Optimized visual recognition algorithm in service robots

Jun W Wu\textsuperscript{1}, Wei Cai\textsuperscript{1}, Shi M Yu\textsuperscript{2}, Zhuo L Xu\textsuperscript{1} and Xue Y He\textsuperscript{1}

Abstract
Vision-based detection methods often require consideration of the robot’s sight. For example, panoramic images cause image distortion, which negatively affects the target recognition and spatial localization. Furthermore, the original you only look once method does not have a reasonable performance for the image recognition in the panoramic images. Consequently, some failures have been reported so far when implementing the visual recognition on the robot. In the present study, it is intended to optimize the conventional you only look once algorithm and propose the modified you only look once algorithm. Comparing the obtained results with the experiment shows that the modified you only look once method can be effectively applied in the graphics processing unit to reach the panoramic recognition speedup to 32 frames rate per second, which meets the real-time requirements in diverse applications. It is found that the accuracy of the object detection when applying the proposed modified you only look once method exceeds 70% in the studied cases.

Keywords
Robot, visual recognition, YOLO algorithm, Panoramic shooting, deep learning

Date received: 5 September 2019; accepted: 19 April 2020

Introduction
Currently, service robots are getting popular in numerous applications at diverse industries. Meanwhile, these devices are continuously miniaturized and become more intelligent. Focusing on the requirements of service robots indicates that although the robot’s functionality is continuously getting stronger, there still exists a serious lack of mature interaction capabilities. Among the functions required for service robots, robotic grasping is the primary step of interacting with the real world. In fact, the robot needs to recognize the position of objects, which require the use of sensors and image recognition approaches.\textsuperscript{1} Involving the object detection methods, a remarkable research model\textsuperscript{2} in the field of computer vision roars across the horizon. The majority of performed researches so far apply sensors, mostly cameras to manually extract features and then conventional machine learning methods are applied to learn the relational mapping\textsuperscript{3} between artificial features and grasping poses from supervised data. Compared to the conventional grasping approaches of industrial robots, which create model libraries based on known objects,\textsuperscript{4} this method can transfer obtained experiences to grasp unknown objects.\textsuperscript{5} However, since the characteristics of the artificial design are restrained by human cognition, there are some limitations in the robot grasping, including the complexity of making more complicated representations.\textsuperscript{6} Therefore, the

\textsuperscript{1}Shanghai Tiger Technology Co, Ltd, Shanghai, People’s Republic of China
\textsuperscript{2}Shanghai High School International Division, Shanghai, People’s Republic of China

Corresponding author:
Jun W Wu, Shanghai Tiger Technology Co., Ltd, Shanghai, People’s Republic of China.
Email: alex.wu@tiger-techs.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
current technology is only valid for a specific object or a certain task. Since the object is distinct, the object detection may be affected by different factors, including shape, size, perspective variation, and external illumination. Accordingly, the extracted features cannot be generalized resulting in poor robustness. Therefore, finding an algorithm for each specific object is highly demanded, which is a great challenge to adapt to new objects.

With recent advancements in the deep learning, which is mainly divided into convolutional neural networks (CNNs) and regional proposal networks (RPNs), object detection has made a huge breakthrough. This is especially more pronounced in the deep learning that promotes the progress of automatic object recognition and localization. To this end, different schemes, including the region-based convolutional neural network (R-CNN), Fast R-CNN, Faster R-CNN, and you only look once (YOLO) detectors have been proposed so far. Among them, the YOLO algorithm is a unified and real-time object detection model, which is based on a single neural network proposed by Redmon and Farhadi. In fact, the YOLO model is a new object detection method that combines the object determination and the object recognition into one by end-to-end detection scheme, which has superior characteristics, including fast detection and high accuracy. Currently, the RCNN algorithms are mainly applied as the mainstream for the deep learning object detection of robotic grasping. However, the corresponding detection speed does not satisfy the real-time requirements. Therefore, the YOLO model has significant potential in robot vision applications. Since the YOLO model learns to predict bounding boxes from supervised data, it is a challenge to generalize prior learning to new objects or objects with unusual aspect ratios or irregular configurations. Since the proposed architecture has multiple downsampling layers from the input image, the YOLO model applies relatively coarse features for predicting bounding boxes. Therefore, the YOLO model is weak in generalization when dealing with irregular aspect ratios in which the same type of object appears in other situations.

On the other hand, the camera sight is another challenge to solve. Normally, a captured image has a limited vision field that shows less image information than a panoramic image. When target objects are located out of sight, the robot cannot detect the object. Consequently, vision-based detection methods often require consideration of the size of the camera sight. Considering this drawback, the panoramic image plays an important role in the object detection and spatial positioning of the robot. Considering limitation in the internal space, design, and the initial cost of the robot, the robot is usually not equipped with a panoramic camera. However, the image stitching technology is quite mature without traces so that it is widely applied to investigate the fields of image processing, computer graphics, and virtual reality. Panoramic image stitching refers to the process of seamlessly stitching a series of ordinary images with overlapping borders to obtain a panoramic image. Moreover, the panoramic field of view is obtained through the panoramic image stitching technology, which can easily detect targets in the surrounding. Panoramic photography is performed by rotating the camera evenly on a horizontal plane. It should be indicated that the achieved effect during the performing of the panoramic photography is similar to the rotation of the eyes or head watching around. Unlike normal images, the length–width ratio of the panoramic picture changes greatly. Therefore, the original YOLO image recognition cannot be applied to this picture size, which causes very poor performance of the object detection. The network structure of the YOLO is flexible, and such problems can be resolved by applying appropriate changes to the YOLO. To solve effects originating from the panoramic images, it is intended to build a deep neural network based on the idea of changing the structure of grid cells. It is expected to improve the detection accuracy by modifying the network structure to achieve a high recognition accuracy in the real time.

Multiobjective recognition algorithms based on the deep learning

To evaluate the performance of the modified YOLO (M-YOLO) algorithm in the robotic grasping, YOLO and Faster R-CNN algorithms are employed to perform the comparison.

Faster R-CNN

Faster R-CNN is a deep learning method for identifying objects in the image. Figure 1 demonstrates the entire framework of the Faster R-CNN.

Figure 1 indicates that the Faster R-CNN consists of four main layers as the following:
1. Convolution layers for feature extraction
As an object detection method for the CNN, Faster R-CNN initially uses a set of basic convolutions, rectified linear unit activation functions, and pooling layers to extract feature maps of the input image. These convolutions are used in the subsequent RPN layers\(^19\) and full connection layers.

2. Region proposal network
RPN is mainly applied to generate regional proposals. First, it generates a pile of anchor boxes. After performing the clipping and filtering processes, Softmax software is employed to determine which anchors belong to the foreground or background. Meanwhile, the bounding box regression modifies the anchor box to form more accurate proposals. More specifically, it is relative to the next box regression of the full connection layer behind it.

3. Region of interest pooling
This layer uses the feature map generated by RPN proposals and the last layer of the visual geometry group (VGG16)\(^20\) to obtain a feature map with the fixed-size proposal, which can be used to identify and locate objects by the fully connected operation.

4. Classifier
The region of interest pooling layer is formed into a fixed-size feature map for full connection operation. Softmax software is applied to classify specific categories.\(^21,22\) Meanwhile, the L1 loss is used for completing the bounding box operation to obtain the precise position of the object.\(^23\)

**YOLO**

Apart from the Faster R-CNN scheme, the YOLO model is another recognition algorithm for the multitarget deep learning.\(^24\) Considering the superior characteristics of the YOLO scheme, it is applied in the present study for robotic grasping. Figure 2 presents the flowchart of the YOLO model. It indicates that the YOLO model consists of three steps as the following:

1. The original image of the camera is captured and then it is divided into a grid with \(S \times S\) resolution.
2. Each grid predicts \(B\) bounding boxes with confidence scores. Each record has five parameters, including \(x, y, w, h,\) and \(p^\ast\) intersection over union (IOU), where \(p\) and the IOU denote probability that a current position is an object and the function that predicts probability of the overlap area. Moreover, \(x\) and \(y\) are the center coordinates, while \(w\) and \(h\) are the width and height of the bounding box, respectively.
3. Then class probability of each grid prediction is calculated as \(C_i = p (\text{class}|\text{object})\).

The YOLO algorithm uses a CNN to implement the multi-object recognition model. In the present study, the pattern analysis, statistical modeling, and computational learning visual object classes (PASCAL VOC)\(^25\) data set are utilized to evaluate the model. The initial convolutional layer of the network extracts features from the image, while the fully connected layer predicts the output class probability and corresponding image coordinates. Figure 3 indicates that to perform the image classification, the network structure reference employs the GoogleNet model with 24 convolution layers and 2 fully connected layers.\(^4\)

During the training session, the sum-squared error is weighted equally in large and small bounding boxes. Furthermore, the error metric indicates that small deviations of large bounding boxes are less relevant than that of small bounding boxes. To solve this problem, the regression is performed based on the square root of the width and height of bonding boxes, instead of the width and height directly. The YOLO algorithm predicts multiple bounding boxes in each grid. During the network training, only one bounding box is required to be responsible for each project. Subsequently, the following loss functions are optimized\(^13\):
\[
\text{Cost} = \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} f_{ij}^\text{obj} \left[ (x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} f_{ij}^\text{obj} \left[ (\sqrt{w_i} - \sqrt{w_i})^2 + (\sqrt{h_i} - \sqrt{h_i})^2 \right]
\]
\[
+ \sum_{i=0}^{s^2} \sum_{j=0}^{B} f_{ij}^\text{obj} \left( C_i - \bar{C}_i \right)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} f_{ij}^\text{noobj} \left( C_i - \bar{C}_i \right)^2 + \sum_{i=0}^{s^2} \sum_{j=0}^{B} f_{ij}^\text{obj} \left( p_i(c) - \bar{p}_i(c) \right)^2
\]

where \( f_{ij}^\text{obj} \) and \( f_{ij}^\text{noobj} \) are the object appearance in the \( i \)th cell and \( j \)th bonding box predictor in the \( i \)th cell, respectively. Moreover, \( \lambda_{\text{coord}} \) denotes the coordinate error weight. In the present study, classification error weights are set to \( \lambda_{\text{coord}} = 5 \) and \( \lambda_{\text{noobj}} = 0.5 \).

When the object is in the cell, the loss function will punish the classification error. Moreover, the loss function penalizes the coordinate error of the bounding box for an active predictor. After the training session, the regression equation can be applied to predict coordinates of the object category and the object in the image coordinate system in real time. Then the three-dimensional position of the object in the camera coordinate system\( ^{16} \) can be obtained by calculating the center depth.

**Differences between the YOLO and Faster R-CNN algorithms**

In this section, it is intended to reveal differences between the YOLO and the Faster R-CNN schemes. First, the R-CNN is a feature extractor. In fact, a selective search is normally applied to extract a certain number (e.g. 2000) of region proposals and then it convolutes the region proposals and extracts features of the \( fc7 \) layer for the classification and regression of the coordinates. In the present study, the support vector machine classification method is utilized instead of the conventional Softmax model. The main contribution of this algorithm is to propose an effective feature utilization method. The majority of researchers in this area use features of the \( fc7 \) layer in engineering practice based on the Fast R-CNN algorithm. Fast R-CNN scheme uses a network to implement all parts except region proposal extraction. Unlike the conventional R-CNN, the classification and loss of coordinate regression in the Fast R-CNN scheme initially update network parameters by the back propagation. Second, not all region proposals but the whole one are put into the extraction when extracting the feature. Then the feature is extracted through the coordinate mapping. It should be indicated that this type of extraction has two advantages, including the fast performance and wide operational range. Since a picture walks through the network only once, the feature is affected by the receptive field so that it can fuse features of the adjacent background to “see” farther. Finally, studies show that it is almost impossible to operate a real-time detection by applying the selective search method.\( ^{27} \) Therefore, it is intended to replace the selective search method with the RPN and share the feature extraction layer\( ^{28} \) with the Fast R-CNN classification and regression network to reduce the computational expense. Experimental results also show that applying the Fast R-CNN improves the speed and accuracy of the prediction. It is found that the RPN is an essence of Faster R-CNN. Moreover, it is the main reason for the high accuracy and low speed of the R-CNN algorithm compared to that of the YOLO scheme.

On the other hand, one of YOLO’s contributions is to convert detection problems into regression problems. In fact, the former Fast R-CNN is divided into two steps, including extracting the region proposal and classifying. The former step judges whether the anchor is the foreground or the background, while the main purpose of the
later step it to see what the foreground is. The YOLO scheme generates coordinates and probabilities of each type directly through the regression.

YOLO is characterized by its fast speed, regression mechanism, and the extraction process of the regional proposal. Unlike other algorithms, YOLO roughly divides the images into $\frac{7}{C2}$ grids to analogize the Fast R-CNN. Each location may belong to two objects by default, and 98 region proposals are extracted. In fact, Fast R-CNN is a sliding window mechanism, and each feature map is a map. Moreover, 9 anchors are returned, totaling about 20 K anchors, and 300 regional proposals are eventually obtained through nonmaximum suppression. There is a huge difference in candidate boxes between two algorithms, so reasonable accuracy of the Faster R-CNN can be justified. Since each location should be refined, the efficiency further decreases. Therefore, the recognition speed of the R-CNN cannot meet the real-time requirements. In addition, YOLO streamlines the network, has slightly less computational expense than VGG, and may result in some speedup. However, the importance of computational expense is less than the other two points mentioned above.

In the following section, the performances of each algorithm will be compared with experimental data.

**M-YOLO algorithm based on the integrated convolution network**

In the present study, robot iKudo is employed to grasp the objects from a complex background containing chairs, tables, and other objects. This session will focus on designing the M-YOLO algorithm.

Figures 4 and 5 indicate that the iKudo has a manipulator and a camera for grasping objects and taking images. General steps of the object grasping are presented in the flowchart. A mission is initially assigned to a robot as a command to grasp an object, for example, a bottle. Once the robot receives the task, it starts to look for an object. There might be some similar objects in the scenario, while the robot should pick a particular object. Therefore, the robot starts to rotate to scan the surroundings through a depth camera and generate a panoramic image. The panoramic cylindrically expanded image is obtained by projecting the annular panoramic image onto a cylindrical surface.

![Figure 4. Object recognition flowchart of the M-YOLO method. M-YOLO: modified you only look once.](image)

![Figure 5. Bottle grasping.](image)
with a specified radius from the mirror. A two-dimensional rectangular cylindrical panorama is obtained by cutting and spreading the cylindrical surface along the radial direction. Figure 6 illustrates the obtained panoramic image.

Based on the obtained panoramic image, the object detection is triggered and the robot is asked to process the image with object detection algorithms to detect the particular object. As a result, the robot finds the object location and starts the navigation to the target object by using the simultaneous localization and mapping technology. Finally, once the iKudo robot arrives at the position with an acceptable standard for the object grasping, the manipulator starts to grasp the detected object.

The YOLO system divides the image into grid cells, where each grid allows to predict two bounding boxes so that there are 2 bounding boxes in total. These generated proposals cover the entire area of the image. If the target center falls in a grid cell, the corresponding cell will be in charge of detecting the object. In this case, the two bounding boxes of such grid cell are expressed as $p(\text{object}) = 1$. In each iteration, the network predicts the location and confidence of 2 bounding boxes and 20 class probabilities of grid cells excluding the background classification. The class probabilities of bounding boxes are predicted by class probabilities of grid cells corresponding to the bounding box center. In the fully connected layer, it requires a fixed-size vector as an input. In other words, this layer requires original images with a fixed size. To this end, the input size of theYOLO is set to $448 \times 448$. Since panoramic images contain many objects to be detected, applying the conventional grid generation method on panoramic images allows the single grid to predict two objects which can easily cause detection failures. This may be attributed to the grid numbers, because the number of target objects in a single grid increases as the grid number decreases.

However, when the panoramic picture of the robot is processed by YOLO, the aspect ratio of the object in the captured image is not an accurate reflection of its ground truth and the features that look high and thin are formed. Therefore, YOLO is less adaptive to the new samples. To solve this problem, the number of forecasting frames in the longitudinal direction is changed, such as in the scene where the height of the object is compressed in the actual image.

M-YOLO, which is proposed in the present study, reduces the number of objects detected in a single grid by increasing the number of transverse grids. In other words, it changes from $S \times S$ to $S \times nS$ where the number $n$ is an adaptive number (see Figure 7) and changes the network structure by adding a feature layer at the end of YOLO’s original network structure. It is worth noting that in the present study, parameters $n$ and $S$ are set to 2 and 7, respectively. The new network structure consists of 19 convolution layers and 5 max pooling layers. Figure 8 shows that these layers generate an improved M-YOLO network based on the YOLO network to meet the requirements of the robot-based panoramic image detection. The main purpose for establishing the M-YOLO algorithm is to provide a more generalized model when the comparison is made with conventional models. Moreover, the established algorithm should allow the model to flexibly detect images with large aspect ratios such as panoramic images. Ideally, this means that until the input image size is regular, the M-YOLO and YOLO algorithms have the same performance. However, in processing images with large aspect ratios such as panoramic images, the M-YOLO algorithm supposed to show a better performance and higher processing speed compared with the YOLO algorithm.

To train the M-YOLO network, a python crawler is applied to generate a training set for the experiment. Then a standard VOC data set is constructed to obtain pretraining parameters after 100 cycles (one epoch). It should be indicated that the input image size of the network is randomly changed every 10 cycles.

**Build a VOC data set**

According to the architecture of the VOC data set, the present study constructs a new data set by using OpenCV to initially read all of the images under the folder. Then, the VOC format is renamed and changed. Figure 9 shows that after preparing the data, it is required to place the image file corresponding to the structure of the VOC data set.

**Labeling the object area**

The object area in the original image is marked with the labeling software for training. The fundamental method is to frame the object area and double-click the category. Then, the complete image is marked and saved. The category selects 20 common types of objects, such as mineral water bottles, parcels, boxes, and chairs. The annotation information is saved in the format of the PASCAL VOC data set, which contains the category of the target and the outsourced border. Then, the actual data of the target are divided by the width and height of the image so that the data are in the range of $0–1$. The data can be read faster during the training and the images of different sizes can be trained. The format is five parameters for a set of data, including index (sequence of category), $x$ (coordinate of the target center), $y$ (coordinate of the target center), $w$ (width of the target), and $h$ (height of the target).
Equation 3 indicates that $x_{\text{max}}$ and $y_{\text{max}}$ are the coordinates of the lower right corner of the frame, while $x_{\text{min}}$ and $y_{\text{min}}$ are the coordinates of the upper left corner of the frame. Moreover, width and height are the image width and height.

**Training**

The configuration of the computing environment that this article utilizes during the training session is configured with Windows 10 system, Intel Core i7-8700 central processing unit (CPU), NVIDIA GeForce GTX1080, graphics processing unit (GPU), 16 GB RAM, with Cuda 8.0, and Python 3.5 installed. Moreover, the corresponding libraries such as Tensorflow v1.1, Numpy, OpenCV3, and CPython extension are used. The robot vision system is connected to the computer equipped with a 1394 capture card. The YOLO parameters are set with a momentum of 0.9 and a weight decay of 0.0005 to prevent overfitting. Table 1 presents the parameters for training.

**Experimental test and the result analysis**

In the experiment, a testing data set consisting of 1000 images is built. It should be indicated that the resolution of each image is $300 \times 1200$. These images consist of 200 images for each class, including occlusion, no occlusion, normal illumination, and weak illumination. The control experiments are set up and tested under YOLO, M-YOLO, and the mainstream detection method of Faster R-CNN. Among them, YOLO and M-YOLO used the Darknet framework, while Faster R-CNN used the Tensorflow framework. Moreover, during the experiments, the bottles and chairs are selected as statistical objects since both of them have obvious motion characteristics.

There are three measurements for the object detection algorithm, including rapidity, accuracy, and robustness. It should be indicated that the detection frame rate per second (FPS), the accuracy rate, and the average overlap rate are considered as comparison parameters.

**Processing speed**

Three detectors YOLO, M-YOLO, and Faster R-CNN are tested with both CPU and GPU. Moreover, the detection frame rate is calculated according to the average time consumption of the test pictures. Liu et al.\textsuperscript{22} investigated the single-shot detector (SSD) and compared the model processing speed (FPS) of SSD300 and SSD512. They showed...
that data with different input sizes adversely affect the processing speed of the model. Therefore, input images with the same input size (448 × 448) are utilized for all models. However, since the YOLO algorithm has to process more images than the M-YOLO algorithm does, the total processing time of YOLO and M-YOLO algorithms are measured to perform the evaluation. Table 2 presents the obtained statistical results in this regard. It is observed that all three detectors do not meet the real-time performances under CPU. However, YOLO and M-YOLO detection speeds exceed 30 FPS under GPU, which is fully satisfied in real time. Moreover, the current detection method of Faster R-CNN is less than 5 FPS. As a result, YOLO and M-YOLO algorithms have a more reasonable performance in comparison with that of the Faster R-CNN scheme.

In the present study, the M-YOLO method is performed on a webcam to evaluate the corresponding real-time performance for fetching images from the camera and display the detected objects.

Table 2 indicates that performing the object detection process on each panoramic image by the M-YOLO takes about 1 s on CPU and 35 ms on GPU. Moreover, when the standard YOLO algorithm is applied to scan the real-time rotation of the robot, several frames should be processed so that longer processing time is required in comparison with the M-YOLO algorithm. In fact, the M-YOLO algorithm only processes a single frame (panoramic image). Table 2 indicates that the processing speed of the standard YOLO algorithm on the CPU is 0.87 FPS, while that of the M-YOLO algorithm is 0.81 FPS. In other words, processing one frame through the YOLO and M-YOLO algorithms takes 1.15 s and 1.23 s, respectively. Therefore, it is concluded that the performance of the YOLO and M-YOLO algorithms in this case study is similar. However, for the object detection in the rotating case, since the YOLO algorithm has to detect several frames during the rotation, the processing time is about 11.49 s, while that of the M-YOLO algorithm is 1.23 s. This remarkable difference in the processing time can be interpreted as the following: In the studied case, the scanning and merging processes in the standard YOLO algorithm take 10 frames with an overlap of 50%, while only a single frame (panoramic image) is processed in the M-YOLO algorithm. A similar result is obtained when the experiment is conducted on the GPU. As a result, it is concluded that the M-YOLO algorithm has a high efficiency rate and reasonable generalization ability for object detection in the rotating case studies.

Accuracy

The accuracy of the object detection is reflected by the error rate. The lower the error rate, the more reliable and accurate the detection model. Among the obtained results, false positive (FP) indicates that the object is classified incorrectly, while a false negative (FN) indicates that the object cannot be located. Images are tested by using YOLO, M-YOLO, and Faster R-CNN. Table 3 presents respective error rates. It is concluded that:

1. The error rate is about 40% by using YOLO for the detection process. Since the object is in a “short and thin” state, the conventional detector cannot be adapted. However, by using the M-YOLO detector, the error rate is reduced to 30% by increasing the number of longitudinal prediction frames to fit the “short and thin” object. Therefore, the performance of the M-YOLO detector is superior to that of the YOLO detector.

2. Comparing the M-YOLO detector with the Faster R-CNN detector, the Faster R-CNN error rate is less than 30%, which is the best detector in terms of accuracy. The error rate of the M-YOLO detector is about 3% higher than that of the Faster R-CNN detector. However, considering the significant advantages of the M-YOLO detection speed, it performs more practically in terms of robot vision detection.

With these comparisons shown above in section A and section B, it could be concluded that the M-YOLO method is a fast, accurate object detector, which is ideal for robot vision applications.

| Model | YOLO | M-YOLO | Faster R-CNN |
|-------|------|--------|--------------|
| Rate  | Chair | Bottle | Chair | Bottle | Chair | Bottle |
| FP    | 0.29  | 0.27   | 0.12 | 0.11 | 0.13 | 0.15 |
| FN    | 0.17  | 0.15 | 0.21 | 0.24 | 0.16 | 0.15 |
| Total | 0.46  | 0.42 | 0.33 | 0.35 | 0.29 | 0.30 |

YOLO: you only look once; M-YOLO: modified you only look once; R-CNN: region-based convolutional neural network; CPU: central processing unit; GPU: graphics processing unit; FPS: frames rate per second.
The bottle is selected as the object to be detected, and other objects such as the chair, box, and table are interference objects. The accuracy and average overlap rate are used as quantitative statistical indicators. The overlap ratio refers to the ratio of the overlap between the detected region and the real region. The higher the value, the more accurate the region of the detection result.

The test results (Figure 10) show that M-YOLO obtains a certain degree of missed detection when the object is occluded. The prediction accuracy and average overlap rate are 6% and 7% lower than the normal environment, respectively. However, the average overlap rate is higher than 65%. When the lack of illumination leads to a certain degree of convergence between the object and the background, the detection accuracy decreases by 5%, and the average overlap rate decreases by 3%. It is concluded that in the occlusion and underlight environment, the M-YOLO method has lower performance. However, it still maintains a higher accuracy and average overlap rate when the comparison is made with the Faster R-CNN. Table 4 presents the robustness of the M-YOLO method.

### Table 4. Robustness quantitative test.

| Case          | Normal | Overlap | Weak light |
|---------------|--------|---------|------------|
| Accurate rate | 0.74   | 0.65    | 0.69       |
| Average Overlap rate | 0.78   | 0.71    | 0.75       |

**Robustness**

The bottle is selected as the object to be detected, and other objects such as the chair, box, and table are interference objects. The accuracy and average overlap rate are used as quantitative statistical indicators. The overlap ratio refers to the ratio of the overlap between the detected region and the real region. The higher the value, the more accurate the region of the detection result.

The test results (Figure 10) show that M-YOLO obtains a certain degree of missed detection when the object is occluded. The prediction accuracy and average overlap rate are 6% and 7% lower than the normal environment, respectively. However, the average overlap rate is higher than 65%. When the lack of illumination leads to a certain degree of convergence between the object and the background, the detection accuracy decreases by 5%, and the average overlap rate decreases by 3%. It is concluded that in the occlusion and underlight environment, the M-YOLO method has lower performance. However, it still maintains a higher accuracy and average overlap rate when the comparison is made with the Faster R-CNN. Table 4 presents the robustness of the M-YOLO method.

**Conclusion**

The original YOLO image recognition does not perform well on the panoramic images, which cause some failures when implementing the visual recognition on the robot. The present study proposes a real-time object detection
method based on the improved YOLO algorithm that is named as the M-YOLO method. The experimental results demonstrate that the M-YOLO method ran with the GPU can detect the panoramic shooting up to 32 FPS, which exceeds the real-time requirement and also shows a good generalization ability for processing the regular and panoramic images. Moreover, it maintains the object recognition accuracy rate of over 70%. However, there are still some problems. For example, the current M-YOLO model only detects a small number of objects and does not obtain a reasonable result in object-intensive scenarios. In the future studies, increasing and training the number of detecting objects should be considered. Moreover, the general ability of the M-YOLO model should be developed to make it perform well in object-intensive scenarios.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD
Jun W Wu https://orcid.org/0000-0002-8139-6304

References
1. He B, Wang S, and Liu YJ. Underactuated robotics: a review. Int J Adv Robot Syst 2019; 16(4): 1–29.
2. Ferrari V, Fevrier L, Jurie F, et al. Groups of adjacent contour segments for object detection. In: IEEE transactions on pattern analysis & machine intelligence, 2007.
3. Shotton J. Textureboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Proceeding of the 9th European conference on computer vision, Berlin: Springer, 2006.
4. Redmon J and Farhadi A. YOLO9000: better, faster, stronger. In: IEEE conference on computer vision and pattern recognition (CVPR), 2017: pp. 6517–6525.
5. Miller A, Knopp S, Christensen HI, et al. Automatic grasp planning using shape primitives. In: Proceeding of IEEE international conference on robotics & automation. Taipei, 14–19 September 2003.
6. Goldfeder C, Allen PK, Lackner C, et al. Grasp planning via decomposition trees. In: IEEE international conference on robotics & automation, Roma, 10–14 April 2007.
7. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, 2016: pp. 770–778.
8. Balasubramanian R, Xu L, Brook P D, et al. Physical human interactive guidance: identifying grasping principles from human-planned grasps. IEEE Trans Robot 2012; 28(4): 899–910.
9. Lecun Y, Bengio Y, and Hinton G. Deep learning. Nature 2015; 521(7553): 436.
10. Blaschko MB and Lampert CH. Learning to localize objects with structured output regression. In: Computer vision ECCV 2008, Berlin: Springer, 2008, pp. 2–15.
11. Girshick R. Fast R-CNN. In: IEEE international conference on computer vision, Santiago, 7–13 December 2015.
12. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Tran Pat Anal Mach Intellig 2017; 39(6): 1137–1149.
13. Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection. In: IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
14. Zhou Y and Tuzel O. VoxelNet: end-to-end learning for point cloud based 3D object detection. In: IEEE conference on computer vision and pattern recognition, 2018, pp. 4490–4499.
15. Redmon J and Farhadi A. YOLOv3: an incremental improvement. arXiv 2018.
16. Bourdev L and Malik J. Poselets: body part detectors trained using 3D human pose annotations. In: International conference on computer vision (ICCV), Kyoto, 29 September–2 October 2009.
17. He B, Liu YJ, Zeng LB, et al. Product carbon footprint across sustainable supply chain. J Clean Prod 2019; 241: 118320.
18. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: IEEE computer society conference computer vision and pattern recognition (CVPR), San Diego, 20–25 June 2005; 1: 886–893.
19. He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European conference on computer vision (ECCV), Berlin: Springer, 2014.
20. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations (ICLR), 2015.
21. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE conference on computer vision and pattern recognition (CVPR), 2014, pp. 580–587.
22. Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multi-box detector. Germany: Springer Verlag, 2016, pp. 21–37.
23. Gould S, Gao T, and Koller D. Region-based segmentation and object detection. Adv Neu Inf Pro Syst 2009; 4: 655–663.
24. Dean T, Ruzon M, Segal M, et al. Fast, accurate detection of 100,000 object classes on a single machine. In: 2013 IEEE conference computer vision and pattern recognition (CVPR), Portland, 23–28 June 2013, pp. 1814–1821.
25. Everingham M, Eslami SMA, Van Gool L, et al. The PAS-CAL visual object classes challenge: a retrospective. Int J Comp Vis 2015; 111(1): 98–136.
26. Han J, Liao Y, Zhang J, et al. Target fusion detection of LiDAR and camera based on the improved yolo algorithm. Mathematics 2018; 6(10): 213.
27. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: *IEEE Conference*, 2015.

28. Zhang X, Yang W, Tang X, et al. A fast learning method for accurate and robust lane detection using two-stage feature extraction with YOLO v3. *Sensors* 2018; 18(12): 4308.

29. Girshick R. Fast R-CNN. In: *IEEE international conference computer vision*, 2015, pp. 1440–1448.

30. Al-masni MA, Al-antari MA, Park JM, et al. Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Comp Meth Prog Bio* 2018; 157: 85–94.

31. Redmon J and Angelova A. Real-time grasp detection using convolutional neural networks. In: *IEEE international conference on robotics and automation*, Seattle, 26–30 May 2015, pp. 26–30.

32. Sihua H, Xiaofang S, Shaoqing Y, et al. Analysis for the cylinder image quality of hyperbolic-catadioptric panorama image system. *Las Inf* 2012; 42 (2): 187–191.

33. Papageorgiou CP, Oren M, and Poggio T. A general framework for object detection. In: *IEEE Sixth international conference oncomputer vision*, 1998, Bombay, 7–7 January 1998, pp. 555–562.

34. Shinde S, Kothari A, and Gupta V. YOLO based human action recognition and localization. *Pro Comput Sci* 2018; 133: 831–838.

35. Ren S, He K, Girshick RB, et al. Object detection networks on convolutional feature maps. *Tran Patt Analy Mach Intellig IEEE* 2017; 39(7): 1476–1481.

36. Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. *Int J Compu Vis* 2015; 115: 211–252.

37. Shen Z, Liu Z, Li J, et al. DSOD: learning deeply supervised object detectors from scratch. In: *IEEE international conference on computer vision*, 2017, pp. 1919–1927.

38. Felzenszwalb PF, Girshick RB, McAllester D, et al. Object detection with discriminatively trained part based models. *IEEE Tran Pat Anal Mach Intellig* 2010; 32(9): 1627–1645.

39. Donahue J, Jia Y, Vinyals O, et al. Decaf: a deep convolutional activation feature for generic visual recognition. In: *IEEE conference on computer vision and pattern recognition* 2013.