Optimized YOLOv5 with Unity 3D for Efficient Gesture Recognition in Complex Machining Environments

Chen Jiang
Department of Urban Construction Engineering, Wenhua College, Wuhan, 430074, China
Email: ali_jojo@163.com

Keywords: kinect 2.0, YOLOv5, attention mechanism, unity 3D, complex processing equipment, gesture interaction

Received: August 21, 2024

To improve the efficiency of human-machine interaction in complex machining environments and optimize the accuracy of gesture recognition, a new gesture recognition system is developed by combining the improved You Only Look Once 5 and Unity 3D software. Firstly, an efficient channel attention mechanism is introduced to optimize the network structure of the fifth version of the algorithm to process higher dimensional gesture image data. Secondly, a twin model of complex processing equipment is constructed, and real-time visualization of gesture data and human-machine interaction are achieved using Unity 3D. The research results indicated that the designed static gesture recognition algorithm achieved image signal-to-noise ratio and image intersection to union ratio of 0.95 and 0.98 during the training process. In practical applications, the gesture interaction recognition model designed using this algorithm exhibited extremely low response time, with a minimum of 0.02s to complete the recognition task. At the same time, the recognition accuracy of this model reached up to 99.1%, which was much higher than the other three comparative models. In the practical performance tests, for the different four datasets, the recognition accuracy of YOLOv5-ECA model was 98.5%, 98.7%, 99.1% and 98.8%, with the recognition time as low as 0.07s, 0.02s, 0.11s and 0.08s, respectively. It can be seen that the gesture recognition system provides a new technical solution for human-machine interaction of complex processing equipment, which can further improve the operational efficiency and safety of human-machine interaction.

Povzetek: Razvit je optimiziran YOLOv5 z Unity 3D za izboljšano prepoznavo gest v kompleksnih strojnih okoljih. Rezultati potrjujejo visoko učinkovitost pri izboljšanju varnosti in operativne učinkovitosti človek-stroj interakcije, kar omogoča napredne rešitve v industrijski avtomatizaciji.

1 Introduction

With the development of industrial automation and intelligent manufacturing, complex processing equipment has become particularly important in modern manufacturing. This equipment has high precision, multi-functionality, and high automation, which can handle more complex process flows [1-2]. In recent years, the development of the Internet of Things and digital twin technology has provided new solutions for complex processing equipment [3-4]. Digital twin technology achieves real-time monitoring, simulation, and remote control of devices by creating virtual models. However, how to improve the real-time monitoring and control efficiency of complex processing equipment, especially the accuracy and efficiency of human-machine interaction, is still an important research topic. At present, gesture recognition technology based on deep learning has been widely applied in academia and industry, especially the You Only Look Once (YOLO) algorithm [5]. This series of algorithms has attracted widespread attention in the object detection and gesture recognition due to their high efficiency and real-time performance. Gestures, as a primitive and natural way of human-machine interaction, existed before the development of language and were mainly used for information transmission. Various gestures and commands can not only convey information concisely, but also perform complex operations. In human-machine interaction, gestures provide a highly flexible communication form, simplifying the interaction process by avoiding direct physical contact between mechanical devices and users. In addition, gesture interaction can provide more intuitive operating methods and a rich interaction experience, better meeting the needs and expectations of users for interaction methods. In previous studies, Zhang proposed three different gesture feature extraction methods to improve the recognition accuracy of human-machine interaction gestures, namely scale invariant feature transformation, local binary mode, and directional gradient histogram. Three feature extraction methods combined with backpropagation neural networks were used to complete gesture classification and recognition tasks. The research results indicated that the gesture feature map information extracted from the directional gradient histogram was closest to the original image. This method, combined with backpropagation neural networks, had a faster convergence speed, the smallest stable error, and the highest recognition accuracy [6]. Li et al. proposed a gesture recognition method based on surface electromyography signals for

human-machine interaction in rehabilitation equipment. In addition, a gesture classification model combining convolutional neural networks and long short-term memory networks was proposed to classify five dynamic gestures. Finally, tests were conducted on five different limb positions. It was found that the dynamic gesture recognition accuracy of this method reached 84.2% [7]. Chakravarthi et al. proposed a gesture recognition system based on extreme learning to address the gesture recognition in human-machine interaction. The system could quickly and accurately recognize gestures by displaying hand movements in front of the camera, which was helpful for people with different backgrounds to use. The research results indicated that the constructed gesture recognition system could quickly interpret different gestures and improve the accuracy of gesture interaction, which was particularly suitable for fields such as healthcare, financial transactions, and smart transportation [8]. The total summary table of related works is shown in Table 1.

Table 1: General summary of related works

Method	Accuracy	Response Time	Operational and Computational Efficiency	Limitations
The method proposed by Zhang	96.1%	-	Moderate efficiency	Limited applicability
The method proposed by Li et al.	84.2%	-	Moderate efficiency	Limited to rehabilitation equipment
The method proposed by Chakraverthi	87.5%	0.33s	Moderate efficiency	Limited applicability
SSD	84.5% - 88.2%	0.23s - 0.33s	General computational efficiency, longer response time	Basic model, lacks additional attention mechanism, lower accuracy in detailed feature recognition
YOLOv5	87.6% - 90.3%	0.15s - 0.26s	Higher computational efficiency and shorter response time than SSD	Basic model, lacks additional attention mechanism, lower accuracy in detailed feature recognition
EMAFF-Net	90.4% - 93.1%	0.09s - 0.17s	Higher computational efficiency and shorter response time	Fewer feature recognition points than YOLOv5-ECA
YOLOv5-ECA (This study)	98.5% - 99.1%	0.02s - 0.11s	High computational efficiency and short response time	May not accurately recognize extreme or rare gestures; significantly affected by hardware devices and environmental factors; higher model complexity

In summary, although current research has made some progress in gesture recognition and human-machine interaction, most systems still face low efficiency and insufficient accuracy in processing high-dimensional data. The research aims to develop an efficient and accurate gesture recognition system by combining You Only Look Once version 5 (YOLOv5) and Unity 3D software, in order to provide a more intuitive and efficient way of human-machine interaction. The innovation of the research lies in optimizing the YOLOv5 network structure by introducing Efficient Channel Attention (ECA) and designing a novel gesture recognition algorithm. Meanwhile, the study combines Unity 3D software to build a digital twin model of complex processing equipment, achieving real-time visualization of gesture data and human-machine

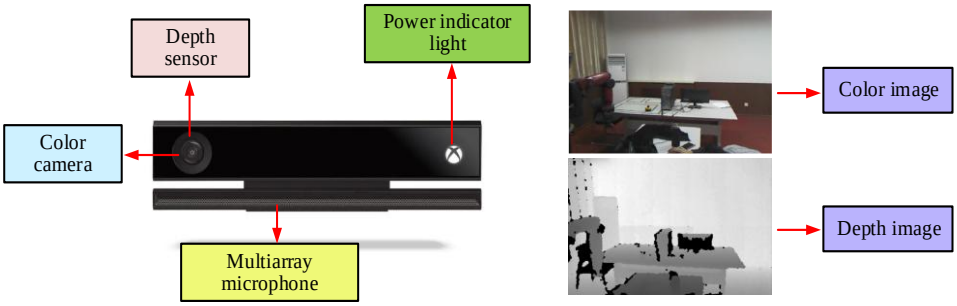
interaction.

2 Methods and materials

In order to achieve efficient and accurate gesture recognition in complex machining environments, the YOLOv5 algorithm is first optimized. An improved algorithm combining ECA is proposed. The study aims to collect gesture data through Kinect 2.0 sensors and introduce them into Unity 3D software to achieve real-time visualization and human-machine interaction of gesture data.

2.1 Design of twin static gesture recognition algorithm based on improved YOLOv5

Kinect is a motion sensing input device developed by Microsoft, first released in 2010. This device uses a series of sensors and cameras to capture player actions, voice, and images without the need for traditional game controllers, allowing users to interact with the game through body movements and voice commands [9-10]. Kinect 2.0 is an upgraded version of the first generation, which not only supports higher resolution color information and can detect infrared images, but also increases the number of detected joints from 20 to 25. The structure of Kinect 2.0 and the collected image information are shown in Figure 1.



(a) Kinect 2.0 structure (b) Color and depth images captured by Kinect

Figure 1: Kinect 2.0 structure and captured images

In Figure 1 (a), the key components of Kinect 2.0 include a color camera, depth sensor, multi-array microphone, and power indicator light. Color cameras are used to capture user's color images, which can directly display user images in games. Depth sensors use infrared projection technology to create a 3D spatial mapping of the player's surroundings, allowing devices to detect the user's position and actions in space, even in dimly lit environments. Multi-array microphones are used to capture sound and enable speech recognition functionality. Figure 1 (b) shows the color and depth images captured by Kinect 2.0. When using Kinect 2.0 to capture image information, the calculation between the camera and the measured object is shown in equation (1) [11-12].

$$d = c\frac{\Delta\varphi}{2\pi f} \tag{1}$$

In equation (1), d represents the distance between the measured object and the camera. c represents the speed of light. $\Delta\varphi$ represents the round-trip phase difference. f represents the given infrared light frequency. Due to the certain spatial spacing and different viewing angles between Kinect 2.0 color and depth cameras inside the device, the correspondence between the two types of images is not completely consistent when collecting gesture images. To successfully complete the static gesture recognition task, it is necessary to register the color gesture image with the depth gesture image. The coordinate relationship between the two images is shown in equation (2).

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = W \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + U \tag{2}$$

In equation (2), W and U represent the rotation matrix and translation matrix, respectively. $\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}$ and $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ represent the coordinates of corresponding points in color gesture images and depth gesture images, respectively. The calculation of transferring coordinate points from deep gesture images to color gesture images is shown in equation (3).

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_{a'} \times a'/c' \\ f_{b'} \times b'/c' \end{bmatrix} + \begin{bmatrix} c'_{a'} \\ c'_{b'} \end{bmatrix} \tag{3}$$

In equation (3), u and v represent the horizontal and vertical coordinates of a point in the color gesture image, respectively. $f_{a'}$ and $f_{b'}$ represent the proportional parameters corresponding to a' and b'. $\begin{bmatrix} c'_{a'} \\ c'_{b'} \end{bmatrix}$ represents the center point coordinates in the color gesture image.

In addition to using Kinect 2.0 to process image data, the study also introduces the YOLOv5 network to design

a gesture recognition algorithm. The core idea of YOLO is to view object detection as an end-to-end regression problem, achieving real-time object detection by dividing grids on the image and predicting the bounding boxes and categories of each grid [13-14]. YOLOv5 inherits the core concept of the YOLO series and improves performance and efficiency by optimizing network structure and data augmentation technology, as shown in Figure 2.

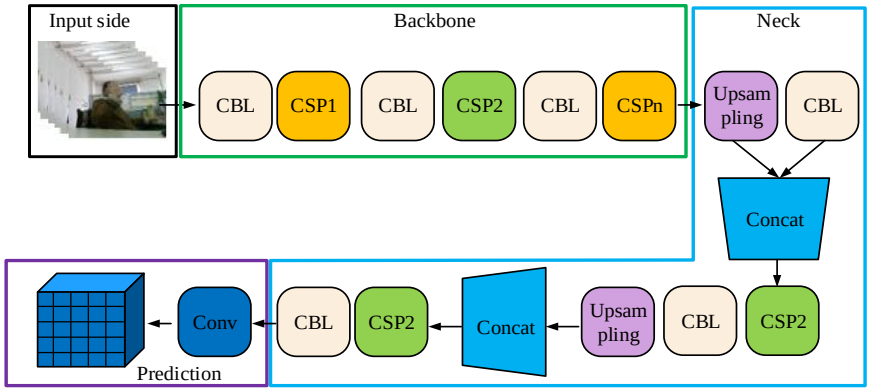


Figure 2: YOLOv5 network structure diagram

Figure 2 shows the main components of YOLOv5, including backbone module, neck module, head module, and prediction module. In YOLOv5, the sub-modules include a Cross Stage Partial Darknet53 (CSPDarknet53) with a Darknet53 neural network and a single Cross Stage Partial (CSP) network. Compared with other YOLO versions, YOLOv5 adopts a two-stage CSP structure, which can effectively reduce gradient information loss, reduce model size, and enhance the comprehensiveness of information extraction. Squeeze-and-Excitation Network (SENet) is a special channel attention mechanism module. In SENet, channel information can be obtained through global average pooling, and then the weight values of the channels can be obtained through learning, ultimately enhancing attention. The process of taking global average pooling to compress global spatial information is shown in equation (4) [15-16].

$$z = \frac{1}{H' \times W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} u'(i, j) \tag{4}$$

In equation (4), (i, j) represents the position coordinates of the input feature map. u' and z respectively represent the input and output of the SENet module, and their specific value ranges are shown in equation (5).

$$\begin{cases} u' \in R^{C' \times H' \times W'} \\ z \in R^{C' \times 1 \times 1} \end{cases} \tag{5}$$

In equation (5), R represents the set of real numbers. C', H' and W' respectively represent the number of channels, feature map height, and feature map width. In order to efficiently utilize the aggregated information in the channel, a learnable module is added to the SENet module to capture channel correlation. Two fully connected layers and a ReLU activation function are used to achieve this, as shown in equation (6).

$$s = \sigma\left(W_2 \delta\left(W_1 z\right)\right) \tag{6}$$

In equation (6), s represents the channel attention output. W_1 and W_2 represent two fully connected operations, respectively. σ represents the Sigmoid function, which limits the channel weight value between 0 and 1. δ represents the ReLU activation function. The range of values for s, W_1 and W_2 are shown in equation (7).

$$\begin{cases} s \in R^{C' \times 1 \times 1} \\ W_1 \in R^{\frac{C'}{r} \times C'} \\ W_2 \in R^{C' \times \frac{C'}{r}} \end{cases} \tag{7}$$

In equation (7), r represents the channel reduction coefficient. The final output of the SENet module obtained by combining equations (4) to (7) is shown in equation (8).

$$x = s \times u' \tag{8}$$

In equation (8), x represents the final output of the SENet module, and $x \in R^{C' \times H' \times W'}$. In the SENet module, in order to further enhance the prediction ability of channel attention on detailed features and reduce the parameters and computational complexity of the fully connected layer, the ECA module is used for improvement. The main reasons for choosing ECA are as follows. Firstly, ECA constructs channel attention through one-dimensional convolution, avoiding the information loss caused by dimensionality reduction and effectively preserving gesture image feature information. Secondly, it has a fast information processing speed, which can meet the real-time requirements of gesture recognition in complex processing environments. Furthermore, the length value determines the size of the receptive field, which in turn determines the effectiveness of attention acquisition, enabling it to

adaptively extract key features and improve the accuracy of gesture recognition. It is very suitable for high-precision human-computer interaction requirements in complex processing environments. The structure of ECA is shown in Figure 3.

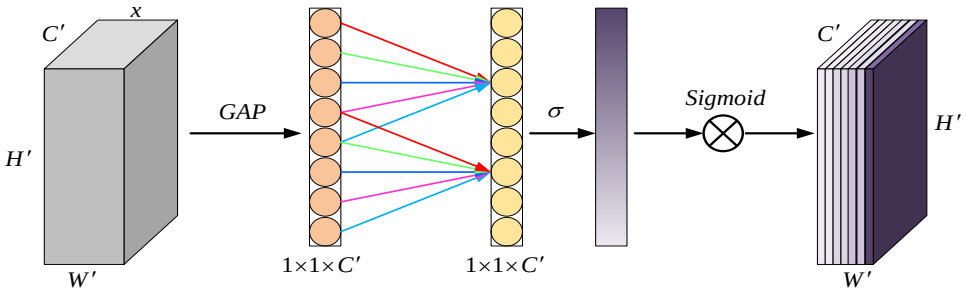


Figure 3: Structural diagram of ECA mechanism

In Figure 3, ECA effectively captures the feature information of local channels and constructs channel attention by using one-dimensional convolution instead of fully connected layers. This feature extraction method not only avoids information loss caused by dimensionality reduction, but also has faster information processing speed. The mathematical model of ECA is shown in equation (9).

$$s = \sigma\left(C1D_k\left(GAP\left(x'\right)\right)\right) \tag{9}$$

In equation (9), x' represents the input feature of ECA. $C1D_k$ represents the one-dimensional convolution with a convolution length of k. GAP represents the global average pooling. The final output of ECA is shown in equation (10).

$$y = s \times x' \tag{10}$$

In equation (10), y represents the final output of ECA. Due to the one-dimensional convolution method used in ECA, the length value determines the size of the receptive field, thereby determining the effectiveness of attention acquisition. A mapping relationship is constructed between k and C' to achieve adaptive convolution, as shown in equation (11).

$$k = \phi\left(C'\right) = \left|t\right|odd = \left|\frac{\log_2 C'}{\gamma} + \frac{\varepsilon}{\gamma}\right|odd \tag{11}$$

In equation (11), γ and ε represent two different hyper-parameters. $\left|t\right|odd$ represents the odd number closest to t. ϕ represents the mapping relationship. The ECA is integrated into the YOLOv5 network. Then, a YOLOv5 static gesture recognition algorithm (You Only Look Once version 5-Effective Channel Attention, YOLOv5-ECA) is ultimately designed. The running process of YOLOv5-ECA is shown in Figure 4.

The YOLOv5-ECA static gesture recognition process in Figure 4 is mainly divided into two parts: gesture segmentation and gesture recognition. In the gesture segmentation stage, Kinect 2.0 is mainly used to collect gesture image data and perform registration and segmentation operations on the collected images. In the gesture recognition stage, YOLOv5 and ECA are used to complete the recognition task. During the training of the YOLOv5-ECA model, the following hyperparameter settings are used. The learning rate is 0.01, and the dynamic adjustment strategy is used to gradually reduce the learning rate as the training rounds increased, in order to achieve better convergence. The batch size is set to 32 to balance the memory footprint and training efficiency. The optimizer selects Adam, which has the characteristic of adaptive learning rate and can adapt to the model training. By setting these hyperparameters, the training process can be effectively controlled, avoiding overfitting and underfitting problems, and improving the performance and generalization ability of the model.

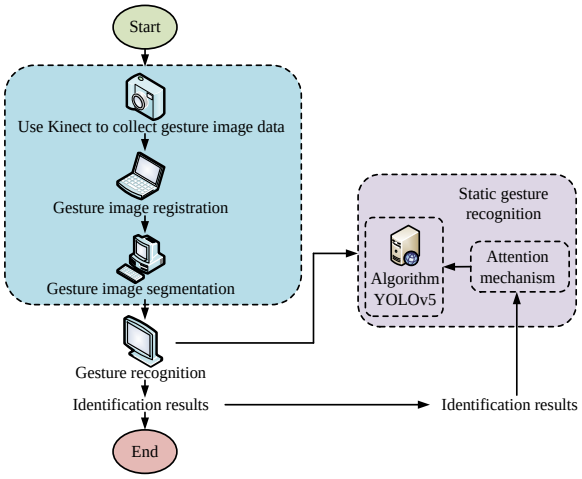


Figure 4: Flowchart of YOLOv5-ECA static gesture recognition

2.2 Construction of twin human-machine interaction model for complex processing equipment in gesture recognition

Complex machining equipment usually refers to mechanical equipment used in industrial manufacturing processes to perform various complex machining tasks. These devices typically have high precision, versatility, and high automation, which can handle complex process flows and production requirements. Typical complex processing equipment includes CNC machine tools, automated production lines, robot processing systems,

additive manufacturing equipment, etc [17-18]. Digital twin technology is used to establish twin models for complex processing equipment. This model not only reproduces the characteristics of various physical devices in virtual space, but also achieves bidirectional information exchange between entities and digital models by simulating the operational behavior of devices in real industrial environments. The twin model framework constructed is shown in Figure 5.

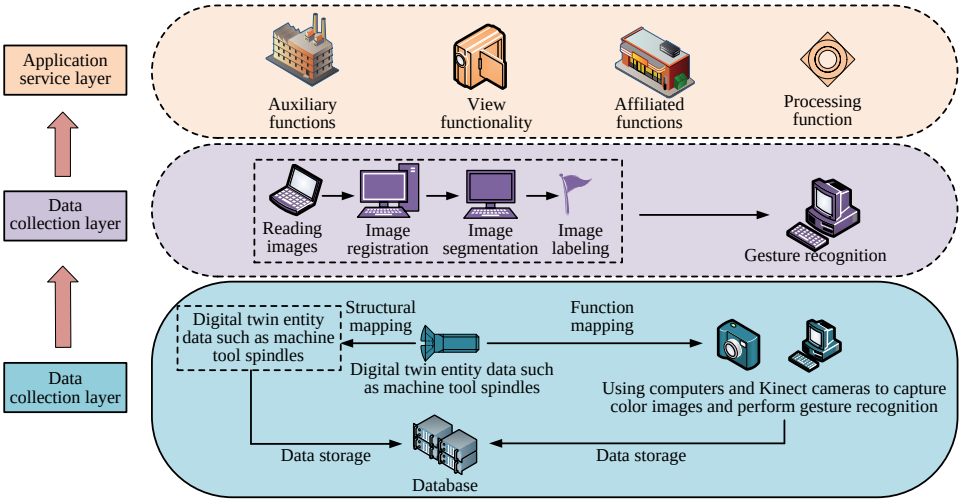


Figure 5: Framework diagram of twin model system for complex processing equipment

Figure 5 shows the system framework of the twin model for complex processing equipment, which is divided into three layers: data acquisition layer, data processing layer, and application service layer. The data collection layer forms the foundation of the system, which not only groups the functions of devices, but also associates these grouped device functions with operational gestures. In this layer, Kinect 2.0 sensors are mainly used to collect gesture data. The collected data includes depth and color images of static gestures obtained in diverse backgrounds and lighting environments. The main task of the data processing layer is to process the collected data. Due to the unsuitability of directly collected gesture data for static gesture recognition, a series of preprocessing is required. After these preprocessing steps are completed, the dataset can be used for gesture training and recognition. The application service layer is located at the top layer of the system. The processed data can interact with this layer to implement various functions. Overall, this study aims to project complex machining equipment into a virtual space and utilize Kinect 2.0 sensors to achieve diverse applications of the equipment in the virtual space.

In the field of industrial manufacturing, complex processing equipment plays an important role [19-20]. Its operational performance and efficiency have a decisive impact on product quality and production efficiency. To improve the operational efficiency and production output quality of these complex processing equipment, real-time monitoring and optimized control are two commonly used key strategies. Traditional monitoring and control methods rely heavily on human and material resources, which are often affected by errors and response delays. The advancement of artificial intelligence technology, especially the promotion of the Internet of Things and intelligent manufacturing, has made digital twin technology a new solution for simulating complex processing equipment. This study combines the optimized YOLOv5 algorithm to design gesture recognition technology. The human-machine interaction process and gesture interaction process in the complex processing equipment twin model after introducing gesture recognition are shown in Figure 6.

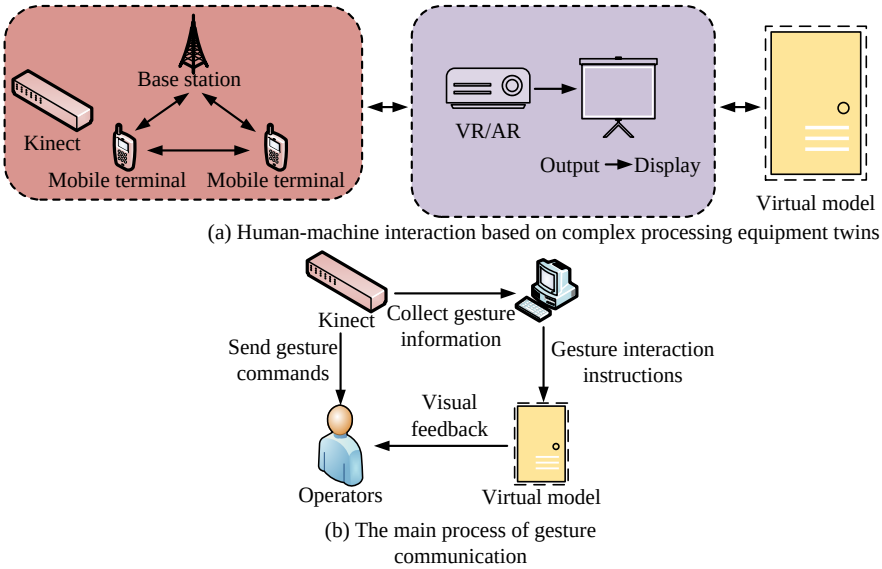


Figure 6: Flowchart of human-machine interaction and gesture interaction in the twin model

In Figure 6 (a), human-machine interaction based on the complex processing equipment twin model refers to creating a virtual model corresponding to the actual physical equipment. Through reliable communication technology, the operator's gesture instructions are transmitted, and the device can adjust its working status in a timely manner based on these instructions to meet the operator's requirements and complete production tasks. When building virtual models, Kinect 2.0 and Unity 3D software are mainly used. For the details of building a virtual model, the first step is to import the gesture data collected by Kinect 2.0 into Unity 3D. By writing scripts, these gesture data can be mapped to the corresponding actions of the virtual model. For example, when the operator makes a gesture, the virtual device in Unity 3D takes the corresponding action to simulate the working state of the real device. Secondly, the communication function of Unity 3D ensures synchronization between the virtual model and the actual device. Based on network protocols, operation instructions are transmitted from virtual environments to actual devices, enabling them to respond promptly to operator instructions. Finally, Unity 3D also supports rich user interface design, providing operators with an intuitive control panel and feedback interface. In a virtual environment, operators can understand the status and operation results of devices through an intuitive interface, improving the efficiency and accuracy of operations. Figure 6 (b) shows the flowchart of gesture interaction. In the gesture interaction, the operator first executes various gestures, and the Kinect sensor captures this gesture information and sends it to the computer system. Subsequently, the computer system parses the data and outputs the recognized gesture results and corresponding instructions. Finally, the digital twin model of complex processing equipment immediately operates based on these instructions. Throughout the entire gesture interaction cycle, operators monitor and adjust actions through visual feedback to ensure accurate execution of gesture commands and interaction continuity.

3 Results

Firstly, the study selects Single Shot Multi-Box Detector (SSD), YOLOv5, and Enhanced Multi-Scale Attention Feature Fusion Network (EMAFF-Net) as comparative algorithms to test the benchmark performance of YOLOv5-ECA algorithm. Secondly, four algorithms are used to construct recognition models to verify the effectiveness of YOLOv5-ECA in practical applications.

3.1 YOLOv5-ECA algorithm performance testing

EgoHands is a publicly available gesture recognition dataset designed specifically for first person perspective gesture recognition, which is used to test the benchmark performance of algorithms. The EgoHands dataset contains various gesture types, such as common gestures such as pointing, grasping, and clenching, totaling approximately 3,000 publicly available gesture image data. In terms of variability, it covers different lighting conditions, ranging from bright to dim environments, and user diversity includes people of different ages, genders, and skin colors. In the preprocessing step, the image is first subjected to size normalization and uniformly adjusted to a specific size, such as 416×416 pixels. Simultaneously, data augmentation operations are performed, including random rotation of a certain angle (such as ±15°), random horizontal flipping, etc. The collected 3,000 public gesture image data are divided into training and testing sets in an 8:2 ratio. Firstly, the loss values of four algorithms are tested on the same dataset, as shown in Figure 7.

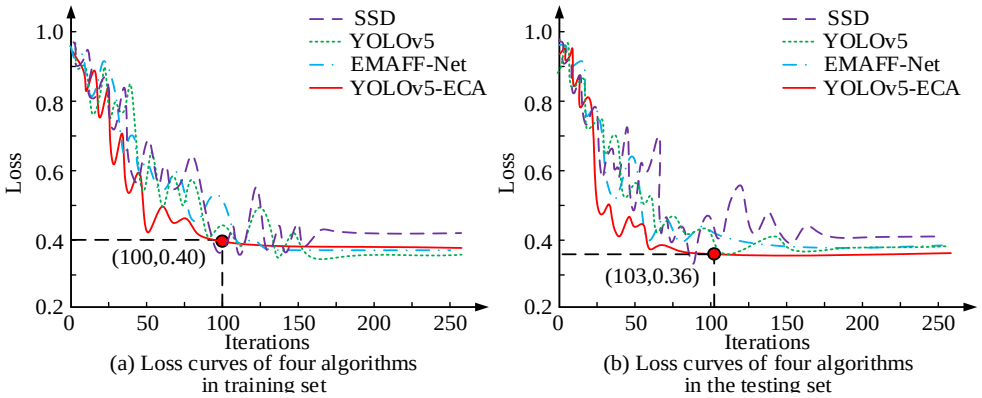


Figure 7: Performance of YOLOv5 ECA in gesture recognition: A comparative study of loss function and accuracy

Figures 7 (a) and 7 (b) show the loss function curves of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA in the training and testing sets, respectively. As shown in Figure 7 (a), YOLOv5-ECA iterated to a stable state faster than the other three algorithms. After reaching a stable state, YOLOv5-ECA had 100 iterations, with a loss value of 0.40. Similarly, in Figure 7 (b), YOLOv5-ECA only required 103 iterations to reach a stable state, with a loss value of 0.36. The p-value of the accuracy difference between YOLOv5-ECA and SSD was 0.01 and the t-value was 3.5, indicating that the difference in performance of the two algorithms was significant at the significance level of 0.05. For the comparison of YOLOv5-ECA and EMAFF-Net, with a p-value of 0.03 and a t-value of 3.2, the differences were also considered significant. These statistical results support that the superior performance of YOLOv5-ECA in loss function and accuracy is not accidental. Then, the study tests the Image Ambiguity (IA) and Structural Similarity Loss (SSL) of the four algorithms during the training process, as shown in Figure 8.

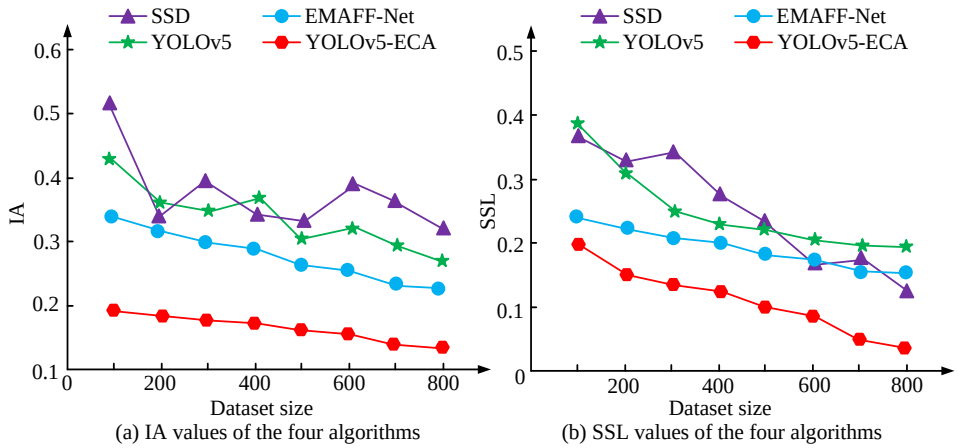
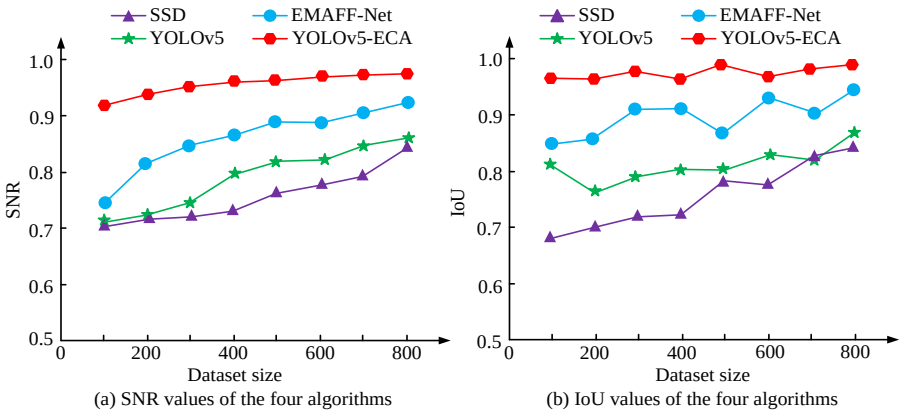


Figure 8: Performance statistical analysis of gesture recognition algorithms: IA, SSL, and accuracy of different algorithms

Figures 8 (a) and 8 (b) show the changes in IA and SSL values of the four algorithms during training, respectively. As shown in Figure 8 (a), when the number of training samples increased from 100 to 800, the IA values of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA decreased from 0.52, 0.43, 0.34, and 0.19 to 0.36, 0.28, 0.25, and 0.07, respectively. The IA value under the YOLOv5-ECA algorithm was always less than 0.20, indicating that the algorithm had the lowest ambiguity in recognizing gesture images. As shown in Figure 8 (b), the SSL values of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA algorithms also decreased with the increase of sample size. When the sample data were 800, the SSL values of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA reached their minimum values of 0.15, 0.23, 0.18, and 0.03, respectively. It can be seen that the YOLOv5-ECA algorithm has the smallest image structural information loss during the training process, which can better preserve the true recognition results. Then, the mean and standard deviation of each model in multiple experiments are calculated. For example, the YOLOv5 ECA model had an accuracy of 98.5%, 98.7%, 99.1%, and 98.8% in recognizing four types of gesture images, respectively.

After multiple experiments, its mean was 98.75% and the standard deviation was 0.2%. Similar processing is also applied to response time. For example, YOLOv5-ECA had a minimum response time of 0.02s. The average value after multiple experiments was 0.06s, the standard deviation was 0.01s, and the p-value was less than 0.0001. By calculating the confidence interval, the significance of model performance improvement can be more accurately determined. The changes in Signal-to-Noise Ratio (SNR) and Intersection over Union (IoU) of the four algorithms during the training process are shown in Figure 9.



(a) SNR values of the four algorithms (b) IoU values of the four algorithms

Figure 9: Empirical study on the effect of sample size on SNR and IoU values of SSD, YOLOv5, EMAFF-Net and YOLOv5-ECA algorithms

Figures 9 (a) and 9 (b) show the SNR and IoU values of the four algorithms, respectively. In Figure 9, as the sample size continued to increase, the SNR and IoU values of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA also showed a gradually increasing trend. However, the overall increase trend of YOLOv5-ECA was the gentlest, and the changes in its SNR and IoU values were also the smallest. As shown in Figure 9 (a), the maximum SNR values of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA were 0.82, 0.84, 0.91, and 0.95, respectively. As shown in Figure 9 (b), the maximum IoU values of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA were 0.83, 0.86, 0.94, and 0.98, respectively. After calculation, the number of floating-point operations for SSD was 1045 FLOPs. YOLOv5 was relatively more complex in structure, with 1513FLOPs. Due to its multi-scale attention feature fusion mechanism, EMAFF Net has a higher computational complexity of approximately 2120FLOPs. YOLOv5-ECA introduces ECA mechanism and interaction with Unity 3D, further increasing the computational complexity to 2502FLOPs. This indicates that YOLOv5-ECA faces a relatively high computational burden while achieving high performance. However, in complex machining environments, its high-precision recognition performance may balance performance and computational costs to some extent.

3.2 Practical application effects of human-machine interaction models considering gesture recognition

In addition to testing the benchmark performance of four algorithms, SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA algorithms are applied to complex processing equipment twin models. Four different types of static gesture interaction recognition models are constructed. Four different static gestures are captured to detect the performance of the four models in practical applications, as shown in Figure 10.

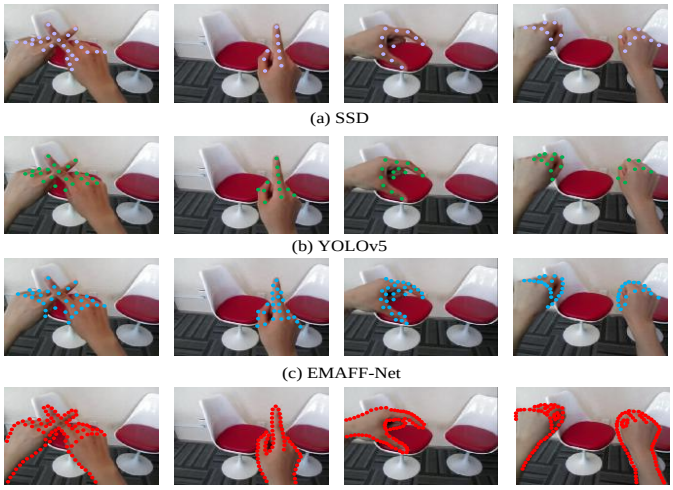


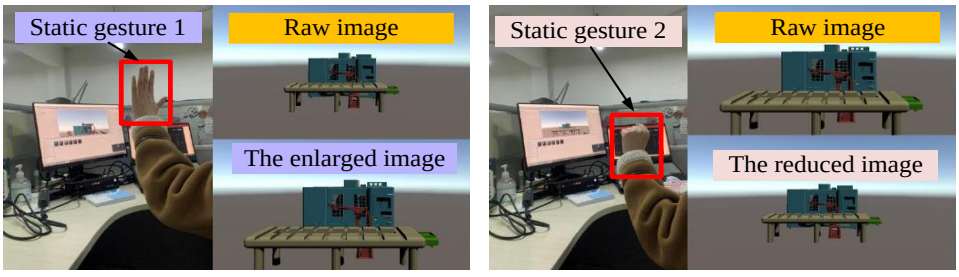
Figure 10: Recognition effects of different models

Figures 10 (a), 10 (b), 10 (c), and 10 (d) show the recognition performance of SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA models for four types of gesture images, respectively. Based on Figure 10, the YOLOv5-ECA model had the best recognition performance. This model fully recognized key points in different gestures and provided a complete gesture recognition trajectory. Secondly, EMAFF-Net had better recognition performance than SSD and YOLOv5, but its number of feature recognition points was less than YOLOv5-ECA model, so its recognition performance ranked second. The recognition performance of SSD and YOLOv5 was poor, because both models were basic models and lack additional attention mechanism structures to increase the recognition accuracy of detailed features.

Table 2 shows the accuracy and time of four models in recognizing four types of gesture images. According to the data in Table 2, the accuracy of YOLOv5-ECA in recognizing four types of gesture images was above 98%, with the highest reaching 99.1%, far higher than SSD and YOLOv5. In addition, YOLOv5-ECA had a shorter recognition time for the four types of images, with the shortest being as low as 0.02s. The interaction effect of the YOLOv5-ECA model in the twin system of complex processing equipment is tested, as shown in Figure 11.

Table 2: Actual recognition accuracy and recognition time of the four models

Image number	Network structure	Accuracy/ %	Time/ s
Gesture image 1	SSD	85.6%	0.33s
	YOLOv5	88.9%	0.26s
	EMAFF-Net	91.7%	0.14s
	YOLOv5-ECA	98.5%	0.07s
Gesture image 2	SSD	86.8%	0.23s
	YOLOv5	88.9%	0.15s
	EMAFF-Net	92.2%	0.09s
	YOLOv5-ECA	98.7%	0.02s
Gesture image 3	SSD	88.2%	0.29s
	YOLOv5	90.3%	0.22s
	EMAFF-Net	93.1%	0.17s
	YOLOv5-ECA	99.1%	0.11s
Gesture image 4	SSD	84.5%	0.31s
	YOLOv5	87.6%	0.26s
	EMAFF-Net	90.4%	0.14s
	YOLOv5-ECA	98.8%	0.08s



(a) The interface amplifies the interaction results

(b) The interface Narrows the interaction results

Figure 11: Static gesture interaction results of YOLOv5-ECA model

In Figure 11 (a), when the operator's gesture was to open the palm, the image of the complex processing equipment twin system in Unity 3D was larger. In Figure 11 (b), when the operator's gesture was to merge the

palms, the image of the complex processing equipment twin system in Unity 3D shrank. Based on the interaction results in Figures 11 (a) and 11 (b), it can be concluded that the designed static gesture recognition model can effectively complete human-machine interaction instructions. Due to its high precision and fast response time in various gesture recognition tasks, YOLOv5-ECA demonstrates excellent foundational performance. For new types of gestures or industrial devices, the model can be fine tuned with a small amount of annotated data to quickly adapt to new application scenarios without the need for large-scale training from scratch, thus saving time and resources. In addition, the high efficiency and low latency characteristics of YOLOv5-ECA make it particularly suitable for real-time interactive systems, such as the human-computer interaction scenario shown in Figure 11. Faced with dynamic gesture recognition, the model's fast updating ability and robustness can also ensure smooth transitions and provide stable and reliable recognition results.

4 Discussion

The study selected SSD, YOLOv5, EMAFF-Net, and YOLOv5-ECA for comparison. In performance testing, YOLOv5-ECA reached a stable state faster by iterating on a dataset of 3,000 gesture images. When the training set was stable, the loss value was 0.40 after 100 iterations, and reached 0.36 after 103 iterations on the testing set. The lowest image blur was 0.07, the minimum structural similarity loss was 0.03, the average accuracy was 98.75%, and the average response time was 0.06 seconds. The highest signal-to-noise ratio and intersection to union ratio were 0.95 and 0.98, respectively, but the computational complexity reached 2502FLOPs.

The YOLOv5-ECA method proposed in this study shows significant advantages in gesture recognition, and its performance is significantly improved compared with baseline methods (SSD, YOLOv5, EMAFF-Net). Firstly, from the perspective of performance improvement, the ECA mechanism has played a crucial role in feature extraction. Traditional feature extraction methods may overlook local feature information between channels, while ECA mechanism effectively captures the feature information of local channels by one-dimensional convolution to construct channel attention.

Secondly, Unity 3D has played an important role in enhancing visualization and interaction. It can map the gesture data collected by Kinect 2.0 to the corresponding actions of the virtual model, achieving real-time visualization of gesture data and human-computer interaction. Through scripting and communication capabilities, Unity 3D ensures synchronization between virtual models and actual devices, providing operators with intuitive control panels and feedback interfaces, and further improving the efficiency and accuracy of human-computer interaction.

However, this method also has some potential limitations. Although YOLOv5-ECA performs well in known datasets and experimental environments, there may be issues with inaccurate recognition for some extreme or rare gesture situations. This is because the training data may not fully cover all possible gesture variations and complex scenarios. Meanwhile, the performance of the model may be affected by hardware devices and environmental factors. For example, in extremely poor lighting conditions or in the presence of occlusion, the image quality captured by Kinect 2.0 may decrease, thereby affecting the input data quality of the model and leading to a decline in recognition performance. In addition, the complexity of the model is relatively high, and it may face slow running speed on devices with limited computing resources. In future research, it is necessary to further optimize the preprocessing process of the scheme and expand the scope of data collection to enhance the generalization ability of the model. At the same time, although this algorithm increases accuracy, it also increases the complexity of the model.

5 Conclusion

In order to improve the accuracy of human-machine interaction gesture recognition in complex processing equipment, a YOLOv5-ECA model was designed by combining ECA and YOLOv5. The experimental results showed that the model significantly outperformed SSD, YOLOv5, and EMAFF-Net on accuracy and real-time performance in gesture recognition. In benchmark performance testing, the model had a faster iteration speed and lower IA and SSL values. It also had excellent performance in SNR and IoU, with higher SNR and IoU values. In practical applications, YOLOv5-ECA exhibited high recognition accuracy and low response time in digital twin systems of complex processing equipment, with a maximum recognition accuracy of 99.1% and a minimum response time of only 0.02s. In summary, the YOLOv5-ECA model performs well in basic testing, achieving excellent detection results in practical applications. Subsequent research can further test the performance of the YOLOv5-ECA model in different scenarios and other recognition tasks to improve the model's generalization ability. However, there are some limitations to using the Kinect 2.0 sensors for gesture recognition. Under low-light conditions, the image quality collected by Kinect 2.0 may decrease, affecting the accuracy of gesture recognition. In addition, occlusion problems can also have adverse effects on the system. When some gestures are blocked, the complete gestures may not be accurately identified. These shortcomings may reduce the applicability of systems in complex environments. For example, in some low-light industrial scenarios, the accuracy of gesture recognition decreases, affecting the efficiency of human-computer interaction. In practical application, these limitations need to be considered. Some measures such as adding auxiliary lighting or optimizing the algorithm to cope with the occlusion situation can improve the stability and applicability of the system.

Fundings

The research is supported by university-level Research Project: Research on the Design of Shared Stalls and Interactive Facilities Based on AEIOU Framework (2024Y07).

Conflict of interest

The author states no conflict of interests.

Data availability statement

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

[1] Chua S N D, Chin K Y R, Lim S F, Jain P. Hand gesture control for human-computer interaction with Deep Learning. Journal of Electrical Engineering & Technology, 2022, 17(3): 1961-1970.

[2] Moin A, Aadil F, Ali Z, Kang D. Emotion recognition framework using multiple modalities for an effective human-computer interaction. The Journal of Supercomputing, 2023, 79(8): 9320-9349.

[3] Li P, Zhao L. A novel art gesture recognition model based on two channel region-based convolution neural network for explainable human-computer interaction understanding. Computer Science and Information Systems, 2022, 19(3): 1371-1388.

[4] Gams M, Kolenik T. Relations between electronics, artificial intelligence and information society through information society rules. Electronics, 2021, 10(4): 514.

[5] Yadav K S, Kirupakaran A M, Laskar R H. End-to-end bare-hand localization system for human-computer interaction: a comprehensive analysis and viable solution. The Visual Computer, 2024, 40(2): 1145-1165.

[6] Zhang F. Human-Computer Interactive Gesture Feature Capture and Recognition in Virtual Reality. Ergonomics in Design, 2021, 29(2): 19-25.

[7] Li Q, Langari R. Myoelectric human computer interaction using CNN-LSTM neural network for dynamic hand gesture recognition. Journal of Intelligent & Fuzzy Systems, 2023, 44(3): 4207-4221.

[8] Chakravarthi S S, Rao B, Challa N P, Ranjana R, Rai A. Gesture Recognition for Enhancing Human Computer Interaction. Journal of Scientific & Industrial Research, 2023, 82(4): 438-443.

[9] Hu D, Zhu J, Liu J, Wang J, Zhang X. Gesture recognition based on modified Yolov5s. IET image processing, 2022, 16(8): 2124-2132.

[10] Ying Z, Lin Z, Wu Z, Liang K, Hu X. A modified-YOLOv5s model for detection of wire braided hose defects. Measurement, 2022, 190(2): 2-12.

[11] Zhang Q, Wang Y, Song L, Han M, Song H. Using an improved YOLOv5s network for the automatic detection of silicon on wheat straw epidermis of micrographs. Journal of Field Robotics, 2023,40(1): 130-140.

[12] Li C, Zhao G, Gu D, Wang Z. Improved lightweight YOLOv5 using attention mechanism for satellite components recognition. IEEE Sensors Journal, 2022, 23(1): 514-526.

[13] Xue J, Zheng Y, Dong-Ye C, Wang P, Yasir M. Improved YOLOv5 network method for remote sensing image-based ground objects recognition. Soft Computing, 2022, 26(20): 10879-10889.

[14] Li X, Luo R, Islam F U. Tracking and detection of basketball movements using multi-feature data fusion and hybrid YOLO-T2LSTM network. Soft Computing, 2024, 28(2): 1653-1667.

[15] Haq M A, Tagawa N. Improving Badminton Player Detection Using YOLOv3 with Different Training Heuristic. JOIV: International Journal on Informatics Visualization, 2023, 7(2): 548-554.

[16] Bian L, Li B, Wang J, Gao Z. Multi-branch stacking remote sensing image target detection based on YOLOv5. The Egyptian Journal of Remote Sensing and Space Sciences, 2023, 26(4): 999-1008.

[17] Hanafi W, Tamali M. Implementing distributed collaboration and applying the YOLO algorithm to robots. Studies in Engineering and Exact Sciences, 2024, 5(1): 277-296.

[18] Mukai N, Suzuki M, Takahashi T, Mae Y, Arai Y, Aoyagi S. Application of Object Grasping Using Dual-Arm Autonomous Mobile Robot—Path Planning by Spline Curve and Object Recognition by YOLO—. Journal of Robotics and Mechatronics, 2023, 35(6): 1524-1531.

[19] Zhou W, Li X. PEA-YOLO: a lightweight network for static gesture recognition combining multiscale and attention mechanisms. Signal, Image and Video Processing, 2024, 18(1): 597-605.

[20] Hu D, Zhu J, Liu J, Wang J, Zhang X. Gesture recognition based on modified Yolov5s. IET image processing, 2022, 16(8): 2124-2132.