Rubber Tyred Gantry Automatic Alignment System based on Computer Vision

Jianbing Yang[1], Na Zhu[1,*], Haojun Ji[1], and Bin He[2]
[1]Shanghai Zhenhua Port Machinery Company Limited(ZPMC), Shanghai, China
[2]The 15th Research Institute of China Electronics Technology Group Corporation, Beijing, China
yangjianbing@zpmc.com, zhuna@zpmc.com, jihaojun@zpmc.com, binhe.cas@foxmail.com
*corresponding author

***Abstract*—The existing rubber tyred gantry (RTG) mainly adopts methods such as traditional laser, magnetic nail or other technologies to orient. In view of the shortcomings of the technology, such as single use scenario, limited information and high construction cost, a visual automatic alignment system based on Mask-RCNN is proposed to accomplish the integration of perception and decision-making. The system based on ROS (Robot Operating System) is mainly composed of target detection and segmentation module, motion decision module and visual human- machine interaction interface. The target detection and segmentation module is mainly used to perceive the target, including its foreground segmentation module based on Mask-RCNN and the target orientation module based on traditional image algorithm; The motion decision module is mainly used to control the movement of the tire crane, including motion modeling module and logic control module; The visual human-computer interaction interface is mainly used to adjust system parameters and monitor system operation status, including the back-end used for providing data and the front-end used for data interaction. Finally, the designed visual automatic alignment system has been tested on the tire crane, and the orientation accuracy and operation efficiency have been improved enormously.**

Keywords- RTG; automatic alignment; computer vision; Mask-RCNN

I. INTRODUCTION

Rubber tyred gantry (RTG) is a common port machinery and equipment, which is widely used in ports around the world due to its low requirements on the operation site [1]. Perceptual positioning is one of the keys to the automation of RTG. For a long time, RTG mainly use traditional laser, magnetic nail and other technologies to achieve positioning. The advantages of this technology are simple and reliable, but the disadvantages are that the use scene is single, the information provided is limited, and the construction cost is high.

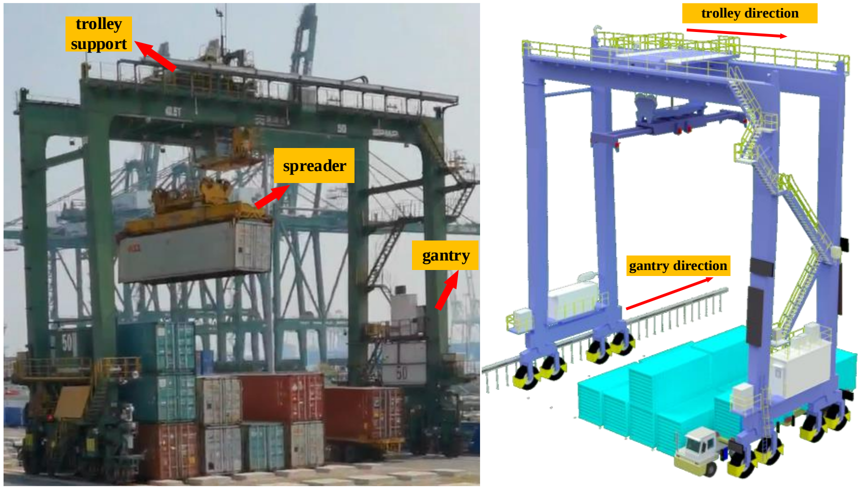
In recent years, the transformation to intelligent port has become an inevitable trend in the industry, and it is difficult to adapt to the current development needs only by using traditional technologies. Under the research of global scholars, the perception technology of using convolutional neural network [2-5] to extract image features has become the mainstream. Image processing technology based on deep learning has the advantages of low sensor requirements, rich perceptual information, and effectively reducing production costs, which makes the technology gradually become a new solution.

Mainstream perception models can be divided into two categories: single-stage object detection and two-stage object detection. Single-stage target detection is directly from the image to the target, such as YOLO[6-8], SSD[9], RetinaNet[10], etc., which are characterized by high speed and low accuracy. Two-stage target detection needs to be performed in two steps. First, the candidate region is obtained, and then the target in the candidate region is classified and regressed, such as the R-CNN series [11-14], which is characterized by high accuracy and slow speed. RTG belongs to heavy machinery, which usually needs to lift dozens of tons of goods. Ensuring its safety is the primary priority. Therefore, in the selection of the baseline of perception algorithm research, this article chooses the two-stage target detection algorithm with higher accuracy, and selects the detection and segmentation algorithm based on Mask-RCNN [14] to achieve the localization and classification of targets.

Another key to realizing port machinery automation is motion decision-making. At present, position-based visual control is widely used at home and abroad [15]. In the 3D Cartesian coordinate system, the camera calibration and coordinate transformation are used to obtain the position of the measured object in the Cartesian space, and form a deviation from the desired position. Then, according to the deviation, the control algorithm is designed to control the motion of the mechanism. This method has the following disadvantages: 1) No matter how the camera is calibrated, the coordinate conversion will bring certain errors; 2) The port machinery is a heavy machinery, and the machine will be deformed when hoisting dozens of tons of goods, which will also bring errors to the coordinate conversion. Due to the existence of errors, the parameters of the port machinery need to be adjusted frequently. Therefore, a large number of debugging personnel have to be deployed on the operation site to adjust the parameters, which increases the operation and maintenance cost. Aiming at this problem, this article proposes a novel motion modeling scheme.

II. SYSTEM OVERVIEW

RTG is a crane used for container hoisting, and its main structure consists of three parts: gantry, trolley and spreader, as shown in Figure 2. Among them, the gantry is equipped with tires, which can move in the direction of each row of containers; the trolley moves in the direction of each row of containers; the spreader is connected to the trolley frame with steel wire ropes. Through the movement of gantrys and trolleys, the spreader can lift containers in any position in the yard. At the same time, the spreader has the micro-movement function along the direction of the gantry, the direction of the trolley and the rotation, which is convenient for the micro-motion adjustment of the spreader during the alignment operation.

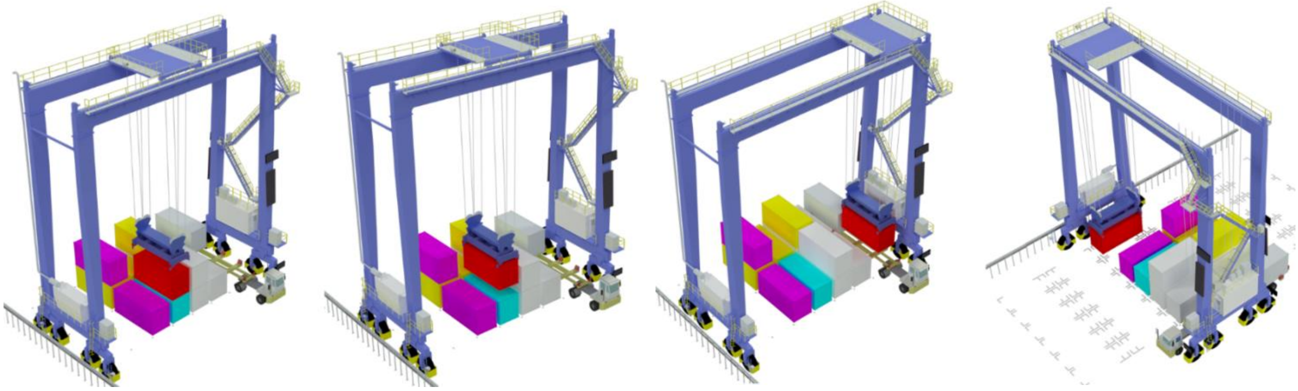


a）physical map

b）schematic diagram

Figure 1.RTG Structure

During the operation of RTG, the automatic control system needs to complete the alignment operation, such as the spreader and the container (grab the container, as shown in Figure 2a)), the container and the container (stacked containers, as shown in Figure 2b)), Containers and trucks (loading, as shown in Figure 2c)), containers and container corner lines (hoisting the first layer of containers in the yard, this action is called bottom container stacking, as shown in Figure 2d) alignment work.



a) grasping

b) stacking

c) loading

d) bottom container stacking

Figure 2. Schematic Diagram Of Alignment Operation

This article will take the bottom container stacking operation as an example to introduce the designed automatic visual alignment system for RTG, which is mainly composed of object detection and segmentation module, motion decision module and visual human- machine interaction interface, as shown in Figure 3. The target detection and segmentation module is mainly used to obtain the position of the target in the image. First, the target image is obtained through the image acquisition module, and then Mask-RCNN segments the foreground image of the target. Finally, the target positioning module performs post-processing on the foreground image and outputs the position of the target in the image. The motion decision module receives the result output by the target detection and segmentation module, and outputs the control instructions required by the tire crane control system. Firstly, the motion modeling of the tire crane alignment operation is carried out, then the deviation is calculated according to the motion model and the position of the operation target in the image, and finally the logic control command is output according to the operation process of the bottom container stacking. Both the target detection and segmentation module and the motion decision module are adjusted through the web human-computer interface.

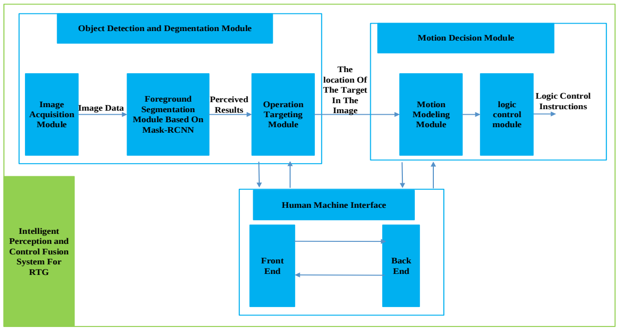

Object Detection and Degmentation Module
Image Acquisition Module
Image Data
Foreground Segmentation Module Based On Mask-RCNN
Perceived Results
Operation Targeting Module
The location Of The Target In The Image
Motion Decision Module
Motion Modeling Module
logic control module
Logic Control Instructions
Human Machine Interface
Front End
Back End
Intelligent Perception and Control Fusion System For RTG

Figure 3. System Overall Block Diagram

III. Object Detection And Segmentation Module

A. *Image Acquisition Module*

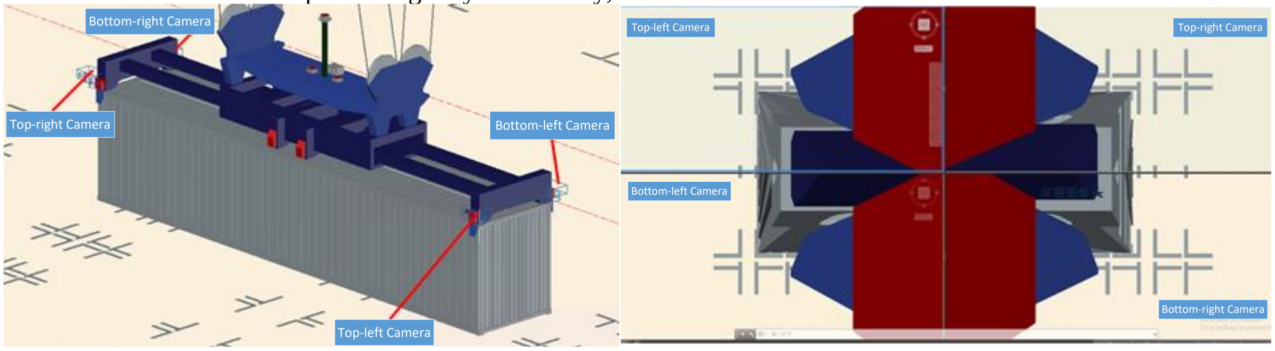
Four cameras are installed independently on the four corners of the spreader to collect the image of the operation target (for the bottom container stacking operation, the operation target is the container angle line), as shown in Figure 4a). The camera shooting angle is shown in Figure 4b). Since the four cameras work independently, in order to ensure that the four cameras capture images synchronously, the soft synchronization method is adopted in this article. The specific methods are as follows:

1) The four cameras capture images at a fixed frame rate with time stamps, and the collected images are stored in;

2) Take one of the cameras as the benchmark to obtain the latest image in its buffer;

3) In the buffers of the other three cameras, select the image with the closest time to the image;

4) These four images are selected as a batch, which is used as the input of the neural network.



a) installation diagram of spreader camera b) schematic diagram of the four-camera shooting angle

Figure 4. Installation Diagram Of Spreader Camera

B. *Task Target Foreground Segmentation Module Based On Mask-RCNN*

（1）Mask-RCNN Perception Model

In this article, the detection and segmentation algorithm of Mask-RCNN [14] is used to realize the foreground segmentation of container corner lines. Target detection is to locate the target with a rectangular frame, and its position in the image is determined by only two points (the top left point and the bottom right point), and the target segmentation is to divide different targets from the perspective of pixels. A mask to identify an object whose position in the image is determined by a set of points (ie, the coordinates of each pixel). Using object detection to locate objects in a continuous image sequence has a disadvantage: the detection container will shake. The jitter of the detection signal is not conducive to the alignment control of the tire crane. Compared with object detection by two points, object segmentation by a group of points is more stable.

The Mask-RCNN network is a two-stage object detection and segmentation network. The first stage extracts image features and generates proposals, and the second stage classifies proposals and generates bounding containers and masks. It mainly consists of the following three parts:

1) Backbone. In this article, the residual network (Resnet50) is used to extract features from the image, and then the Feature Pyramid Network (FPN) is used for multi-scale feature fusion;

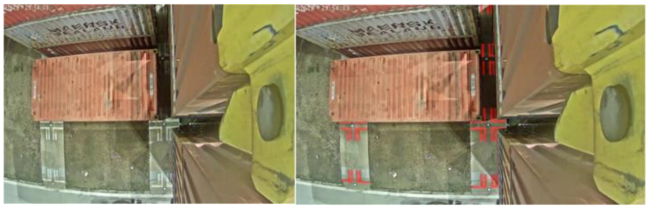
2) Region Proposal Network (RPN). The network structure is similar to the RPN in Faster-RCNN, the only difference is that the input is the output of the FPN. The network judges the positive and negative samples of each anchor feature and roughly locates it;

3) Region of Interest Align (RoI Align). Compared with RoI Pooling, RoI Align uses the bilinear interpolation method to calculate the value of each feature point, which realizes the continuity of the feature aggregation process. Pixel position shift is avoided when performing Mask regression.

（2）Data Set

The training data set adopts the MS-COCO data set and the container angle data set collected for the bottom-opening operation. MS-COCO (Common Objects in COntext) is one of the most commonly used datasets in the field of computer vision [16]. It has 80 categories and more than 200,000 annotated images, which can be used for research such as object detection, segmentation and semantic understanding. In this article, the MS-COCO data set is used to pre-train the perception model, so that the convolutional neural network has a certain awareness of the target, and then the container angle data set is used to fine-tune the perception model to improve the recognition and positioning accuracy.

The container corner line data set is mainly composed of two parts: 1) The corner line image data collected from multiple ports. This data set is used as a general detection and segmentation data set for corner line objects, with a total of 10,838 images, of which 9,758 are training sets, the validation set is 1080; 2) In the port scene of the actual bottom container stacking operation, the corner line image data during the bottom container stacking operation are collected in a targeted manner. There are 6028 images in total, including 5004 images in the training set and 1024 images in the validation set. When marking, all container corner lines appearing in the figure should be marked, no matter whether it is blocked or not, as shown in Figure 5.



a) original image b) annotated image

Figure 5. Example of Container Corner Line Data Set Annotation

（3）Training

During training, we keep the hyperparameters in Mask-RCNN unchanged. The difference from the original text is that in this article, in order to improve the real-time performance of the model, the resolution of the input image is scaled to 640×640.

Initialization: For the backbone of Mask-RCNN, ImageNet1k pretrained weights are used for initialization [17]. For both one-stage and two-stage classification and regression networks, weights with bias b = 0 and Gaussian variance σ = 0.01 are used as initialization.

Optimizer: This article uses a Stochastic Gradient Descent (SGD) optimizer to train the entire network, where the momentum is set to 0.9 and the weight decay is set to 0.0001. During the training process, the batch size is set to 4, and it is evenly distributed to two 2080 Ti graphics cards. The initial learning rate is set to 0.001, and as training progresses, the learning rate is divided by 10 at the 4K, 6K, and 8K iterations to achieve a dynamic decrease in the learning rate. The training process is iterated 10K times in total. During training, only horizontal flips of images are employed to augment the dataset. (Note: The above training process is pre-training with MS-COCO. During the fine-tuning stage, the learning rate is set to 0.0001 throughout, and the rest remain unchanged).

（4）Experimental Results

On the container corner target general detection segmentation dataset, the AP accuracy reaches 96.04%, and on the fine-tuning dataset for the actual open bottom condition, the AP accuracy reaches 99.49%. The experimental results are shown in Figure 6.

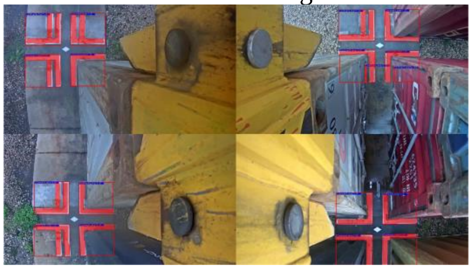


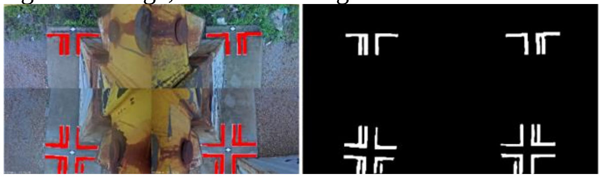
Figure 6. Container Corner Detection And Segmentation Results

C. Target Positioning Module

The target foreground segmentation module based on Mask-RCNN only outputs the segmentation result, and does not output the position of the expected target in the image. In this article, the traditional image processing method is used to analyze the perception results, and output the precise position of the target in the image.

（1）Acquisition of container corner foreground image

Save the container line mask obtained by the target foreground segmentation module to the same image and perform binarization processing to obtain the corner line foreground image, as shown in Figure 7.



a) corner line mask map b) corner line foreground binary map

Figure 7. Binary Image Of Corner Line Foreground

（2）Canny edge detection

Firstly, perform morphological processing on the obtained corner line foreground binary image, the specific performance is as follows:

1) perform an erosion operation to remove some small noise spots;

2) perform an expansion operation to connect the adjacent areas to enhance the corner line completeness.

Finally, canny edge detection is performed on the preprocessed foreground image, and the minimum bounding rectangle of the contour is obtained, as shown in Figure 8.

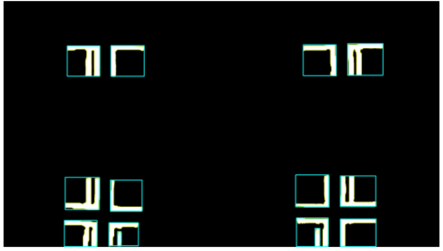


Figure 8. Canny Edge Detection Result

（3）Anchor Point Selection

The bottom container stacking operation requires that the container be hoisted into the area marked by the container corner line. Through observation, it is found that when the bottom container stacking meets the requirements, the container will not block the corners of the container corners. Therefore, in this article, the corner point of the corner line will be the positioning point during the container alignment operation, as shown by the light yellow point in Figure 9.

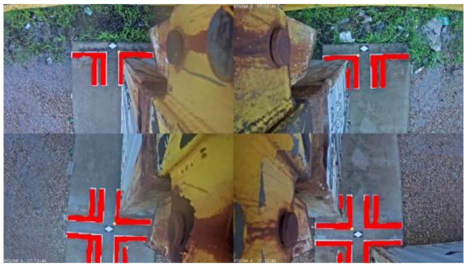


Figure 9. Anchor Points

（4）Extraction of Standard Container Corner Positioning Points

In the process of bottom container stacking alignment operation, the positioning point extraction faces the following problems: 1) The container will inevitably block the corner line of the container, resulting in the disappearance of the corner point, as shown in Figure 10; 2) 2) Although the performance of the neural network is excellent, the perception results are still uncertain due to the existence of interference factors (illumination intensity, electromagnetic interference, etc.) and the "black box" attribute of the neural network, resulting in false detection (as shown in Figure 11). Even if the probability of false detection is very low, in order to improve the reliability of the system, this situation cannot be ignored; 3) There will be multiple container corners appearing in the camera's field of view, and the canny edge detection algorithm will detect the outline of each container corner (as shown in Figure 8). At this time, it is necessary to filter out the desired container corner. Aiming at these problems, this article proposes a standard container angle positioning point extraction algorithm.

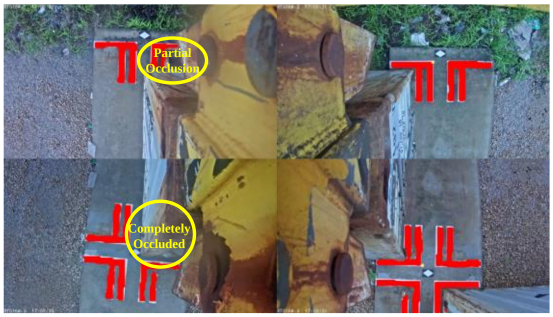


Figure 10. Legend of Container Corners Being Occluded

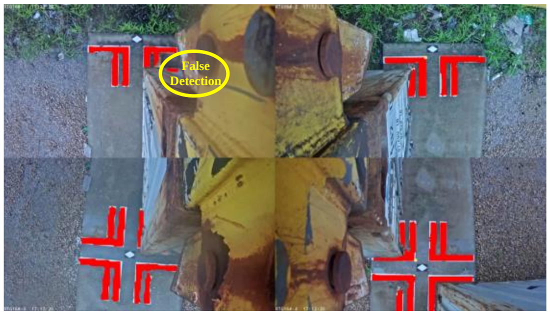


Figure 11. Legend of Container Corner False Detection

1) For questions 1 and 2

Each contour detected by canny is analyzed, and only standard container corners are filtered out, as shown in Figure 12. For the standard container corner line, this article gives three criteria: the aspect ratio of the container corner line (that is, the aspect ratio of the minimum bounding rectangle of the container corner line), the opening direction of the container corner line and the shape of the container corner line (L-type or F-type). Only if these three criteria are met at the same time, it will be judged as a standard container angle.

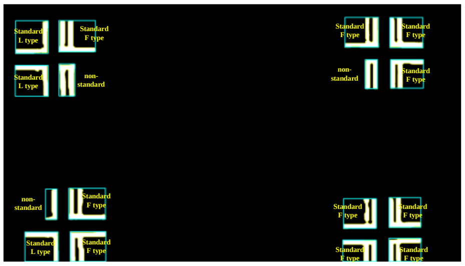
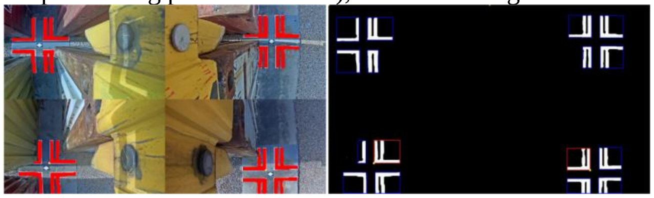


Figure 12. Standard and Non-standard Container Plot Legends

For the top-left, top-right, bottom-left, and bottom-right cameras, it is expected that the opening directions of the container corner lines are bottom-right, bottom-left, top-right, and top-left, respectively. According to the standard container corner line judgment conditions, the container corner line is judged and the positioning point is extracted (Note: Only the standard container corner line can be used for positioning point extraction), as shown in figure 13.



a) original perception result map b) anchor point extraction map

Figure 13. The Extraction Legend of The Standard Container Corner Line Positioning Point (Note: the positioning point is marked with a light yellow point, the container corner line that meets the judgment standard is marked with a red rectangle, otherwise it is marked with a blue rectangle.)

Since only the standard container corners are extracted for positioning points, the problems of occlusion and false detection can be effectively solved. The stricter the standard container angle determination standard is set, the lower the false detection rate, but the higher the missed detection rate. A positioning point extraction algorithm with balanced performance can be obtained by adjusting the threshold value of the judgment standard.

If the camera from one or several perspectives fails to identify the positioning point, the posture of the container relative to the container angle can be inferred based on the perception results of other cameras. For the specific method, see Section 4.2.

2) For question 3

When the spreader lifts the container, the spreader and the container, and the spreader and the camera are rigidly connected, so the relative position of the camera and the container will not change. In other words, the position of the container on the camera frame is fixed. When the container is aligned with the container corner line, it is necessary to obtain the position of the container corner line and the container. The position of the container corner line is the positioning point, and for the position of the container, since the position of the container on the camera screen is fixed, a point can be directly marked on the image as the position of the container. After analysis, this article uses the four corners of the container as the calibration points of the container position, as shown by the yellow points in Figure 14.

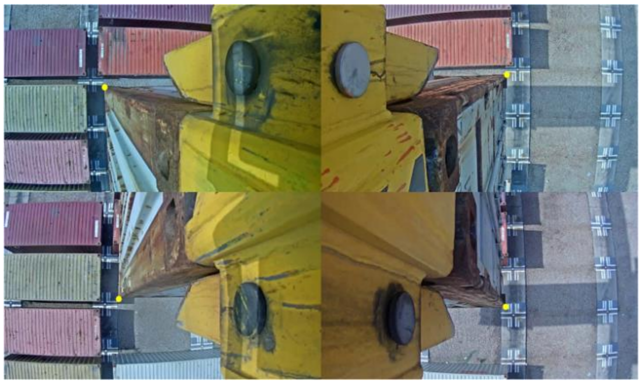


Figure 14. Container Position Calibration Points

Before the bottom container stacking operation, the crane control system will drive the tire crane to move above the corresponding container angle line, as shown in Figure 14. In Figure 14, multiple standard container corners will appear in the same camera screen, as shown by the red container in Figure 15.

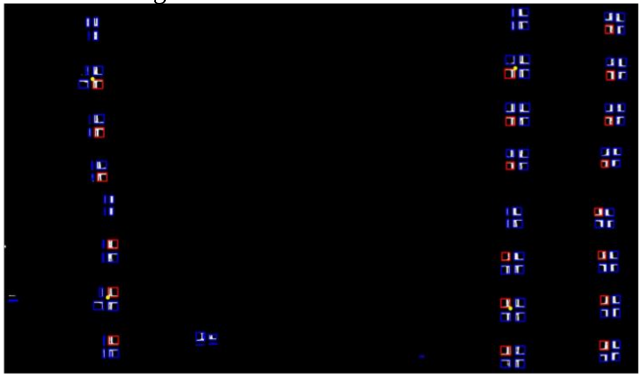


Figure 15. Multiple Standard Container Corner Lines Appear in The Unified Camera Screen (Note: the above picture is a stitched image of 4 cameras)

Each standard container corner line can extract the positioning point. By calculating the distance between each positioning point and the calibration point, the positioning point with the smallest distance from the calibration point is used as the desired positioning point. The final positioning point detection result is shown in Figure 16. The small light yellow point in the figure is the desired positioning point, and the large yellow point is the calibration point.

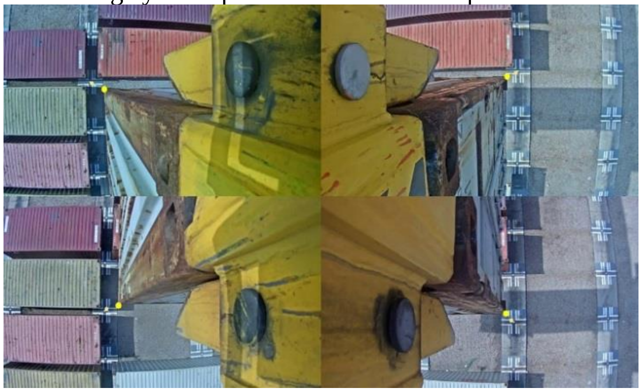


Figure 16. Expected Location Point Detection Result

The target detection and segmentation module proposed in this article adopts the method of combining deep learning and traditional image processing, which greatly improves the accuracy and reliability of the model, and successfully applies vision to the port open production environment.

IV. Motion Decision Module

After obtaining the position of the container corner line and the container in each camera image, it is necessary to carry out motion modeling according to the geometric relationship between them and the mechanical structure, and generate the deviation along the gantry, trolley and rotation direction. Finally, according to the operation process of the RTG, the final logic control signal is output.

A. *Motion Modeling Module*

（1）Analysis of bottom container stacking motion

If the installation height of the cameras and the focal length of the lenses are the same, and the center points of the cmos target surfaces of each camera are aligned horizontally and vertically, and the sides of the cmos target surfaces are parallel, the image stitched by the four cameras can be regarded as an image captured by one camera. In this ideal case, the pixel plane coordinate system can be used directly, as shown in Figure 17. Among them, the origin of the coordinates is the top left point of the image, the x direction is the direction of the cart's movement, and the y direction is the direction of the car's movement.

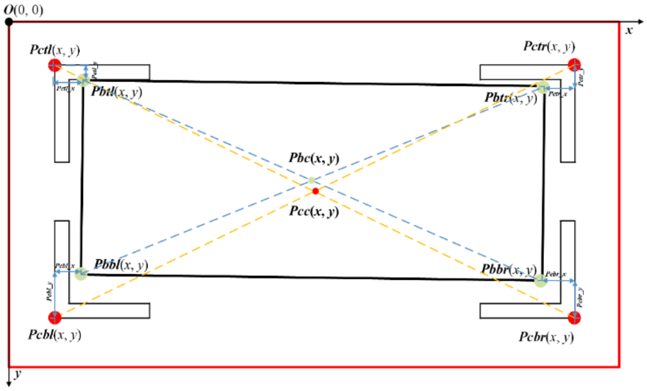


Figure 17. Schematic Diagram of The Ideal Installation Position of The Camera (the four-camera mosaic image is in the red area; the black rectangle in the image is the container; the L-shaped sign is the container corner line)

In Figure 17, ***Pctl***，***Pctr***，***Pcbl***，***Pcbr*** are the container corner line positioning points, and ***Pbtl***，***Pbtr***，***Pbbl***，***Pbbr*** are the container position calibration points. ***Pbc*** is the center point of the container, and ***Pcc*** is the center point of the container corner line positioning area.

During the alignment operation, the calculation formula for the direction deviation *err_x* of the cart and the direction deviation *err_y* of the trolley is as follows:

$$\begin{cases} err\_x = Pbc\_x - Pcc\_x \\ err\_y = Pbc\_y - Pcc\_y \end{cases} \tag{1}$$

The formulas for calculating the coordinates of point Pbc and point Pcc in the x and y directions respectively are as follows:

$$\begin{cases} Pbc\_\mathrm{x} = \frac{Pbtl\_x+Pbtr\_x+Pbbl\_x+Pbbr\_x}{4} \\ Pbc\_y = \frac{Pbtl\_y+Pbtr\_y+Pbbl\_y+Pbbr\_y}{4} \\ Pcc\_x = \frac{Pctl\_x+Pctr\_x+Pcbl\_x+Pcbr\_x}{4} \\ Pcc\_y = \frac{Pctl\_y+Pctr\_y+Pcbl\_y+Pcbr\_y}{4} \end{cases} \quad (2)$$

Substituting formula (2) into formula (1), we get:

$$\begin{cases} err\_x = \frac{(Pbtl\_x-Pctl\_x)+(Pbtr\_x-Pctr\_x)+(Pbbl\_x-Pcbl\_x)+(Pbbr\_x-Pcbr\_x)}{4} \\ err\_y = \frac{(Pbtl\_y-Pctl\_y)+(Pbtr\_y-Pctr\_y)+(Pbbl\_y-Pcbl\_y)+(Pbbr\_y-Pcbr\_y)}{4} \end{cases} \quad (3)$$

In the four cameras, the calculation formula for the displacement of the calibration point and the positioning point in the x and y directions (see Figure 17 for details) is as follows:

$$\begin{cases} Petl\_x = Pbtl\_x - Pctl\_x \\ Petl\_y = Pbtl\_y - Pctl\_y \\ Petr\_x = Pbtr\_x - Pctr\_x \\ Petr\_y = Pbtr\_y - Pctr\_y \\ Pebl\_x = Pbbl\_x - Pcbl\_x \\ Pebl\_y = Pbbl\_y - Pcbl\_y \\ Pebr\_x = Pbbr\_x - Pcbr\_x \\ Pebr\_y = Pbbr\_y - Pcbr\_y \end{cases} \quad (4)$$

Substituting formula (4) into formula (3), we get:

$$\begin{cases} err\_x = \frac{Petl\_x+Petr\_x+Pebl\_x+Pebr\_x}{4} \\ err\_y = \frac{Petl\_y+Petr\_y+Pebl\_y+Pebr\_y}{4} \end{cases} \quad (5)$$

It can be seen from formula (5) that *err_x* and *err_y* are the average displacements of the calibration point and the positioning point in the *x* and *y* directions in the four cameras.

During the alignment operation, the angle between the line segment（***Pbbl***, ***Pbtr***）and the line segment（***Pcbl***, ***Pctr***）can be calculated for the rotation deviation. However, it is found in practical applications that it can be measured by the difference in displacement between the calibration point and the positioning point on the same side in the x or y direction. For example, the difference between the displacements of the top left and top right cameras in the *y* direction is selected as the rotation deviation, and the formula is as follows:

$$err\_\theta = \ Petl\_y - Petr\_y \quad (6)$$

（2）Multi-camera fusion positioning coordinate system

The above is the calculation of the deviation of the camera in the ideal installation state. However, in practice, it is difficult to install the camera in place. Even if the camera is installed in place, the installation position of the camera will be shifted due to vibration with the cycle operation of the tire crane, as shown in Figure 18. In this case, the pixel plane coordinate system is no longer applicable.

In response to this problem, this article proposes a multi-camera fusion positioning coordinate system, with the vertex of the container line (the red point in the figure) as the origin, and the two sides of the container line as the x-axis and y-axis, as shown in Figure 18. The reasons why the coordinate system can be constructed in this way: 1) No matter how the camera is installed, the relative positional relationship between the container and the container corner line in each camera image will not change with the installation angle of the camera; It looks like the container and the corner line of the container have been deformed, but it is not. Therefore, formulas (5) and (6) can still be used for the calculation formulas of the gantry direction deviation err_x, the trolley direction deviation err_y and the rotation deviation err_θ, but they should be calculated in the multi-camera fusion positioning coordinate system.

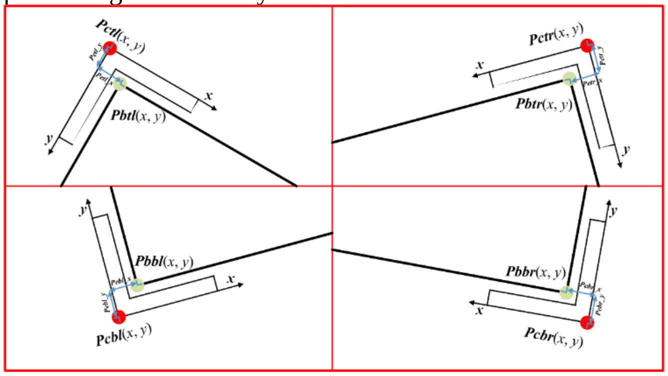


Figure 18. Four-camera Composite Image of Non-ideal Camera Installation Position

B. Logic Control Module

（1）Logic control under occlusion state

During the tire crane alignment operation, the logic control strategy of the spreader's micro-movement is as follows:

1) rotate the spreader until the movement is in place;

2) Move the trolley in the direction until the movement is in place;

3) Finally carry out the direction movement of the gantry until the movement is in place.

According to the logic control strategy and the calculated deviation, the bottom container stacking operation can be carried out. However, during the bottom container stacking operation, the container may block the corner line of the container, so that the positioning point cannot be detected. In this case, the positional relationship between the container and the corner line of the container can be inferred based on the recognition of the remaining cameras. The specific strategy is as follows:

1) Unobstructed. In this state, the calculation formulas of err_x, err_y and err_θ are the same as formulas (5) and (6).

2) Only block one container corner line. When only one container line is occluded and the positioning point cannot be identified, it can be calculated based on the positioning points in the other three cameras. Take the top right container corner line being blocked as an example, as shown in Figure 19.

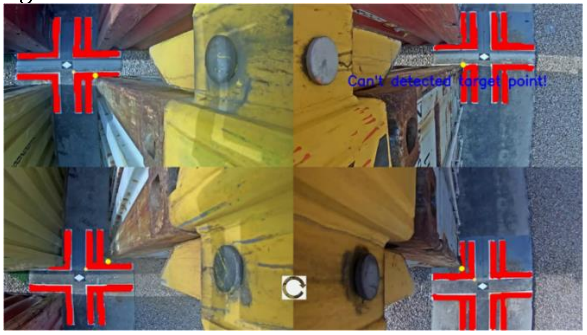


Figure 19. The Top Right Container Corner Line Blocked

The calculation formulas of err_x and err_y are as follows:

$$\begin{cases} err\_x = \frac{Petl\_x+Pebl\_x}{4} + \frac{+Pebr\_x}{2} \\ err\_y = \frac{Petl\_y+Pebl\_y}{4} + \frac{+Pebr\_y}{2} \end{cases} \qquad (7)$$

The calculation formula of *err_θ* is as follows:

$$err\_\theta = \ Pebl\_y - Pebr\_y \qquad (8)$$

The calculation method for the remaining three cases is similar.

3) Only cover two container corner lines

Limited by the geometric relationship between the container and the container corner line, if two container corners are blocked at the same time, the blocked container corners must be on the same side, as shown in Figure 20.

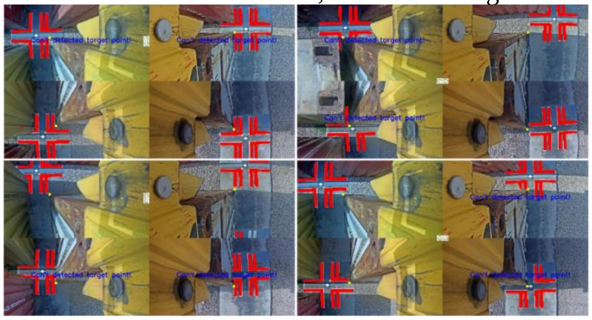


Figure 20. Blocking Two Container Corner Lines at The Same Time

Taking the simultaneous occlusion of the two top container corner lines as an example (that is, the top left of Figure 20), the calculation formula of err_θ is the same as formula (8). The calculation formula of err_x is as follows:

$$err\_x = \frac{Pebl\_x+Pebr\_x}{2} \qquad (9)$$

Since the two positioning points on the same side cannot be identified, the deviation err_y of the trolley direction cannot be accurately calculated, but the movement direction of the trolley direction can be deduced based on the recognition of the four cameras. In this case, set another step You can control the trolley to move in the correct direction. As the motion progresses, the occlusion disappears, after which it can be calculated precisely according to equation (5) or (7).

4) Block three or four container corner lines

Due to the geometric relationship between the container and the corner line, this situation does not occur. If this happens, it can be considered that the foreground segmentation module recognizes an error, and at this time, it will jump from automatic control to remote manual operation.

（2）Control rate design

The dimensions of *err_x*, *err_y* and *err_θ* calculated above are all pixels, which are not the position, speed, etc. required for actual control. How to apply these deviations to the tire crane control system also needs to be designed for the control rate. During the alignment operation, the hoisted cargo weighs dozens of tons, which belongs to the heavy-duty low-speed system. The control logic of the existing manual operation is as follows:

1) The driver judges the movement direction of the spreader at the next moment by observing the relative positional relationship between the current container and the container corner line;

2) Operates the corresponding handle, gives the switch value, and controls the tire crane. Perform alignment work.

3) Repeat steps 1) and 2) until the accuracy requirements are met.

The design of the control rate in this article simulates manual work. For the three control variables of *err_x*, *err_y*, and *err_θ*, a fixed step size is set: *step_x*, *step_y*, *step_θ*, and the control rate formula is as follows:

$$err\_x = \begin{cases} step\_x, err\_x > 0 \\ 0, err\_x = 0 \\ -step\_x, err\_x < 0 \end{cases} \qquad (10)$$

$$err\_y = \begin{cases} step\_y, err\_y > 0 \\ 0, err\_y = 0 \\ -step\_y, err\_y < 0 \end{cases} \qquad (11)$$

$$err\_\theta = \begin{cases} step\_\theta, err\_\theta > 0 \\ 0, err\_y = 0 \\ -step\_\theta, err\_\theta < 0 \end{cases} \qquad (12)$$

C. *Experimentation*

（1）Multi-camera fusion positioning coordinate system

The logic control module is used to make motion decisions for each frame of images during bottom container stacking operations. For the decision-making results, this article adopts the manual verification method. Finally, Finally, the decision accuracy reached 99.9% in 4000 frames. Some experimental results are shown in Figure 21.

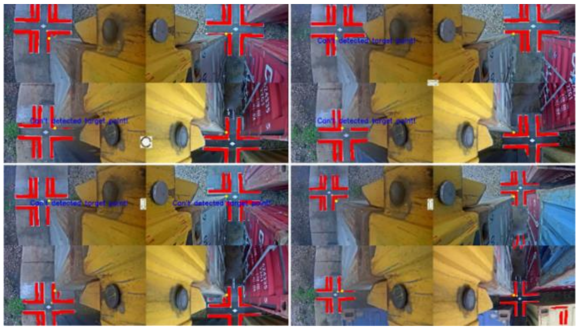
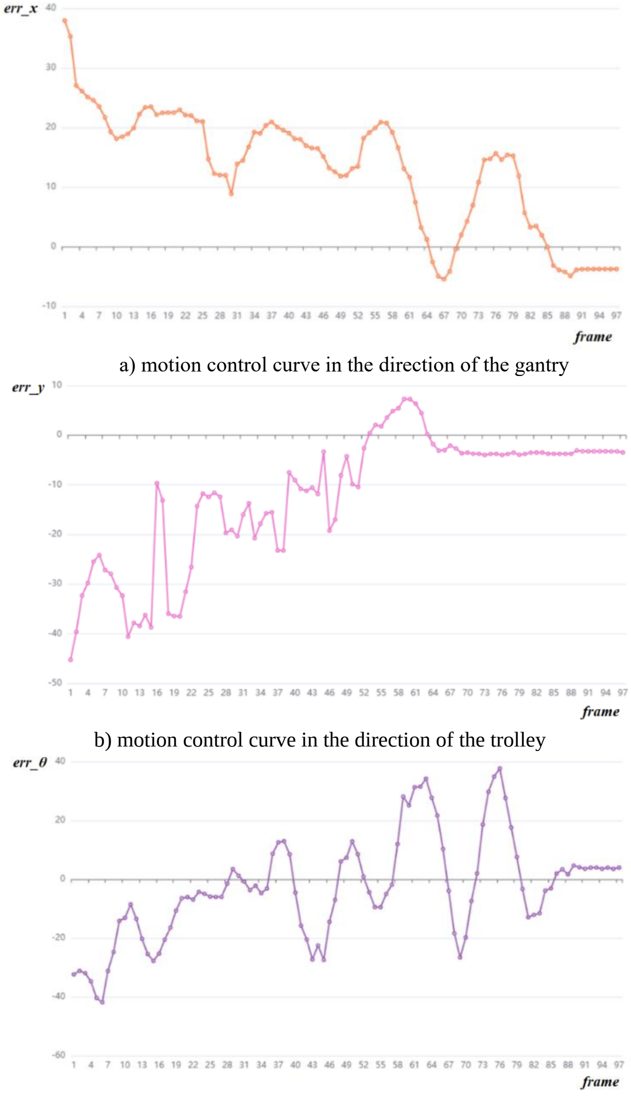

Can't detected target point
Can't detected target point
Can't detected target point
Can't detected target point
Can't detected target point
Can't detected target point

Figure 21. Decision Result Diagram of Logic Control Mode (Note: The arrow in the picture represents the direction of the tire crane's movement at the next moment)

（2）Motion control experiment

According to the designed control rate, the motion control curves of the tire suspension along the gantry, trolley, and rotation directions are shown in a), b), and c) in Figure 22.



a) motion control curve in the direction of the gantry

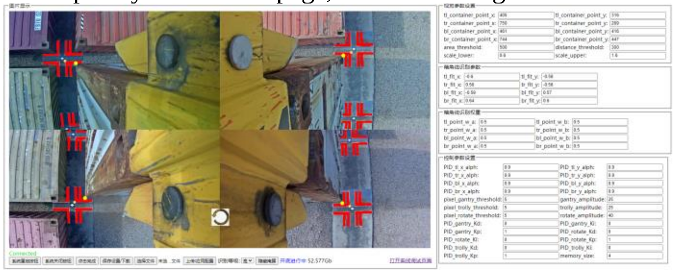
b) motion control curve in the direction of the trolley

c) rotation direction motion control curve

Figure 22. Bottom Container Stacking Motion Control Curve

V. Human Machine Interface

In this project, the Human Machine Interface is developed by HTML webpage, as shown in Figure 23.



tl_container_point_x:	406	tl_container_point_y:	316
tr_container_point_x:	750	tr_container_point_y:	289
bl_container_point_x:	401	bl_container_point_y:	416
br_container_point_x:	744	br_container_point_y:	447
area_threshold:	500	distance_threshold:	300
scale_lower:	0.6	scale_upper:	1.6

tl_fit_x:	-0.6	tl_fit_y:	-0.58
tr_fit_x:	0.58	tr_fit_y:	-0.58
bl_fit_x:	-0.59	bl_fit_y:	0.57
br_fit_x:	0.64	br_fit_y:	0.6

tl_point_w_a:	0.5	tl_point_w_b:	0.5
tr_point_w_a:	0.5	tr_point_w_b:	0.5
bl_point_w_a:	0.5	bl_point_w_b:	0.5
br_point_w_a:	0.5	br_point_w_b:	0.5

PID_tl_x_alph:	0.9	PID_tl_y_alph:	0.9
PID_tr_x_alph:	0.9	PID_tr_y_alph:	0.9
PID_bl_x_alph:	0.9	PID_bl_y_alph:	0.9
PID_br_x_alph:	0.9	PID_br_y_alph:	0.9
pixel_gantry_threshold:	5	gantry_amplitude:	26
pixel_trolly_threshold:	5	trolly_amplitude:	25
pixel_rotate_threshold:	5	rotate_amplitude:	40
PID_gantry_Kd:	0	PID_gantry_Ki:	0
PID_gantry_Kp:	1	PID_rotate_Kd:	0
PID_rotate_Ki:	0	PID_rotate_Kp:	1
PID_trolly_Kd:	0	PID_trolly_Ki:	0
PID_trolly_Kp:	1	memory_size:	4

Connected

52.577Gb

Figure 23. Human Machine Interface

The debugging personnel only need to open the browser and enter the corresponding URL to log in to the human machine interaction interface. The key parameters are provided in the interface to facilitate the debugging personnel to adjust the parameters of different tire cranes, so as to ensure the normal operation of the crane visual automatic alignment system.

VI. Conclusion

Aiming at the problem of automatic alignment control of RTG, a novel visual automatic alignment system is proposed, which realizes the integration of perception and motion decision-making. Firstly, a target detection and segmentation module is designed to perceive and locate the target, which fully combines the powerful perception ability of neural network with the interpretability of traditional image algorithms, greatly improves the accuracy and reliability of the model. Then a motion decision module is designed to convert the perception and positioning results into control signals, which greatly simplifies the camera calibration process and realizes the integration of visual positioning and alignment control. Finally, an html-based human-computer interface is developed for the system, which can be used by the debuggers to adjust the system parameters and monitor the system running status. The experimental results show that the decision-making accuracy of the visual automatic alignment system designed in this article is as high as 99.9%. The practical application results show that the operation efficiency is greatly improved and the failure rate is low.

The visual automatic alignment system for RTG proposed in this article is the first time to apply the deep learning-based visual technology to the industrial control in the port environment, but the design of the control rate is relatively simple. In subsequent research, the design of the control rate can be further explored in the direction of reinforcement learning to further improve the operating efficiency of the system.

References

[1] Zhang Dewen. The status quo of the standardization of container terminal technology and equipment in China[J]. Port Science AND Technology,2021(5):1-6.

[2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems. 2012:1097-1105.

[3] Bhavana D, Kishore Kumar K, Bipin Chandra M, Sai Krishna Bhargav P V, Joy Sanjana D, and Mohan Gopi G. Hand sign recognition using CNN[J]. International Journal of Performability Engineering, 2021, 17(3): 314–321.

[4] Dongcheng Li, W. Eric Wong, Wei Wang, Yao Yao, and Matthew Chau. Detection and mitigation of label-flipping attacks in federated learning systems with KPCA and K-means[C]. 2021 8th International Conference on Dependable Systems and Their Applications (DSA), 2021, 551–559.

[5] Chhabra, Megha, Manoj Kumar Shukla, and Kiran Kumar Ravulakollu. Intelligent optimization of latent fingerprint image segmentation using stacked convolutional autoencoder[J]. International Journal of Performability Engineering, 2021, 17(4): 379–393.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 779-788.

[7] Joseph Redmon, Ali Farhadi. YOLO9000: Better, Faster, Stronger[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 6517-6525.

[8] Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement[C]. CVPR, 2018: 202.113.61.185.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiContainer Detector[C]. European Conference on Computer Vision, 2016:21-37.

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár. Focal Loss for Dense Object Detection[J]. Transactions on Pattern Analysis and Machine Intelligence, 42(2):318-327.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[12] Ross Girshick. Fast R-CNN: Fast Region-based Convolutional Networks for object detection[C]. 2015 IEEE International Conference on Computer Vision (ICCV), 2015: 1440-1448.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017, 39(6): 1137-1149.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[15] Hutchinson S, Hager G D, Corke P I. A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 1996, 12(5): 651-670.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision[C], 2014:740-755.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition[C], CVPR, 2009:248-255.