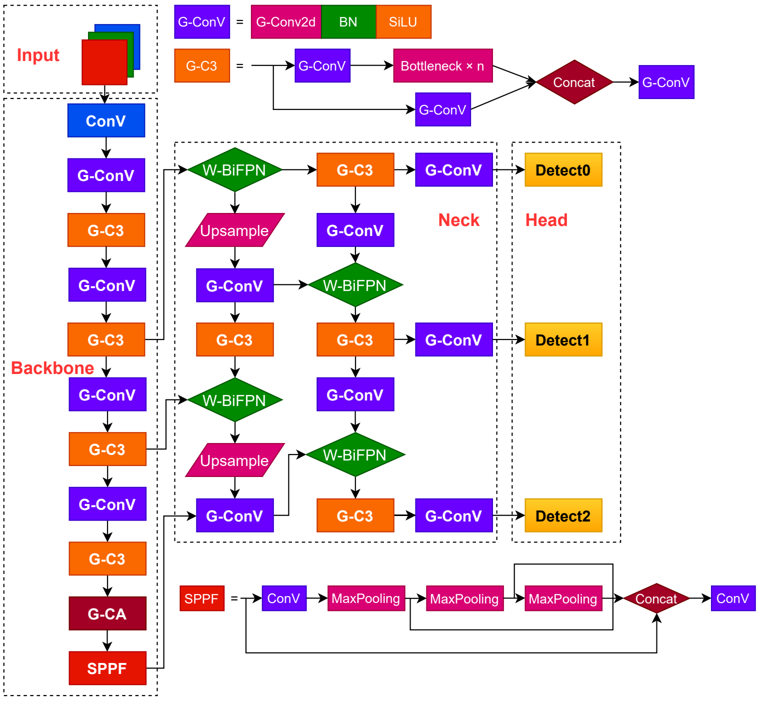

Figure 3. YOLOv5s-Z network structure.

3.1. G-Backbone

Based on the lightweight neural network GhostNet, the Conv and C3 modules are lightly processed to form the new lightweight G-Conv and G-C3 modules, and the model is pruned to construct the lightweight G-Backbone, and the convolutional coordinate attention mechanism G-CA is incorporated into the G-Backbone to enhance the perceptual field and the ability to capture location information of the backbone network to better extract the features of the input image. GhostNet is a new end-side neural network architecture proposed by Huawei Noah's Ark Lab, which builds a lightweight neural network GhostNet by stacking Ghost modules to get Ghost BottleNeck. The structure of the G-Backbone network is shown in Table 1.

Table 1. G-Backbone network structure.

layer	from	params	module	arguments
0	-1	4656	Conv	[3,32,6,2,2]
1	-1	12416	G-Conv	[32,64,3,2]
2	-1	14776	G-C3	[64,64,1]
3	-1	43232	G-Conv	[64,128,3,2]
4	-1	51472	G-C3	[128,128,2]
5	-1	160160	G-Conv	[128,256,3,2]
6	-1	189808	G-C3	[256,256,3]
7	-1	615200	G-Conv	[256,512,3,2]
8	-1	621472	G3	[512,512,1]
9	-1	24608	G-CA	[512,512,32]
10	-1	700208	SPPF	[512,512,5]

In above table, **layer** denotes the number of layers, **from** denotes the layer from which the module comes, where -1 denotes the previous layer, **params** denotes the number of parameters, **module** denotes the name of the module, and **arguments** denote the relevant information of the module, mainly including the number of input channels, the number of output channels, the size of the convolution kernel, step information, etc.

For a conventional convolutional neural network, the dimension of the input feature map is c×h×w, where c represents the number of channels, h represents the height of the feature map, and w represents the width of the feature map. The size of the convolution kernel is c×k^2×n, where k represents the size of the convolution kernel and n represents the number of channels of the output feature map. Let the size of the output feature map be h′×w′×n, the total computation is h′×w′×n×c×k^2, and the number of parameters is c×k^2×n. The output data of the ordinary convolution Y=X∗f+b, where X represents the input data, f represents the c×n convolutional operations with the size of the k^2 convolution kernel, b represents the bias term, and the ordinary convolution is shown in Figure 4.

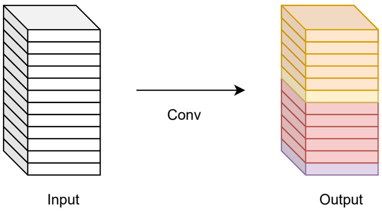

Figure 4. The Normal Convolution.

Compared with the ordinary convolution, Ghost convolution is shown in Figure 5. The main steps: firstly, a small number of feature maps are generated by ordinary convolution, and then a linear operation is performed on the feature maps to generate Ghost feature maps, and the two different groups of feature maps are stitched together to finally obtain the same output results as ordinary convolution.

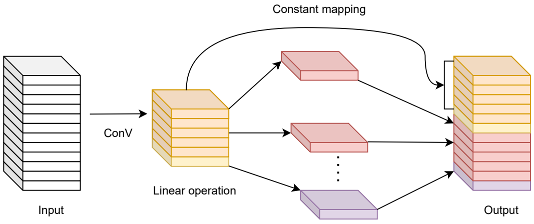

Figure 5. The Ghost Convolution.

The output data Y′=X∗f′+b of Ghost convolution is calculated as follows: firstly, the feature map Y′ is obtained by ordinary convolution, the feature map of each channel in Y′ is generated by a linear operation to generate Ghost feature map Y_{ij}, and then the two groups of feature maps are stitched according to the channels, and the output result is finally obtained. The feature map ϕ_{ij} is calculated as shown in Equation (1), where ϕ_{ij} denotes i feature map $Y_i{}'$ generated in the first step of convolution for the j linear operation, and $Y_i{}'$ denotes i feature map in Y′.

$$Y_{ij} = \phi_{ij} * Y_i', i \in [1, m], j \in [1, s] \tag{1}$$

The Ghost module mainly contains a small number of convolutions, a constant mapping, and linear operations m×(s-1), each with an average kernel size of d×d. The theoretical speedup ratio of the Ghost module to upgrade the ordinary convolution is calculated as shown in Equation (2):

$$r_s = \frac{n \cdot h' \cdot w' \cdot c \cdot k^2}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k^2 + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d^2} = \frac{c \cdot k^2}{\frac{1}{s} \cdot c \cdot k^2 + \frac{s-1}{s} \cdot d^2} \approx \frac{s \cdot c}{s + c - 1} \approx s \tag{2}$$

The compression ratio of the number of parameters between the normal convolution and Ghost convolution is calculated as shown in Equation(3):

$$r_c = \frac{n \cdot c \cdot k^2}{\frac{n}{s} \cdot c \cdot k^2 + \frac{s-1}{s} \cdot d^2} \approx \frac{s \cdot c}{s + c - 1} \approx s \tag{3}$$

where d represents the size of the convolution kernel when linear mapping is performed for each of the m channels, from the calculation results, we know that the theoretical acceleration ratio and compression ratio are equal, then it can be inferred that the computational and parametric quantities of ordinary convolution are s times larger than those of Ghost convolution. Ghost convolution is a faster and lighter module, and this study builds a lightweight backbone network based on GhostNet, G-Backbone for better feature extraction of the input image.

3.2. Convolutional coordinate attention

In this study, we propose a convolutional coordinate attention G-CA, as shown in Figure 6. By incorporating the convolutional coordinate attention mechanism in the G-Backbone backbone network, multiple convolutions enable the network to obtain location information in a larger area, further enhancing the ability of the backbone network to sense the field and capture location information, and enhancing the expression of the learning features of the mobile network.

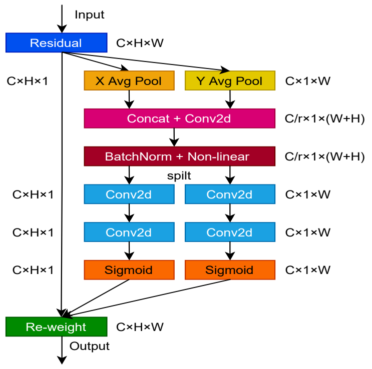

Figure 6. Convolutional coordinate attention.

Studies of neural networks have shown that channel attention significantly improves the performance of the model, but ignores some important location information that facilitates the generation of spatially selective attention maps. Therefore, to alleviate the loss of location information caused by two-dimensional global pooling proposed by attention mechanisms such as SENet [30] and CBAM [31], a novel attention mechanism designed for lightweight networks, called coordinate attention CA [32], was proposed by the National University of Singapore. Compared to channel attention, coordinate attention transforms the feature tensor into individual feature vectors using two-dimensional global pooling and decomposes channel attention into two one-dimensional feature encoding processes that aggregate features along two directions, one of which captures remote dependencies along the spatial direction and the other retains precise location information along the spatial direction. The resulting feature maps are finally encoded separately to generate a pair of direction-aware and position-sensitive feature maps, which are complementarily applied to the input feature maps and used to enhance the precise localization of targets.

Coordinate attention consists of two main steps: coordinate information embedding and coordinate attention generation. First, given an input feature map X with dimension c×h×w, and using two pooling kernels with spatial ranges (H,1) or (1,W) to encode each channel along the horizontal and vertical coordinates, respectively, the output of the c channel with height h and the c channel with width w is calculated as shown in Equations (4)-(5):

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} X_c(h, i) \tag{4}$$

$$Z_c^w(h) = \frac{1}{H} \sum_{0 \le j \le H} X_c(j, w) \tag{5}$$

The above two transformations perform feature aggregation along two directions, respectively, and cascade to generate two feature maps, which generate feature maps of spatial information in horizontal and vertical directions f by convolution operation, which is beneficial to the network for accurate target localization, and the calculation formula is shown in Equation (6):

$$f = \delta(F_1([Z^h, Z^w])) \tag{6}$$

After the coordinate information is embedded, the above changes are subjected to the cascade operation, which is a nonlinear activation function that is an intermediate feature map of spatial information encoded along the horizontal and vertical directions, which is decomposed into two tensor sums along the spatial dimension. The transformation operation is then performed using the convolutional transform function, which in turn yields the attention weights of the two spatial directions as g^h and g^w, respectively, calculated as shown in Equations (7)-(8):

$$g^h = \sigma((F_h(f^h))) \tag{7}$$

$$g^w = \sigma((F_w(f^w))) \tag{8}$$

The σ in the above equation is the Sigmoid [33] activation function, and to reduce the complexity and computational overhead of the model, the number of channels is usually reduced using a suitable scaling ratio, and the output and are expanded as attention weights, respectively. The final output of the coordinate attention mechanism is obtained, and the calculation formula is shown in Equation (9):

$$y_c(i, j) = X_c(i, j) \times g_c^h(i) \times g_c^w \tag{9}$$

3.3. G-Neck Network Layer

The lightweight G-Neck network layer is constructed, and the weighted bidirectional feature pyramid W-BiFPN with settable learning weights is proposed to be incorporated into the G-Neck network layer, and the settable learning weight coefficient W is set based on the weighted bidirectional feature pyramid BiFPN to further strengthen the feature fusion capability and improve the detection speed to achieve more efficient weighted bidirectional feature fusion, which is more conducive to network extraction features.

Figure 7 shows the development process of Neck networks in recent years, starting with the top-down unidirectional fusion FPN feature pyramid structure, which establishes a top-down pathway for feature fusion and uses feature maps for prediction, which can improve the accuracy to a certain extent but will be limited by the unidirectional information flow. the network structure of PANet[34] for bidirectional fusion, based on the FPN[35] adding a bottom-up path aggregation network. The main idea is that the feature map at the top layer has stronger semantic information, which is beneficial for object classification, and the feature map at the bottom layer has stronger location information, which is beneficial for object localization. the PANet network passes the location information from the bottom layer to the prediction feature layer, which makes the prediction feature layer have both higher semantic information and location information, which is more beneficial for target detection and thus improves the detection accuracy. the PANet network structure also Adaptive feature pooling and full-connected fusion are proposed in the PANet network structure, where adaptive feature pooling is used to aggregate features between different layers to ensure the integrity and diversity of features, and full-connected fusion is used to obtain accurate prediction layers. The main idea of weighted bidirectional feature pyramid network structure BiFPN[36] is effective bidirectional cross-connection and weighted feature fusion, top-down feature fusion followed by bottom-up feature fusion, and multi-scale feature fusion is an aggregation of features at different resolutions.

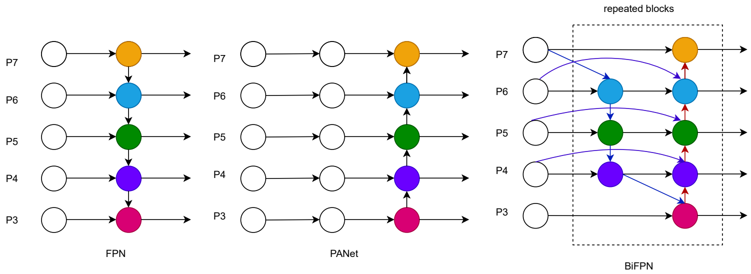

Figure 7. FPN PANet and BiFPN structure.

Since different input features have different resolutions, the contribution to the output features is uneven. To solve this problem, an additional weight is added to each input so that the network learns the importance of each input feature.BiFPN uses a fast normalized weighted fusion method, which is calculated as shown in Equation (10):

$$o = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i \tag{10}$$

where $w_i \geq 0$ is achieved by adding the ReLU activation function after each w_i, and the weights are divided by the weighted sum of all values to achieve the normalization operation, and the value of each normalization weight is between 0 and 1. BiFPN integrates bidirectional cross-scale connectivity and fast normalized fusion, and the computational equations of the fusion feature process are shown in Equations (11)-(12) for BiFPN at level 6 nodes.

$$p_6^{td} = Conv(\frac{w_1 \cdot p_6^{in} + w_2 \cdot Resize(p_7^{in})}{w_1 + w_2 + \epsilon}) \tag{11}$$

$$p_6^{out} = Conv(\frac{w_1' \cdot p_6^{in} + w_2' \cdot p_6^{td} + w_3' \cdot Resize(p_5^{out})}{w_1' + w_2' + w_3' + \epsilon}) \tag{12}$$

In the above equation, p_6^{td} denotes the intermediate features of the sixth layer in the top-down path, and p_6^{out} denotes the output features of the sixth layer in the bottom-up path. To improve efficiency, feature fusion is performed using depth-separable convolution, and batch normalization and activation functions are added after each convolution, where Resize is the upsampling or downsampling operation, and w is the learned parameter to distinguish the importance of different features in the feature fusion process.

3.4. EIOU Loss

The default loss function in YOLOv5s is CIOU Loss[37]. CIOU takes the aspect ratio of the regression frame into account in the loss function based on DIOU[38], and increases the Loss of the detection frame scale as well as the Loss of the length and width, which makes the prediction frame more realistic and further improves the regression accuracy. The disadvantage is that there is a vague definition of aspect ratio and the balance problem of difficult and easy samples is not considered. In this study, EIOU Loss is introduced, and the longitudinal influence factors of the prediction frame and the rear frame are split based on the penalty term of CIOU Loss, and the length and width of the prediction frame and the rear frame are calculated separately to solve the problems existing in CIOU Loss.

The formula for calculating EIOU Loss is shown in Equation (13):

$$L_{EIOU} = L_{IoU} + L_{dis} + L_{asp} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{13}$$

From the above equation, we can see that the EIOU Loss consists of three main components: the overlap loss between the predicted frame and the real frame L_{IoU}, the center distance loss between the predicted frame and the real frame L_{dis}, and the width and height loss between the predicted frame and the real frame L_{asp}. where L_{IoU} and L_{dis} continue the method in CIOU, and the width and height loss L_{asp} directly makes the difference between the width and height of the predicted frame and the real frame minimizes the difference between the width and height of the predicted frame and the rear frame, which makes the convergence faster.

The comparison diagram of the iterative process of CIOU and EIOU loss prediction frames is shown in Figure 8, where the red and green boxes represent the regression process of CIOU and DIOU prediction frames respectively, the blue box is the real box, and the yellow box is the pre-defined anchor box.From the comparison graph, it can be seen that the width and height of EIOU can be increased or decreased at the same time, but CIOU cannot. In general, EIOU outperforms CIOU, so EIOU Loss is introduced as the loss function in this study.

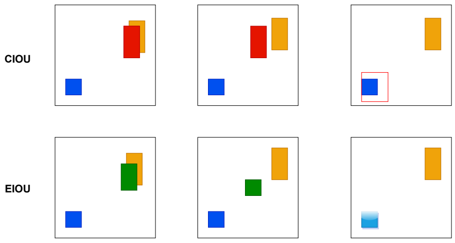

Figure 8. Comparison of CIOU and EIOU iterative process.

3.5. Meta-ACON activation function

The default activation function in YOLOv5s is ReLU, which is the most common activation function, mainly because of its non-saturation, sparsity, and other properties, with the drawback that it can have the serious consequence of neuronal necrosis. ReLU is essentially a MAX function, and the formula is shown in Equation (14):

$$ReLU(x) = MAX(0, x) \tag{14}$$

Consider the n values of the standard maximum function MAX, whose smoothness and differentiability are approximated by S_β, calculated as shown in Equation (15):

$$S_\beta(x_1, x_1, x_2, ..., x_n) = \frac{\sum_{i=1}^{n} X_i \cdot e^{\beta x_i}}{\sum_{i=1}^{n} e^{\beta x_i}} \tag{15}$$

where β is a connection coefficient, and S_β tends to the maximum when β tends to infinity, and S_β tends to the arithmetic mean when β tends to zero. In neural networks, the common activation function is expressed in the form of $max(\eta_a(x), \eta_b(x))$, where $\eta_a(x)$ and $\eta_b(x)$ are linear functions.

In recent years, the Swish activation function obtained by the NAS search technique is an approximate smoothing of the ReLU activation function, and the general form of the Maxout series activation function of ReLU is analyzed to obtain the general form of the ACON activation function of Swish. ACON is generalized to obtain variants of ACON-A, ACON-B, ACON-C, Meta-ACON, etc. In this study, Meta-ACON is introduced to adaptively select whether to activate neurons or not, and a switching factor is introduced to learn the parameter switching between nonlinear activation and linear inactivation to improve the detection accuracy of the algorithm.

doi:10.20944/preprints202211.0561.v1

4. Experimental and results

4.1. Experimental environment

This experiment is based on Pytorch 1.11.0 framework, CUDA version 11.5, and conducted on the PyCharm platform, and the model training is accelerated by GPU. The specific experimental environment parameters are configured as shown in Table 2.

Table 2. Experimental environment.

Name	Configuration parameters
Operating System	Windows 11 64-bit
CPU	Intel Core i5-12400F
GPU	NVIDIA GeForce RTX3060 12G
Memory	16GB
Python Version	3.8
Deep Learning Framework	Pytorch 1.11.0 CUDA 11.5
Experimental Platform	PyCharm Community Edition 2022.2.3

4.2. Datasets

In this study, 1180 homemade tennis ball datasets are used, and the sources of the datasets include tennis ball pictures taken by the monocular camera assembled with Robomaster EP, tennis ball pictures taken by cell phones, and tennis ball pictures obtained by crawlers, containing different colors, different scenes, and different time tennis ball pictures to ensure the diversity of the datasets, and the specific information is shown in Table 3.

Table 3. Datasets.

Category	Parameters
Color	green, blue, orange, purple, pink, black
Scene	tennis court, laboratory, open space
Period	morning, noon, evening

The datasets are normatively labeled using Make Sense, with clear annotation. At the same time, the training set, validation set, and test set are divided according to 8:1:1. Before training the model, some of the datasets are pre-processed, including randomly increasing or decreasing the brightness and contrast of images. The datasets are enriched with the Mosaic data enhancement method that comes with YOLOv5 to enhance the generalization ability of the model and the robustness of the validation model. Figure 9 shows an example figure representing the datasets.

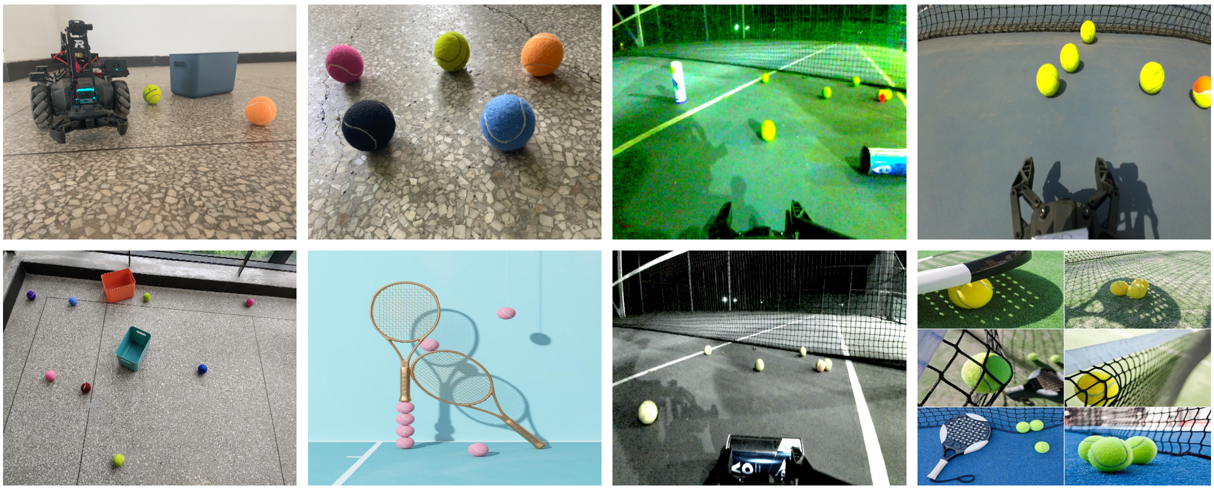

Figure 9. Representative datasets.

4.3. Training strategy and evaluation index

All models are trained from scratch using the same training strategy and parameters, hyperparameter profiles, and preheat training parameters, all without pre-training weights. The parameters were updated iteratively using an SGD optimizer with an initial learning rate of 0.01, a momentum parameter of 0.937, and a batch size of 16. The warm-up method with the epoch of 3 and momentum parameter of 0.8 was used to warm up the learning rate, and all models were trained for 300 rounds.

In this study, the model is evaluated using evaluation metrics including mAP@0.5, Recall, Parameters, GFLOPs, and Weight. mAP@0.5 represents the average AP at the IOU threshold of 0.5, which is used to reflect the recognition ability of the model. Recall represents the ratio of correctly detected positive samples to all positive samples, Parameters represents the number of parameters of the model, and GFLOPs represent the number of floating point operations performed by the model. Parameters and GFLOPs are important indicators of the model algorithm, which measure the complexity of the model in the dimensions of time and space, respectively. The calculation equations are shown in Equations (16)-(19).

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{16}$$

$$AP = \int_0^1 P(R)dR \tag{17}$$

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

Where n represents the number of categories, p represents precision, R represents recall, P(R) represents the precision and recall curves, TP represents the number of detection frames with IOU set threshold, FP represents the number of detection frames with IOU set threshold, and FN represents the number of missed targets.

4.4. Comparative experimental results and analysis

To verify the effectiveness of the improved algorithm, commonly used target detection algorithms were selected for comparative analysis, and the same training strategy and parameters were used for each group of experiments, and the experimental results are shown in Table 4.

Table 4. Contrast experiment.

Model	mAP@0.5	Recall	Parameters	GFLOPs	Weight/MB
YOLOv5s	0.957	0.911	7022326	15.8	14.5
SSD	0.839	0.443	26285486	63	91
Faster R-CNN	0.93	0.935	28536850	181	109
YOLOv5s+Shufflenetv2	0.939	0.915	3792950	7.9	8.1
YOLOv5s+Mobilenetv3	0.947	0.895	**3542756**	**6.3**	**7.5**
YOLOv3	0.964	0.944	61523734	154.9	123.6
YOLOv3-tiny	0.945	0.884	8669876	12.9	17.4
YOLOv4	0.909	0.885	64363101	60.5	244
YOLOv4-tiny	0.868	0.854	6056606	7.0	22.4
YOLOX-s	0.966	0.966	8968255	26.9	34.3
YOLOX-tiny	0.971	0.969	5055855	15.4	19.4
YOLOv7	0.962	0.97	37194710	104.9	71.3
YOLOv7-tiny	0.968	**0.98**	6014038	13.1	11.7
Ours	**0.978**	**0.98**	4100759	8.8	8.8

From the comparative experimental results, it is clear that the algorithm proposed in this study has the most comprehensive performance, which takes into account the needs of lightweight models and detection accuracy, has strong generalization ability, has the highest average precision mean and recall, and has slightly more number of parameters, computation and model size than the lightweight neural networks Mobilenet and Shufflenet, but Mobilenet and Shufflenet have lower average precision mean and recall. Compared with the original YOLOv5s algorithm, the number of parameters and computation is reduced by 42% and 44%, respectively, the model size is reduced by 39%, and the average precision mean value is improved by 2%, which verifies the effectiveness of the improved algorithm. Compared with the classical target detection algorithms SSD and Faster R-CNN, the comprehensive performance is more excellent, and the average precision means and recall increase significantly while the number of parameters, computation, and computation is reduced significantly. Compared with the YOLO series of target detection algorithms YOLOv3, YOLOv4, YOLOX, and the lightweight model, the comprehensive performance is still more excellent. Even compared with the current best-performance YOLOv7 algorithm, the comprehensive performance of the proposed algorithm is better. Compared with YOLOv7-tiny, the number of parameters and computation is reduced by 32% and 33%, respectively, and the model size is reduced by 25%, which verifies the effectiveness of the improved algorithm and the lightweight of the model.

4.5. Results and analysis of ablation experiments

To verify the feasibility of the improvement module, six sets of ablation experiments were designed on the basis of YOLOv5s, and the same training strategy was used for each set of experiments, and the results of the ablation experiments are shown in Table 5.Where Improve1 indicates the introduction of lightweight G-Backbone, Improve2 indicates the addition of G-CA attention mechanism, Improve3 indicates the addition of W-BiFPN module, Improve4 indicates the introduction of EIOU Loss, and Improve5 indicates the introduction of Meta-ACON activation function.

Table 5. Ablation experiments.

Model	G-Backbone	G-CA	W-BiFPN	EIOU Loss	Meta-ACON	mAP@0.5	Recall	Parameters	GFLOPs	Weight/MB
YOLOv5s						0.957	0.911	7022326	15.8	14.5
Improve1	√					0.956	0.926	**3684542**	**8.1**	**7.9**
Improve2		√				0.961	0.919	7046934	15.9	14.5
Improve3			√			0.959	0.918	7170943	16.4	14.8
Improve4				√		0.963	0.97	7022326	15.8	14.5
Improve5					√	0.962	0.931	7421478	16.2	15.4
Ours	√	√	√	√	√	**0.978**	**0.98**	4100759	8.8	8.8

From the results of the ablation experiments, it can be seen that the introduction of G-Backbone significantly reduces the number of parameters, computation, and model size of the network structure, while the average precision means value remains stable, which verifies the effectiveness of the lightweight module. With the introduction of G-CA and W-BiFPN, although the number of parameters and computational volume increase slightly, the average precision means to value and recall rate are improved, which verifies the effectiveness of the improved module. With the introduction of EIOU Loss, the number of parameters, computation, and model size remains unchanged, but the average precision mean value is slightly increased and the recall rate is increased by nearly 7%, which verifies that the performance of EIOU Loss is better than CIOU Loss. . This study incorporates all the improved modules, the number of parameters, computation, and model size is reduced, and the average precision mean and recall are improved by 2% and 7% respectively, which takes into account the demand of lightweight model and detection accuracy, further verifies

doi:10.20944/preprints202211.0561.v1

the effectiveness of the improved algorithm, and adapts Robomaster EP to achieve accurate detection and real-time recognition of tennis balls.

4.6. Case study

We further empirically investigated the detection performance under different scenarios, and the detection results are shown in Figure 10, all based on real detection scenarios, where Figures (a)-(d) represent the detection results of YOLOv5s, and Figures (e)-(h) represents the detection results of YOLOv5s-Z. The detection scenes in Figures (a) and (e) are tennis rackets on an open field, Figures (b) and (f) are tennis courts in the morning, Figures (c) and (g) are tennis courts in the evening, Figures (b)-(g) are real-time detection scenes based on the Robomaster EP monocular camera and Figures (d) and (h) are laboratory detection scenes. The detection results show that the detection effect of YOLOv5s-Z is better than the YOLOv5s algorithm in different scenes or different time periods, and the YOLOv5s-Z algorithm has higher detection accuracy and stability and better recognition, which further verifies the detection performance of the YOLOv5s-Z algorithm.

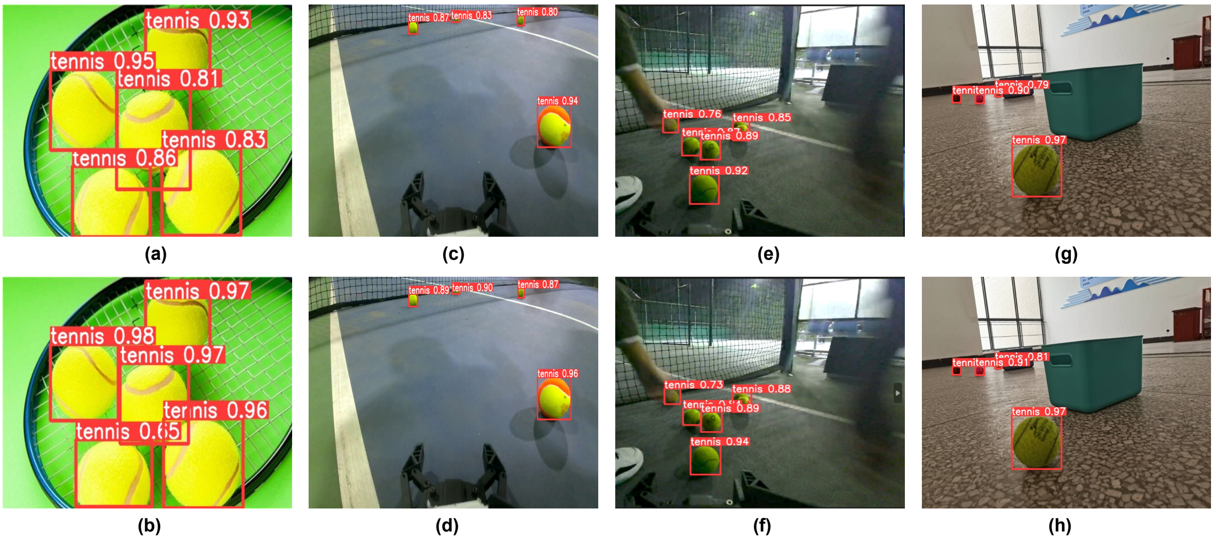

Figure 10. Comparison chart of test results.

5. Conclusion

To adapt Robomaster EP for accurate detection and real-time recognition of tennis balls, we propose YOLOv5s-Z algorithm, construct lightweight G-Backbone and G-Neck network layers, propose a convolutional coordinate attention mechanism and incorporate it into the backbone feature extraction network, makes the network obtain location information of a larger area through multiple convolutions, further enhances the feature extraction. The G-Neck network layer incorporates a weighted bi-directional feature pyramid W-BiFPN with settable learning weights to further enhance the feature fusion capability and achieve more efficient weighted feature fusion and bi-directional cross-scale connectivity. The loss function EIOU Loss is introduced to split the influence factor of aspect ratio based on the penalty term of CIOU Loss to calculate the length and width of target and anchor frames respectively, and the activation function Meta-ACON is introduced to adaptively select whether to activate neurons to improve the detection accuracy. Finally, the YOLOv5s-Z algorithm is deployed to Robomaster EP to achieve accurate detection and real-time recognition of tennis balls, which verifies the effectiveness of the YOLOv5s-Z algorithm and the lightweight of the model, and has some practical significance and future prospects in the field of tennis ball detection. In future work, we will further optimize the network model and optimize the network structure more comprehensively and deeply to achieve mobile target detection and improve detection efficiency and detection accuracy.

Author Contributions: Writing—original draft preparation,Y.Z.;writing—review and editing, L.L.; Validation W.Y. All authors have read and agreed to the published version of the manuscript.

Funding:

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

[1] Kerstin Severinson-Eklundh, Anders Green, and Helge Hüttenrauch. "Social and collaborative aspects of interaction with a service robot". In: *Robotics and Autonomous systems* 42.3-4 (2003), pp. 223–234.

[2] Dulanjana M Perera et al. "Development of a Vision Aided Automated Ball Retrieving Robot for Tennis Training Sessions". In: *2019 14th Conference on Industrial and Information Systems (ICIIS)*. IEEE. 2019, pp. 378–383.

[3] Athanasios Tsalatsanis, K Valavanis, and Ali Yalcin. "Vision based target tracking and collision avoidance for mobile robots". In: *Journal of Intelligent and Robotic Systems* 48.2 (2007), pp. 285–304.

[4] Ioannis Pitas. *Digital image processing algorithms and applications*. John Wiley & Sons, 2000.

[5] AI Hopkins. "DJI RoboMaster AI Challenge Technical Report". In: ().

[6] Sergi Foix, Guillem Alenya, and Carme Torras. "Lock-in time-of-flight (ToF) cameras: A survey". In: *IEEE Sensors Journal* 11.9 (2011), pp. 1917–1926.

[7] Limin Ren, Weidong Wang, and Zhijiang Du. "A new fuzzy intelligent obstacle avoidance control strategy for wheeled mobile robot". In: *2012 IEEE International Conference on Mechatronics and Automation*. IEEE. 2012, pp. 1732–1737.

[8] Marcel Schweiker et al. "Review of multi-domain approaches to indoor environmental perception and behaviour". In: *Building and Environment* 176 (2020), p. 106804.

[9] Qingguo Zeng, Xiangru Li, and Haitao Lin. "Concat Convolutional Neural Network for pulsar candidate selection". In: *Monthly Notices of the Royal Astronomical Society* 494.3 (2020), pp. 3110–3119.

[10] Yi-Fan Zhang et al. "Focal and efficient IOU loss for accurate bounding box regression". In: *Neurocomputing* 506 (2022), pp. 146–157.

[11] Ningning Ma et al. "Activate or not: Learning customized activation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8032–8042.

[12] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[13] Shenshen Gu et al. "A new deep learning method based on AlexNet model and SSD model for tennis ball recognition". In: *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*. IEEE. 2017, pp. 159–164.

[14] Forrest N Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[15] Wei Liu et al. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[16] Shenshen Gu et al. "A deep learning tennis ball collection robot and the implementation on nvidia jetson tx1 board". In: *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2018, pp. 170–175.

[17] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[19] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[20] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[21] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[22] Kaiming He et al. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[23] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[24] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[25] Xin He, Kaiyong Zhao, and Xiaowen Chu. "AutoML: A survey of the state-of-the-art". In: *Knowledge-Based Systems* 212 (2021), p. 106622.

[26] Forrest N Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[27] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[28] Xiangyu Zhang et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6848–6856.

[29] Kai Han et al. "Ghostnet: More features from cheap operations". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 1580–1589.

[30] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[31] Sanghyun Woo et al. "Cbam: Convolutional block attention module". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[32] Qibin Hou, Daquan Zhou, and Jiashi Feng. "Coordinate attention for efficient mobile network design". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 13713–13722.

[33] Jun Han and Claudio Moraga. "The influence of the sigmoid function parameters on the speed of backpropagation learning". In: *International workshop on artificial neural networks*. Springer. 1995, pp. 195–201.

[34] Shu Liu et al. "Path aggregation network for instance segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768.

[35] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[36] Mingxing Tan, Ruoming Pang, and Quoc V Le. "Efficientdet: Scalable and efficient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10781–10790.

doi:10.20944/preprints202211.0561.v1

[37] Zhaohui Zheng et al. “Enhancing geometric factors in model learning and inference for object detection and instance segmentation”. In: *IEEE Transactions on Cybernetics* (2021).

[38] Zhaohui Zheng et al. “Distance-IoU loss: Faster and better learning for bounding box regression”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 12993–13000.