Development of a Computer Vision based Robot Workstation for Sorting Applications

SURF
Summer Undergraduate Research Fellowship

Abstract

This project aims to develop a robot workstation for sorting application with a computer vision system. The most important advantage with computer vision is that the robot can become versatile rather than (for traditional robotic systems based on offline programming) being able to fulfil just single tasks. The workstation needs to sort out workpieces of different shapes and colours.

The detailed tasks involve completing the basic hardware configuration, and, choosing, optimizing and implementing appropriate algorithms for imaging processing and motion planning. The resulting platform with the proposed working principle will be meaningful for future use in both teaching and research.

School of Advanced Technology，Xi’an Jiaotong - Liverpool University
Surf code:
2021061
Supervisor:
Quan Zhang Junwei Wu(PHD Candidate)

Name:
Muhuan Wu Tiancheng Zhang Seungyeop Kang
Fangrui Guo Tianxin Li Xiaoyi Zhou

Deep learning vision algorithm

1. Mask-RCNN

Theory of mask-rcnn

Mask R-CNN is an instance segmentation algorithm, which can be used for "target detection", "target instance segmentation" and "target key point detection". It has been used to distinguish the different shapes and identify their exact positions. Detailed steps are listed as bellow.

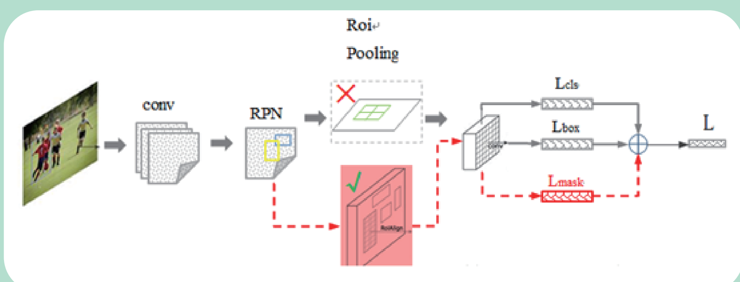


Figure1: Detailed steps of Mask-RCNN

Realization and modification

In this part, we will explain the difficulties we encountered through the developing process, and how we managed to solve them.

Initially, after setting up the environment and directly used the pretrained weights file from COCO to run mask-RCNN model, we started to train the dataset of shapes. However, some problems came out through the training process. Because of the limited dataset, the accuracy is not ideal, only around 60%. Then we modified the resnet50 to resnet101, increasing the depth of the CNN to achieve higher accuracy. However, it turned out that the training time is quite long, and the GPU quality is challenged, too. Therefore, we changed the algorithm of the loss function, decrease the original learning rate (0.001) by one fifth, and break the whole training process when the loss beginning to increase. This modification applied the basic theory early stop method, which means the training was stopped when the error of verification set stops falling but begins to rise, so as to effectively prevent over-fitting.Finally, the model works well in the task.

Results and conclusion

The pixel center point of each shape has been listed in the upper part, and these will be transmitted to control end. Each shape has its unique mask to distinguish themselves between others, which is the meaning of instance segmentation algorithm.

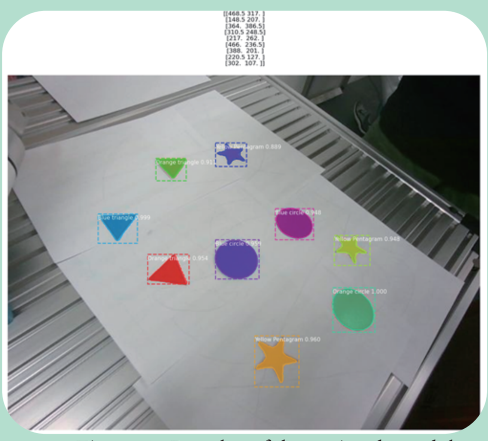


Figure 2: Results of the trained model

2. YOLOv3

Theory of mask-rcnn

YOLO is an object detection algorithm based on one-stage object detection which is widely used in Real-Time object detection. As a result, it is also commonly used in Robot workstations due to its speed and accuracy. In this project, YOLOv3 algorithm is applied due to its increased accuracy and speed than the previous versions.are listed as bellow.

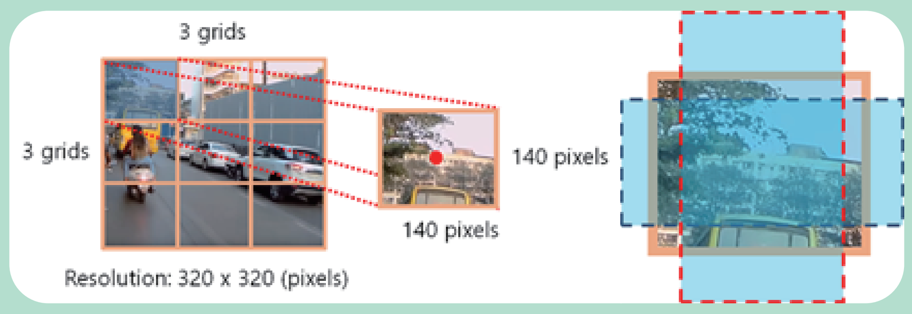
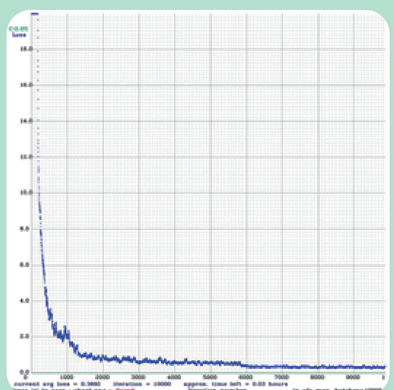


Figure 3: Grid and Anchor Box

To apply YOLO algorithm to this project, the custom dataset is required for the specific tasks. Therefore, about 500 pictures were labelled with high precision to train the required five classes. Then, after 10,000 iterations (5×2000), the loss will be decreased as the figure below.



After obtaining the custom dataset, the YOLOv3 algorithm was modified to operate with a Realsense camera and communicate with the B&R Robot Workstation.

As the figure4 shows, the created bounding boxes can be observed with the labelled names and center points of the objects. Based on this result, the coordinates of the detected points will be printed as the figures to convert the pixel values to the world coordinates in the unit, ‘mm’. According to the test, the task can be completed with high accuracy with minute error around 1 to 2mm.

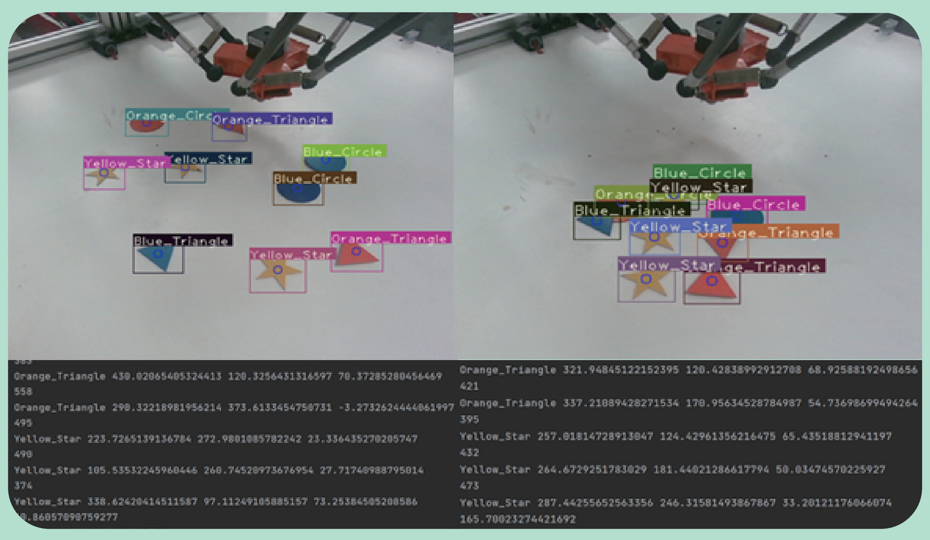


Figure 4: The detection result (Sparse case) The detection result (Dense case)

3. SSD

SSD, named after Single Shot Multibox Detector, is a particularly efficient depth learning vision algorithm. This algorithm is based on Yolo and Faster R-CNN, so it inherits the characteristics of fast and accurate recognition of these two algorithms. As shown in figure 1, in this algorithm, after each input picture is compressed to the size of 300 * 300, it passes through several convolution layers, pooling layers and full connection layers, and finally the position of the object in the picture can be recognized.

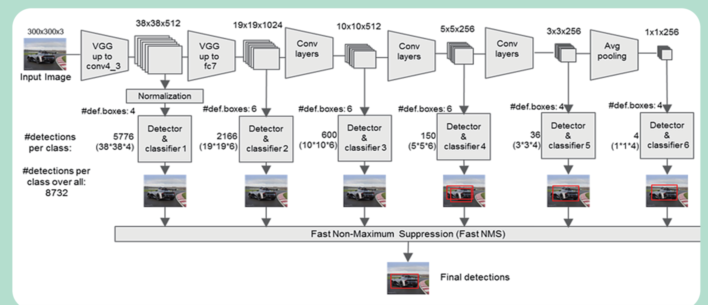


Figure 5: SSD theroy

SSD algorithm has six feature layers in total, and the number of prior frames corresponding to each feature point of each feature layer is 4, 6, 6, 4 and 4 respectively. The significance of a priori box is that the algorithm can easily identify targets of various sizes in each picture. Each feature layer divides the original 300 * 300 pictures into different numbers of fragments, and the a priori box is the number and range of target features extracted from each fragment, as shown in the figure.

The training set in voc2007 format was used in this algorithm, and the transfer learning was carried out through the pre-training model of 20 categories downloaded from the Internet. In the training of the new training set based on 200 + photos, two epochs of training were able to improve the accuracy rate to more than 80%, and two hundred epochs of training finally improved the accuracy rate to more than 98%. Such a high accuracy rate was also beyond expectation.

As shown in the figure below, after adding depth information, SSD algorithm can identify the specific coordinates of all targets in each frame and keep the number of frames above 30 in the CUDA accelerated environment.

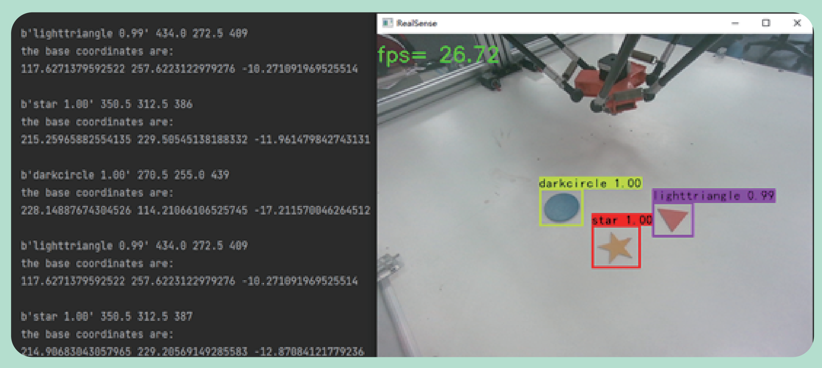


Figure 6: SSD results

Calibration

Calibration is a method that is able to convert the 2D image coordinates detected by the camera into the base world coordinate system of the control system. The formula is as follows:

$$ {}^{Base}_{Image}T = {}^{Base}_{Board}T \, {}^{Board}_{Camera}T \, {}^{Camera}_{Image}T $$

It is easy to know that there are three conversion relationship from the formula, each of which is a transformation matrix transferring a coordinate from one coordinate system to another. The camera used has the capability to detect distance from the object, where the transformation matrix from the image to the camera is able to add the depth information to the pixel coordinate. And by producing with internal matrix, which is a fixed parameter of a camera, all the coordinate points can be represented in the camera coordinate system.

The transformation matrix from the camera to the board, which is also called as external matrix, is calculated by taking one picture of a calibration plate, which is the board. In this surf project the coordinate system of the board is the same with the base one, therefore the last transformation matrix does not need to be considered.

Control

Control Unit Design:

The design of our control cabinet for this project is shown in Figure 7, where the programming logic controller (PLC) module provided by B&R is our main controller that can receive data from our vision algorithm discussed previously.

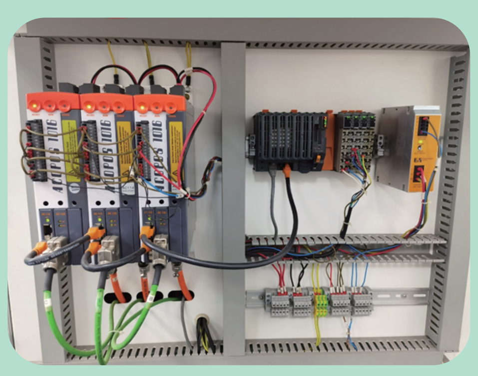


Figure 7: The wiring of the control cabinet

Inverse Kinematics of the delta robot:

The transformation from the robot’s joint coordinates to the cartesian coordinates of its end-effector is also considered. Due to its special mechanical structure, our delta robot is more comprehensive to be controlled than ordinary cartesian robots. During the process of controlling, the controller can only change the rotation of the motors at the upper joints instead of directly moving the end-effector. Therefore, inverse kinematics should be considered so that we can determine how the motors should rotate to ensure the end-effector moving along the predefined trajectory, thus completing the job of controlling the robot. This process is achieved in the software Automation Studio by programming with the structured text (ST) language.

Trajectory Optimization:

After successfully detecting all the objects laid at the workstation, their coordinates can be sent to the PLC through HTTP communication. Given such information, the robot can successively pick all the workpieces and put them to the target positions. To reduce the time of completing this task and improve the efficiency, we tried to find out an optimal sequence of picking each workpiece. This is very similar to the well-known Travelling Salesman Problem (TSP). In terms of solving this kind of problems, genetic algorithms are effective since they can yield relatively optimal solution. Thus, they will be used to optimize this problem. The result of the algorithm is displayed in Figure 8, from which we can see that the distance for the robot to travel decreases dramatically after the iteration process.

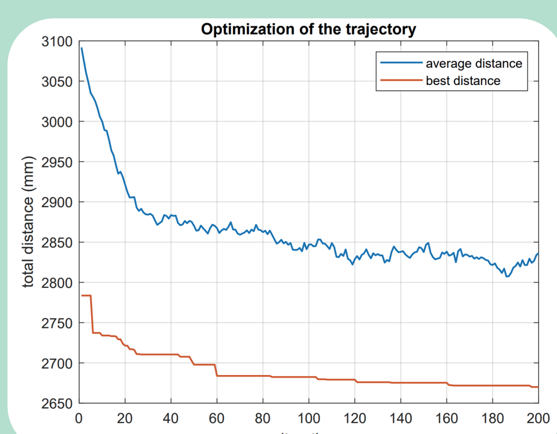


Figure 8: Training results of the genetic algorithm

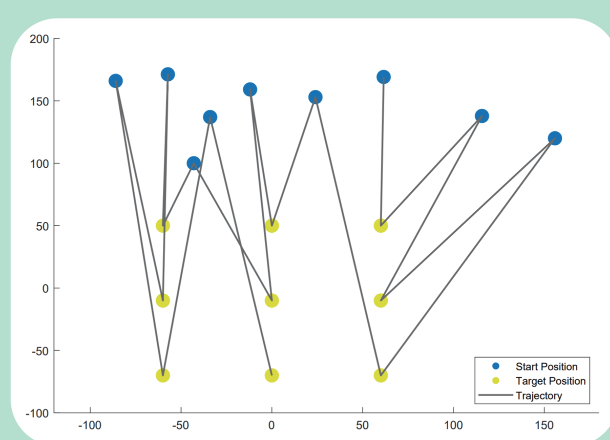


Figure 9: Actual trajectory of the robot

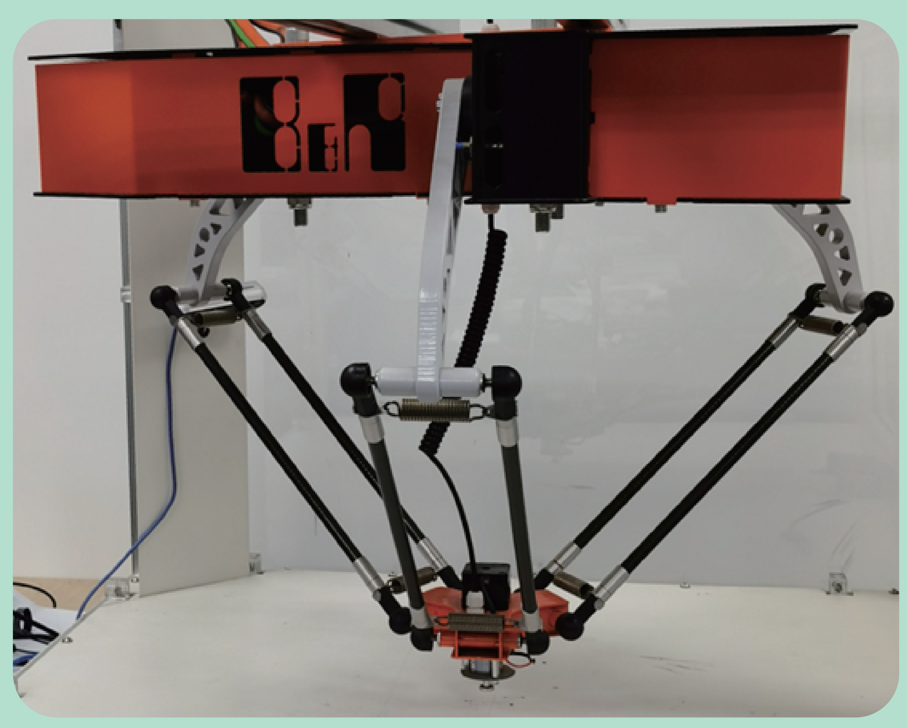


Figure 10: The delta robot

Conclusion

In conclusion, this poster illustrates the process of how the robot workstation can complete sorting tasks based on computer vision. In terms of computer vision, three algorithms, i.e. Mask-RCNN, YOLOv3 and SSD, are applied to compare the results obtained by different algorithms. As the explanation, we obtained three different custom datasets by training each labelled image. Based on these datasets, we are able to detect the customized objects, displaying the centre points of each object. However, the centre points obtained by these algorithms do not include the depth information which is important to convert the 2D-coordinate to 3D-coordinate. Therefore, by adding the depth information, the 3D-coordinate of each object can be obtained. After this process, we achieved the control process of the delta robot. Initially, based on the control units in the control cabinet, we programmed in structured text to implement the inverse kinematics based on the mechanical structure of the robot. Next, through HTTP communication, we successfully sent the coordinate data of each workpiece to the PLC. Then, to improve the efficiency of completing the task, we implemented genetic algorithms to find out an optimal path that the robot will travel in relatively short duration while successfully completing the task of sorting all the workpieces detected.

While completing tasks in this project, we came out some new ideas towards future developments. For the future study, to increase the accuracy and efficiency, we are going to realize the target of identifying, catching and placing the overlapping workpieces.

References

Available from:
https://blog.csdn.net/u010901792/article/details/100044200
https://github.com/matterport/Mask_RCNN
http://blog.csdn.net/l297969586/article/details/79154150
https://pjreddie.com/media/files/papers/YOLOv3.pdf
https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b
https://github.com/ayooshkathuria/pytorch-yolo-v3
https://blog.csdn.net/weixin_44791964/article/details/104981486