Applied Intelligence
https://doi.org/10.1007/s10489-022-03888-4

Deep learning based 3D target detection for indoor scenes

Ying Liu[1] · Du Jiang[1] · Chao Xu[1] · Ying Sun[1,2,3] · Guozhang Jiang[3] · Bo Tao[1,2,3] · Xiliang Tong[1,2] · Manman Xu[1,2] · Gongfa Li[1,2,3] · Juntong Yun[1]

Accepted: 11 June 2022

Abstract
3D target detection is a research hotspot in recent years. In the field of autonomous driving, 3D target detection is mainly targeted at outdoor scenes that the camera height is constant. In a few indoor scenes, 3D target detection is mostly at the category level. However, it is difficult to generate instance-level 3D target detection datasets. In complex indoor scenes, instance-level 3D target detection is used as the research object in this paper. The indoor 3D target detection dataset is constructed by Aruco marker. A pixel-by-pixel key point voting network for joint semantic segmentation of RGB images is established, and a new key point assumption strategy is proposed. Combined with depth images, the key point detection is extended to three dimensions, and the bit pose is optimized by the ICP algorithm. The evaluation metrics and visualization of the model are analyzed and compared. It is tested and visualized under validation set, truncated validation set and unlabeled. The generalization of the method in this paper is proved, and 3D target detection in indoor scene based on RGB image and RGB-D image is achieved.

Keywords RGB-D image · Key point detection · Semantic segmentation · Pose estimation · Indoor scene

1 Introduction

Vision has the most abundant external information. This information is passed to a computer or robot, allowing them to perceive the environment and make decisions instead of humans, which is a hot topic of current research [1, 2]. In recent years, the computer's understanding of 2D information in images has made considerable progress. From object detection, recognition, semantic segmentation to instance segmentation, more and more dominant algorithms have been proposed. Computer vision's understanding of 2D scenes has moved from theory to applications, such as industrial positioning and defect detection, community security, and smart phones. Because of these applications, our lives are convenient and work efficiency is improved [61–64]. However, 2D scene understanding algorithms cannot perceive and understand 3D spatial information, which greatly limits the application of 2D target detection. The target class predicted by 2D target detection and its position can mostly only provide a decisive reference for the computer or robot [55, 67, 68]. It does not provide specific guidance for the computer or robot's next action. In scenarios such as robot grasping and augmented reality, 2D target detection seems helpless.

With the maturity of the 2D target detection algorithm and its limitations, the 3D target detection has begun to enter people's field of vision, and it can well make up for this defect. It can predict the pose of the target in space, so as to better assist the robot in more complex tasks, and provide the robot with accurate real pose information of the target [3, 4]. 3D target detection is to predict the category and pose of the target through RGB-D image or point cloud data and display it in the form of 3D bounding box [5–9]. 3D target detection can

✉ Du Jiang
jiangdu@wust.edu.cn

✉ Bo Tao
taoboq@wust.edu.cn

✉ Gongfa Li
ligongfa@wust.edu.cn

[1] Key Laboratory of Metallurgical Equipment and Control Technology of Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China

[2] Research Center for Biomimetic Robot and Intelligent Measurement and Control, Wuhan University of Science and Technology, Wuhan 430081, China

[3] Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering, Wuhan University of Science and Technology, Wuhan 430081, China

Published online: 16 August 2022

obtain accurate and rich information of the target in space. The 3D target detection can not only used in many applicable scenes of 2D target detection, but also can guide the robot to grasp. Accurate obstacle information for autonomous driving can be provided, and the problem of tracking or occlusion of virtual and real in augmented reality is solved [10].

Compared to outdoor scenes, indoor scenes have smaller views, more compact targets and greater differences in lighting. And due to the random nature of the camera height, it will result in a more complex and varied pose. At the same time, 3D target detection has challenges such as a wide range of applications, more difficult detection and more complex computation. Meanwhile, it consumes a large amount of computational resources, improving the efficiency of the algorithm is quite necessary. With the wide application of service robots, indoor scene understanding has attracted the attention of many researchers, which is closely related to indoor scene target detection. Based on this, aiming at the pixel-by-pixel key point voting of joint semantic segmentation, this paper contributes to reducing unnecessary feature points, so as to cut down the consumption of equipment calculation.

The key contributions of this work are:

(i) A 3D target detection instance level dataset was produced.
(ii) A new key point assumption strategy is proposed based on the basis of the PvNet model. Meanwhile, a fusion method of pixel-by-pixel key point voting and depth image is applied to improve the performance of the model.
(iii) The performance of the proposed algorithm has been analyzed and experimentally verified.

The rest of this paper is organized as follows: Section 2 discusses related work, followed by the explanation of the data set and methods of this article in Section 3. Section 4 is the theory and calculations in the experiment. Section 5 shows the experimental results and the discussion of the results, the conclusion is in Section 6.

2 Related work

3D target detection has always been a research hot spot in recent years. According to the method, it can be divided into: feature descriptor-based, template-based matching and learning-based [52–54].

The method based on 2D feature descriptors is mainly aimed at RGB images. Multi-view projection is performed on the target and multiple color images are obtained as a template. The template records the target pose or the corresponding relationship between the 3D point and the point on the image. In combination with certain feature point matching algorithms, a single image is input to find the 2D pixel points corresponding to the template image. Further 3D points of the target are obtained to solve the target pose problem [11]. The main representative methods are SIFT [12], SURF [13], FAST [14], and ORB [15]. The method based on 3D feature descriptors focuses on point clouds. It is based on the principle of finding the correspondence between two point clouds by using the distance between points, point cloud normal vectors, etc. Then, it represents local features such as point cloud contours, lines, edges, etc. Representative methods based on 3D feature descriptors include RoPS, TriSI, THRIFT, PPFH, etc. [16–18]. However, these methods either have certain requirements on the texture of the target object, and they deal with symmetrical objects [19–22].

Based on the template matching method, the first step is to make a template, which requires multi-view imaging of the target object. But it is different from the feature descriptor, which is finding the corresponding points in the input image and the template image. It mainly extracts the contour, surface curvature and edge information of the target. It has not very intensive requirements for the template [23–25]. The template in the template-based matching method refers to a specific pose. It requires the template to contain as many possible poses as possible, and then the pose estimation problem is turned into a depth estimation and image classification problem. Methods based on template matching include Linemod [26], SSD-6D [27], PoseCNN [28], Recovery 6d [29], Robust 6d etc. [30, 65, 66]. In addition, NOCS and Latentfusion etc. method can use implicit methods to represent a class of objects, which is also a method of template matching [31–33]. Based on the template method, the angle prediction cannot be continuous and the accuracy is too dependent on the template. If the number of templates is too small, the accuracy will decrease. But the number of templates will increase, the classification difficulty will increase and the efficiency will decrease.

3D target detection algorithms based on learning can be divided into two types: one stage and two stage. One is to directly use the neural network to regress or classify the 6D pose of the target, and the other is to first use the neural network to estimate the position distribution of the key points in the image, and then use the mapping relationship between 2D points and 3D points to solve the target pose [34, 35]. The method of using the deep learning algorithm to directly return to the 3D bounding box of the target generally divides the network structure into several branches, which are used to predict the position of the center point, and there are also branches that are used to predict the rotation angle [56, 57]. However, since the rotation of the target is a non-linear change, the method of direct regression is still not ideal in terms of accuracy. Methods based on key points can be divided into multiple categories, according to the selection of key points and the different ways of forecasting [50, 51]. BB8 [36]

and Yolo-6D [37] algorithms choose to predict the 8 vertices on the 3D bounding box of the target as key points, which makes it easier to use the PNP [38] algorithm in the later stage, but since these 8 vertices are not on the target surface, this method robustness is not good. Some scholars choose some points on the target surface with significant texture or color changes as key points, which makes the detection of key points easier, but it cannot guarantee the uniform distribution of key points. When the key points are obscured, the results are also not satisfactory. There are also some methods, such as PvNet [39] and PVN 3D [40] to select several key points on the target surface. There are also methods that directly predict the relative coordinates from pixel-by-pixel to key points. This will introduce two variables and cause the error to be superimposed [47–49]. Other scholars predict the probability that each pixel is a key point, thus forming a heat image. But once the key point is occluded, which is difficult to predict successfully. These methods are not dense enough to predict, which will further affect the robustness of the model.

In summary, there are many methods for quasi-3D target detection at this stage. The method based on feature descriptors has high accuracy, but the real-time performance is not good, and it cannot handle symmetrical targets, relying on the target texture. The method based on template matching can solve the real-time problem, but since the discretization of the continuous problem. It is difficult to have both accuracy and real-time. The method of 3D target detection based on deep learning is a hot research topic in recent years, and it also shows strong competitiveness at this stage. Due to the urgent need of autonomous driving technology, people have done a lot of work on 3D target detection in outdoor scenes. However, with the widespread popularity of service robots, the robots' understanding of indoor scenes needs to be further strengthened. Therefore, it is of great significance to carry out research on 3D target detection in indoor scenes.

3 Materials and methods

3.1 Data set image acquisition

In this paper, research on 3D target detection in indoor scenes, which comprehensively considers factors such as indoor environment complexity, target occlusion, camera performance, subsequent target pose and bounding box acquisition is done. The Aruco code disc was made. Some animal models and daily necessities of moderate size were selected as the targets to be tested. The targets were placed on the Aruco code disc to make the dataset [45, 46]. In the process of making a data set, a batch of images are collected for each scene, and the relative pose between the targets in the same scene is unchanged, only the pose of the camera is changed. Figure 1 is an aligned color depth image collected in a certain scene. During the collection process, the target in the scene is still and only the pose of the moving camera. The resolution of the picture is 640*480 pixels. Since the depth image ranges from 0 to 8 m, it does not look obvious. In order to make the depth image more obviously, we are pseudo-coloring it.

Based on the Aruco code to obtain the target pose information, the Qlone mobile phone software was used to obtain a 3D model of the target, and the acquired images were annotated to obtain color and mask images as shown in Fig. 2.

The obtained annotation file was visualized and a random part of the image was taken as shown in Fig. 3.

It can be found intuitively that the labeling pose information obtained by the above method is relatively accurate. It can be used as labeling information for 3D target detection.

On this basis, we used a rich data set to detect the types of targets, include lions, flashlights, puppies and shoe polish bottles other different shapes and sizes. There are very common targets in daily life and the data set is expanded. Each target has 800 images sets. The pandas, air fresheners and other targets with similar shapes and colors are added to interfere. Meanwhile, in order to increase the generalization performance of the model, the Blender 3D animation software is used to generate the same amount of rendering data as the real dataset. As shown in Fig. 4.

3.2 Selection of key points based on FPS algorithm

For the two-stage 3D target detection algorithm, a very critical step is the selection of key points. Some methods directly select the 8 vertices of the target 3D bounding box as key points, therefore eliminating the subsequent PNP (Perspective-n-Point) pose solving process. However, since the vertices of the target bounding box are generally outside the target pixels, it brings great difficulty to the subsequent prediction. There are also some methods that select some points with significant changes in the color of the target surface texture as feature points, but the distribution of these points cannot be guaranteed, and once they are occluded, it will increase the difficulty of detection. This section will use the FPS [41] algorithm (Farthest Point Sampling) to select points on the target surface as key points as the subsequent detection objects.

The selection of the key points in this section is to make the key points evenly distributed on the target surface, so the geometric center of the target is selected as the initial point. Since the geometric center of the target has been changed to the origin of the world coordinate system in second section, there is only one point in the initial point set (0, 0, 0). When the number of sampling points is set to 4 and 8 respectively, the corresponding key point clouds and RGB images are obtained, as shown in Fig. 5.

Fig. 1 Colorful depth image of partially aligned

4 Theory and calculation (3D target detection in indoor scene based on RGB-D image)

4.1 Key point detection model based on RGB image joint semantic segmentation

4.1.1 Network model

The network structure in this section follows the PvNet model, but it has made some changes in the output and subsequent key point selection. The key point detection model of joint semantic segmentation in this paper is shown in Fig. 6. Using the first half of ResNet, when the feature image is reduced to certain size, the expansion convolution transition is used to ensure the acceptance domain. Then through a series of convolution and nonlinear interpolation, the feature image is restored to the original input size. In order to fully retain the shallow features, the shallow feature image during the sampling process is repeatedly jumped in the feature image of the up-sampling process [8, 42–44]. In terms of output, the section of PvNet has been changed. PvNet predicts semantic segmentation and pixel points to the unit vector of key points, but predicting is the semantic category of pixels and the distance to the key point.

The lower right corner in Fig. 6 is a heat image generated by the network based on the distance from a pixel to a key point. From red to yellow-green to blue, it means that the distance between the pixel point and the key point becomes longer. Compared with the method of unit vector intersection in PvNet, the paper can get the key point information faster and more intuitively. However, this kind of key point detection based on the heat image is difficult to deal with the truncation problem, because once the target is truncated, and the hot image cannot be generated in the truncated part. So the key points projected on the truncation cannot be further obtained.

4.1.2 Based on an improved key point voting scheme for occlusion

In the above network structure, the network chooses to predict the per-pixel distance to the key point, rather than the probability that per-pixel is the key point. This provides a hypothetical solution for the subsequent selection of key points. For each pixel, the distance value is predicted and the error is

Fig. 2 Partial colorful and mask images in the dataset

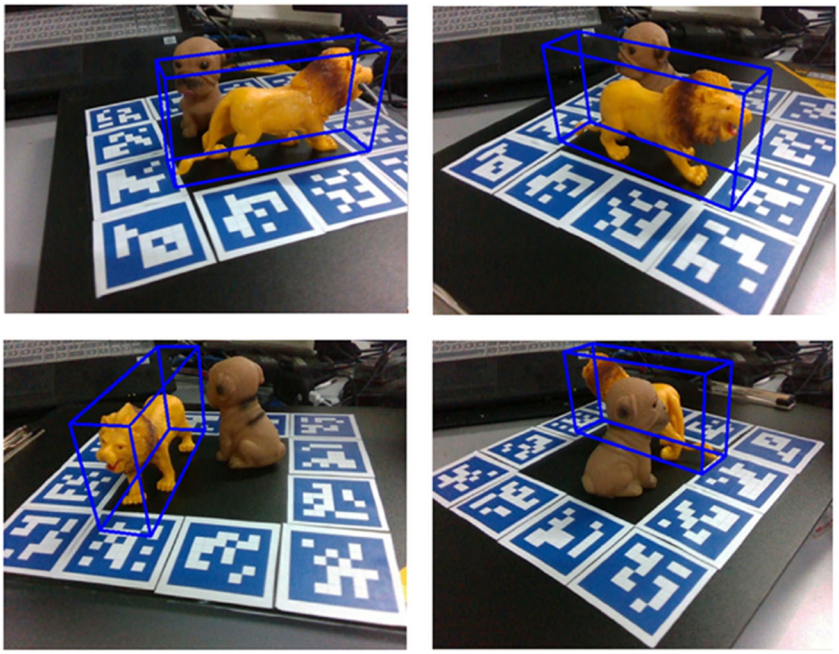


Fig. 3 Visualization of positional annotation information

ignored. The key point must be on the circle with this pixel as the center and the distance predicted by this pixel as the radius. Two circles with different centers have three possible relative positions: intersect at one point, intersect at two points, and disjoint. Ignoring the disjoint case, if two pixels are used to predict key points at the same time, at least one potential key point positive solution will be obtained.

In order to obtain the key points in the truncated case, the two pixels are randomly sequence selected the pixel point as the center of the circle. The distance of the predicted key points is the intersection of the radius and all potential key point sets are obtained. Figure 7(a) shows the potential key points obtained by using four pixels and their predicted distance information. The green dots in the diagram are the predicted key points, the purple triangles are the off-target noise points and the red square dots are the truth key points. Now, the key points are concentrated in half of the noise, but it is obvious that these noises are randomly distributed, and the potential key points are distributed near the real key points. If you want to select the key points from these points, the first thing to do is to remove the noise as much as possible.

Figure 7 reflects the process of removing the noise points in the potential key points. First, the noise points falling outside the target are removed. Secondly, a circular sliding window of suitable size is used to find the position with the most points in the window as a new set of potential key points. Subsequent voting on key points based on RANSAC (Random Sample Consensus) will focus on this key point.

Fig. 4 Partial expanded target dataset and composite dataset images

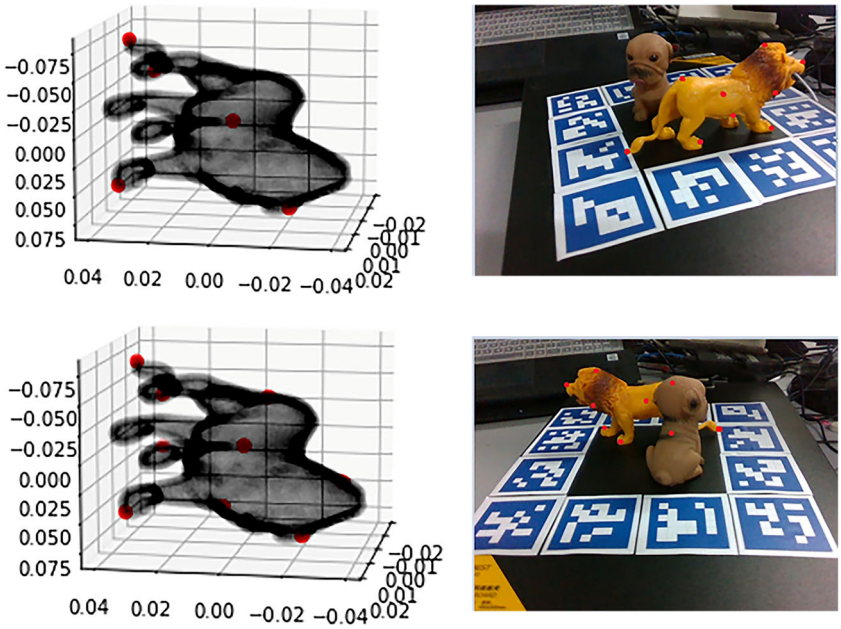


Fig. 5 Distribution of target and image key points

4.1.3 Voting based on key points of RANSAC

RANSAC is a commonly used algorithm in machine vision [58, 59]. It is generally used to select correct data from abnormal data and obtain model parameters that conform to these data. These abnormal data may be due to incorrect measurements, incorrect assumptions, incorrect calculations, etc.

Suppose that for a certain key point x_k, the potential key point set $\{h_{k,i} | \ i = 1, 2, \cdots n\}$ is obtained through the above steps, then all the pixels on the target of detected are voted, and are scored on all the points in the set of potential key points. Defining the score obtained by point $h_{k,i}$ as $\omega_{k,i}$, the formula is shown in Eq. (1).

$$w_{k,i} = \sum_{p \in o} \coprod \left(1 - \frac{|\|x_k - p\|_2 - \|h_{k,i} - p\|_2|}{\|x_k - p\|_2} \right) \tag{1}$$

In which, $\coprod$ denotes the indicator function. p represents a certain pixel.$p \in o$ represents that the pixel belongs to the target to be detected.

The voting score characterizes the relative error between the distance of the pixel to the predicted key point and the distance to the true key point. A higher score means that the hypothesis is better and the point is closer to the key point. For the key point mean u_k and covariance $\sum_k$, it is calculated by the Eq. (2) and Eq. (3).

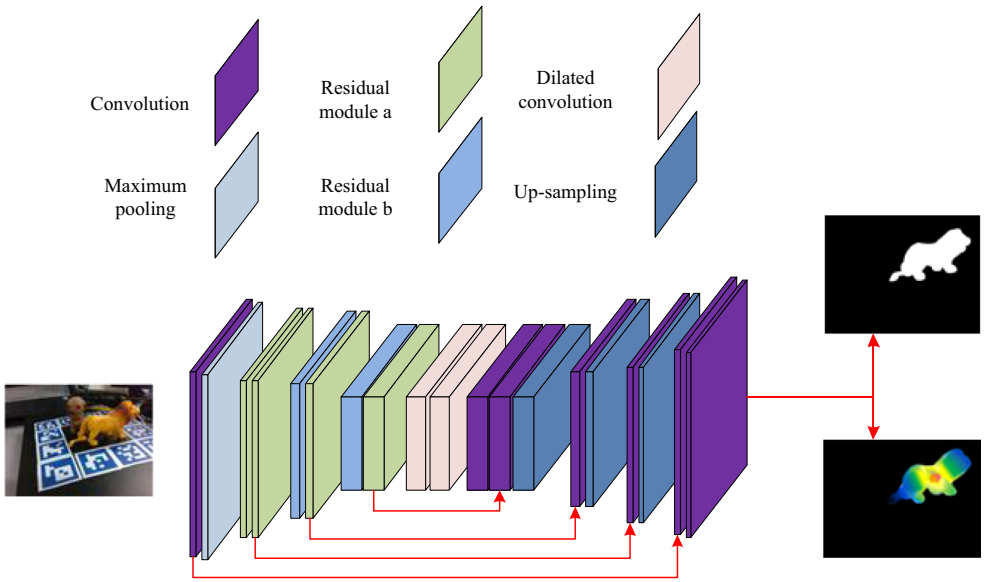


Fig. 6 Key point prediction model of joint semantic segmentation

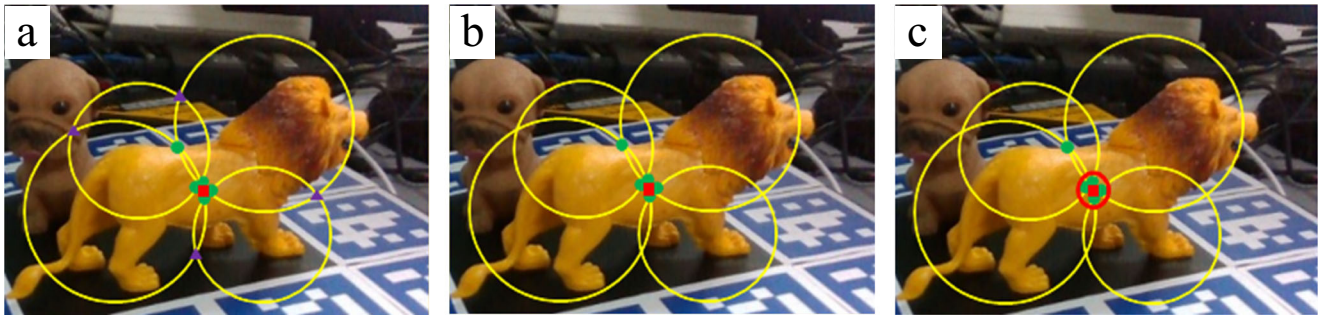


Fig. 7 Noise elimination of potential key point set (**a**: Key point set containing half of the noise, **b**: Key point set after removing the noise outside the target, **c**: Set of key points selected by sliding window)

$$u_k = \frac{\sum_{i=1}^{N} \omega_{k,i} h_k}{\sum_{i=1}^{N} \omega_{k,i}} \tag{2}$$

$$\sum_k = \frac{\sum_{i=1}^{N} \omega_{k,i} \left(h_{k,i} - u_k\right)\left(h_{k,i} - u_k\right)^T}{\sum_{i=1}^{N} \omega_{k,i}} \tag{3}$$

In which, $\omega_{k,\ i}$ represents the score, $h_{k,\ i}$ represents the potential key point set.

Using the camera projection model and 2D points, convert to the camera coordinate system. Create four control points in the camera coordinate system with the same relationships as the world coordinate system. Then solve for the coordinates of these four control points. Finally, the pose problem is further solved using ICP. It is calculated by the Eq. (4) and Eq. (5).

$$\text{minimize} \sum_{k=1}^{k} \left(\left(\overrightarrow{X_k} - u_k \right)^T \sum_{K}^{-1} \overrightarrow{X_k} - u_k \right) \tag{4}$$

$$\overrightarrow{X_k} = \pi(RX_k + t) \tag{5}$$

In which, X_k represents the 3D coordinates of the key points. $\overrightarrow{X_k}$ is a 2D mapping of 3DX_k.

The PNP algorithm initializes the parameters R and t according to four key points whose covariance matrix trajectory is minimized. It is then solved using the Levenberg Marquardt algorithm to minimize the re-projection error.

4.2 3D target detection in indoor scene based on RGB-D image

In the previous section, only RGB images were used to complete key point detection, but pure RGB images lacked scene spatial information. Therefore, this section adds a depth image to the 3D target detection model of indoor scenes based on RGB images. On the one hand, it can improve the accuracy of semantic segmentation. On the other hand, it can increase the prediction of key point depth information.

In this section, a two-channel model of 3D target detection based on RGB-D is established. The general process of the model is shown in Fig. 8. The upper part of the figure uses the RGB-based 3D target detection model, while similar structure is used to train the depth image in the lower part of the model. The only difference is that the color image network has deeper layers and more thorough down sampling. In the down-sampling stage, the feature image obtained from the depth image is connected to the color channel. In the up-sampling stage, the feature image obtained from the color channel is connected to the depth channel. Then the color channel network is used to predict the pixel-by-pixel distance to the key point, and the depth image is used to predict the pixel-by-pixel distance from the key point in the depth direction. The color and depth images are jointly used to obtain target semantic segmentation. Combining the 2D key points and the pixel-by-pixel depth difference, the key points are optimized and projected to 3D, and finally the pose is solved by the ICP (Iterative Closest Point) algorithm [60]. The process of the 3D target detection model based on RGB-D will not be described too much. The following will focus on how to extend the key point detection on the 2D image in section 3 to 3D.

In this section, by adding a deep image channel network, feature fusion is performed at a certain level to improve the accuracy of semantic segmentation. On the other hand, the channel predicts the pixel-by-pixel and key point depth difference. The key point will be projected on the 2D image, but the corresponding depth value of the projection point on the depth image may not be the true depth value of the key point. Because the key point may be on the back side, if the key point is the target center point, it must be Will be blocked. The contour information of the target can be well reflected by the depth image, so the depth image channel network can be used to predict the true depth information of the key points. Meanwhile, in order to avoid the influence of occlusion truncation on the prediction of key point depth information, a method similar to that is adopted in third section. It can predict the difference between the pixel-by-pixel depth value to the key point depth value, and also can predict the key point depth value at the same time for every two pixels. The average value of the depth values of the two-pixel prediction key points is taken as the predicted depth value, and this depth value is aligned on the pixel coordinates of the potential key points

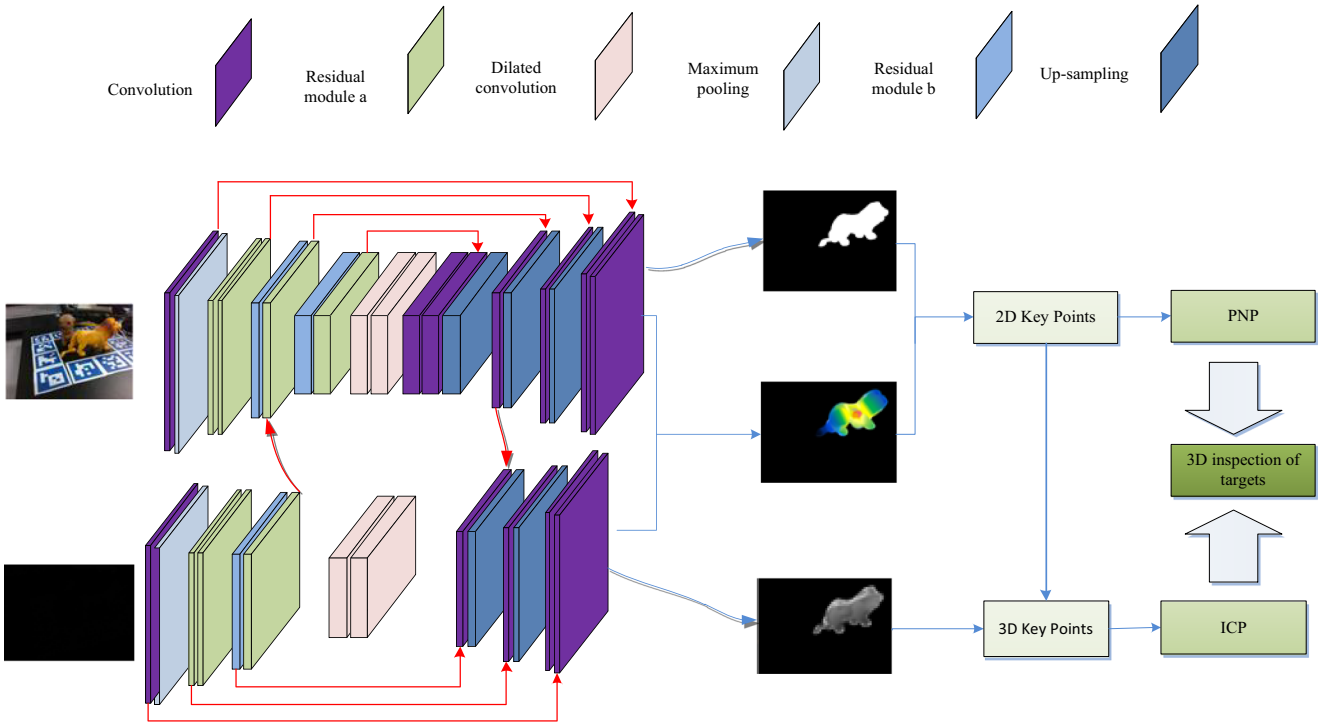


Fig. 8 3D target detection process of indoor scene

predicted by the two pixels. The potential key point set is converted into a 3D potential key point set using the known camera's internal parameters, then the potential key point set is narrowed down using a 3D sliding window, and finally each potential 3D key point is scored using a RANSAC-based poll. Eq. (6) is the score calculation formula for each potential key point.

$$W_{k,i} = \sum_{P \in O} \coprod \left(1 - \frac{\left| \|X_k - P\|_2 - \left\|H_{k,i} - P\right\|_2 \right|}{\|X_k - P\|_2} \right) \tag{6}$$

In which, $\coprod$ denotes the indicator function. P denotes a point in the camera coordinate system. $P \in O$ represents that this point is obtained by combining the pixel coordinates of the target surface and its corresponding depth value with the camera's internal parameters.

X_k is the 3D coordinates of the real key points in the camera coordinate system, which can be obtained by adding the marked camera posture to the world coordinates of the key points. $H_{k,\,i}$ is the 3D point set $\{H_{k,\,i} | \; i = 1, 2, \cdots n\}$ corresponding to the potential 2D key point set $\{h_{k,\,i} | \; i = 1, 2, \cdots n\}$. $W_{k,\,i}$ is the score of potential key points.

In terms of loss function, semantic segmentation still uses cross-entropy loss. Bring in Eq. (7). The loss function of key points is defined as Eq. (8).

$$\Delta d_k(P, w) = \frac{\left| \|X_k - P\|_2 - \|X_k - P\|_2 \right|}{\|X_k - P\|_2} \tag{7}$$

$$\varphi(w) = \sum_{k=1}^{k} \sum_{P \in O} \varphi\Big(\Delta d_k(P, w) \tag{8}$$

In which, w denotes network parameters. P denotes a point in the camera coordinate system. $\Delta d_k(P,\ w)$ is the relative error of the distance between the predicted key point and the real key point.X_k denotes the real 3D coordinates of key points.

5 3D starget detection experiment in indoor scene based on RGB-D image

5.1 Model training parameter configuration

This article uses the Pytorch deep learning framework. Pytorch is a deep learning framework developed by Facebook based on Python. Its advantage uses the Python version of the Torch library in processing images, which is conducive to providing dynamic graph calculations and making it easier to run on the GPU. This article uses the Python deep learning framework to configure and build the environment under the Ubuntu 18.04 × 86 hardware environment. In order to improve the training speed of the neural network, GPU is used to accelerate the calculation of the neural network, and the computing power is improved by configuring Cuda and Cudnn. The main parameter configuration of the experimental environment is shown in Table 1.

Table 1 Related parameters of the experimental environment

Classification name	Detail
Operating system	Ubuntu 18.04
CPU	Intel core i7
GPU	GTX 2070
RAM	16G
Cuda with Cudnn	10.0,7.0
Pytorch	1.4
Python	3.6
Opencv	3.0

When training the network model in this paper, the various parameter configurations are as follows: the learning rate is 0.001, the learning rate is halved every 20 epochs, and all models are trained for 240 epochs. Adam algorithm can converge faster than the stochastic gradient descent algorithm. So Adam algorithm is used to optimize the network, and the key points are defined as 8.

5.2 Evaluation criteria

In this paper, the 3D target detection adopts the joint semantic segmentation method, and the pose-corresponding mask is used in the training process to constrain the semantic segmentation. Because the projection mask is also fixed for determining the pose of the target. This paper uses average precision (AP) to judge the accuracy of the graph. If the predicted pose corresponding to the mask overlaps the real pose corresponding to more than 70%, the prediction is considered to be successful. Eq. (9) is the specific formula.

$$AP = \frac{\sum_{i=1}^{n} Precision_i}{n} \tag{9}$$

In which, n denotes the number of verified images. $Precision_i$ denotes the accuracy of the mask corresponding to the pose on the image. In the data set of this article, there is only one or no target to be detected in the general image, and the recognition accuracy of each image for target detection is either 0 or 1.

In terms of pose evaluation, this article uses two commonly used evaluation criteria to evaluate the model of this article: 2D mapping quantification and 3D distance (Average Distance to the 3D model, ADD) quantification. The 2D Projection metric calculates the average pixel distance between the 2D projection point of the target and the predicted 2D projection point of the target in the real pose. If the distance is less than 5 pixels, the detection is considered correct. However, the ADD metric then uses the real and predicted obtained target pose. In combination with the target point cloud model, all 3D coordinate points of the target are obtained. Calculate the sum of the distances between all corresponding points, and finally take the average value. If the average distance difference is less than 10% of the target diameter, the detection is considered successful. Eq. (10) is the calculation formula for the 2D mapping metric.

$$\text{2D\_Projection metric} = \frac{1}{m}\sum \left\| x_p - x_t \right\|_2 \tag{10}$$

In which, m is the number of points on the target point cloud model file. x_p and x_t are respectively the predicted target pose and the real target pose. Combined with the camera's internal parameters, the projection of the corresponding point on the two-dimensional image is obtained.

Eq. (11) is the calculation formula of ADD metric.

$$ADD = \frac{1}{m}\sum_{X \in O} \left\| (R_p X + T_p) - (R_t X + T_t) \right\|_2 \tag{11}$$

In which, X is a 3D point. $X \in O$ represents the point on the 3D model of the target. R_p and T_p represent the predicted target posture. R_t and T_t represent the real target posture.

5.3 Result analysis

In terms of loss function, the loss function of the 3D target detection algorithm is equal to the weighted sum of the cross-entropy loss of semantic segmentation and the key point loss, which is visualized by tensorboard. Since there are more model trainings in this article, and the parameters of the model have not changed under the same data input. So, this section will only analyze the loss function changes and recognition rate of the lion model trained on RGB-D data, as shown in Fig. 9.

It is clearly evident from the Fig. 9 that the overall change trend of various loss functions is the same. The overall loss function drops sharply to 0.015 between 0 and 10 k iterations, which means that the model has learned a lot of shallow features during this time. From 10 k to 15 k, it starts to slowly drop to 0.01, then oscillates around 0.01. Finally, basically reaches a state of convergence indicating that the model is well trained.

Meanwhile, the model in this chapter is evaluated on the validation set every epoch. The changes in the evaluation indicators and the overall loss function in the evaluation process are shown in Fig. 10. It can be found that the various evaluation indicators of the model converge to a higher level, and the overall loss function is also consistent with the training set. At the same time, it can be found that the average accuracy of the model and the 2D projection mapping metric are both close to 1. It is not that the model achieves over-fitting, but this is determined

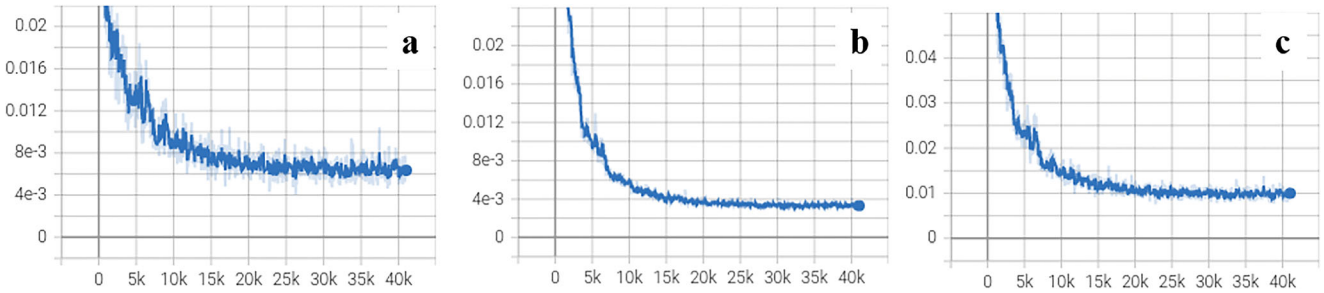


Fig. 9 Changes in various loss functions of 3D target detection (**a**: Semantic segmentation loss function changes, **b**: Key point voting loss function changes, **c**: Overall loss function change)

by the rubric, which results in the error in the projected posture being suitably reduced.

In this paper, the recognition effect of different targets under different input data is shown in Fig. 11. It can be seen from Fig. 11: after adding the depth image, the model corresponding to each target has improved in different evaluation standards. After adding the depth data to the average accuracy, the AP value increased by 3.2%, the 2D projection value increased slightly, and the ADD increased by about 7%.

There are two main reasons for the above results: On the one hand, because the 3D target detection in this article is based on the key point detection of pixel-by-pixel voting based on semantic segmentation, the key point detection largely depends on the accuracy of semantic segmentation. However, with the addition of depth images, illumination and occlusion can be well overcome to better distinguish foreground hind-grounds improving semantic segmentation accuracy. The accuracy of key points is improved and the accuracy of further poses is improved. On the other hand, relative depth distance prediction is added to the RGB-based key point detection. The key points are further pushed from 2D to 3D, and the target poses are solved directly using ICP with better accuracy. Finally, the model was tested on the RTX 2070 graphics card. When the input data were RGB images and RGB-D images, the speed of the model could reach 21 frames per second and 18 frames per second respectively. Test on the validation set and visualize the results shown in Fig. 12.

In Fig. 12, the blue represents the real label box, and the green represents the predicted 3D bounding box. The final recognition effect obtained from the visualization results is still relatively ideal. In order to further verify the effectiveness of the algorithm in this paper, the PvNet algorithm is used to train and test on the data set of this paper, and the evaluation criterion is ADD, and the results shown in Fig. 13 are obtained. It can be seen from the table that the algorithm based on RGB-D image proposed in this paper is better than the PvNet algorithm, and the PvNet algorithm is better than the algorithm based on

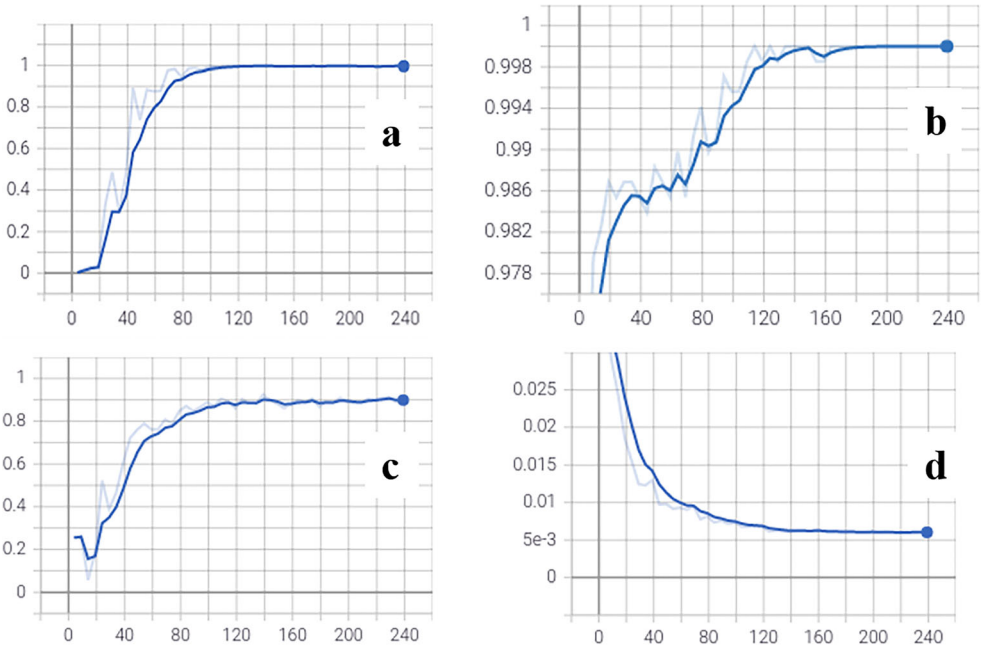
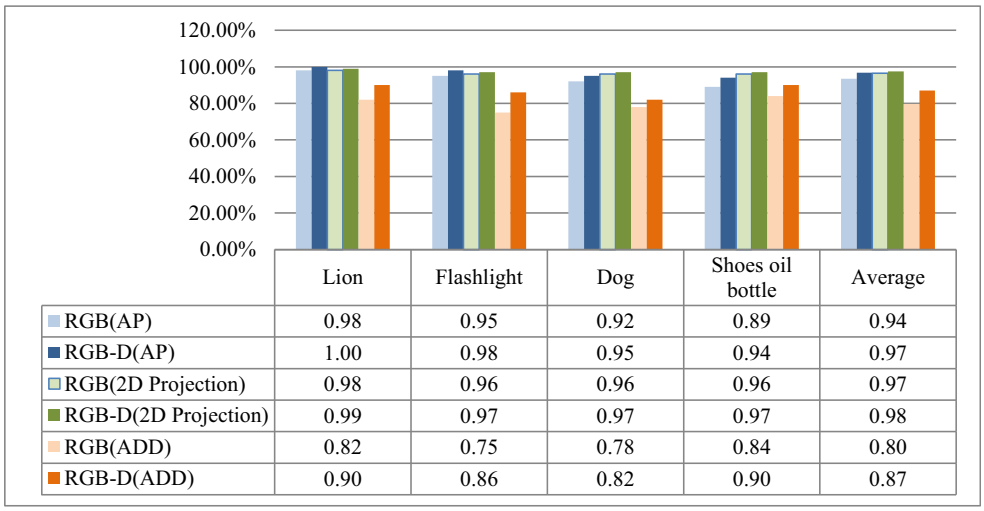


Fig. 10 Evaluation indicators and overall loss changes on the validation set (**a**: AP, **b**: 2D Projection, **c**: ADD, **d**: Overall loss function)



	Lion	Flashlight	Dog	Shoes oil bottle	Average
RGB(AP)	0.98	0.95	0.92	0.89	0.94
RGB-D(AP)	1.00	0.98	0.95	0.94	0.97
RGB(2D Projection)	0.98	0.96	0.96	0.96	0.97
RGB-D(2D Projection)	0.99	0.97	0.97	0.97	0.98
RGB(ADD)	0.82	0.75	0.78	0.84	0.80
RGB-D(ADD)	0.90	0.86	0.82	0.90	0.87

Fig. 11 Accuracy comparison of each model under different input types and different criteria

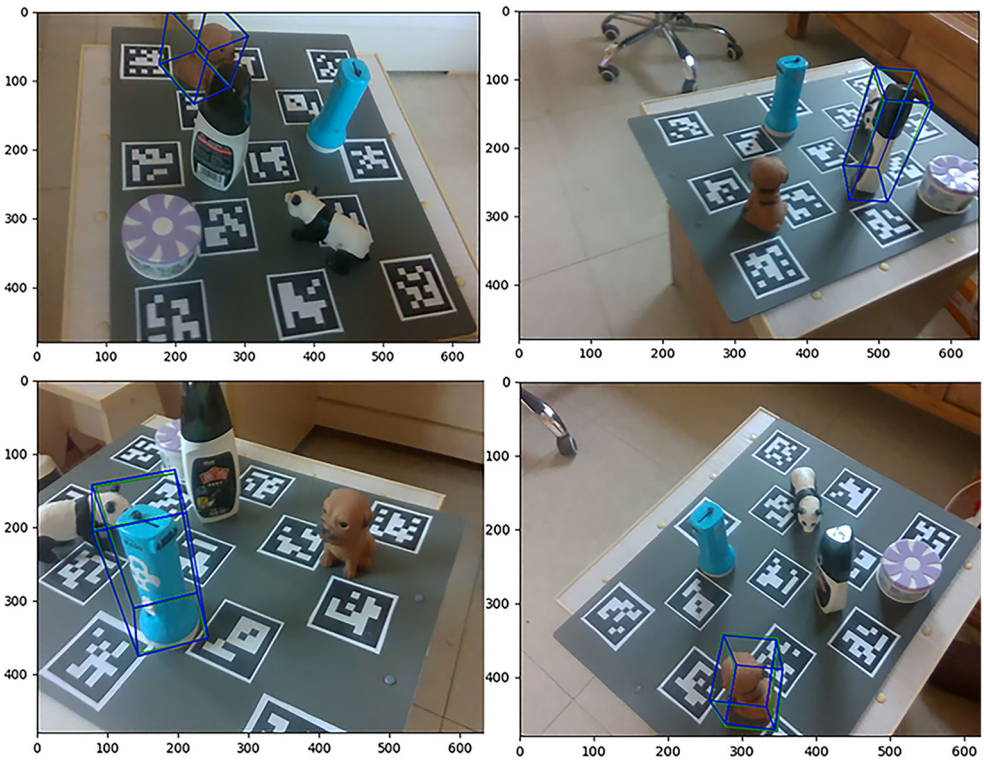


Fig. 12 Models tested and visualized on the validation set

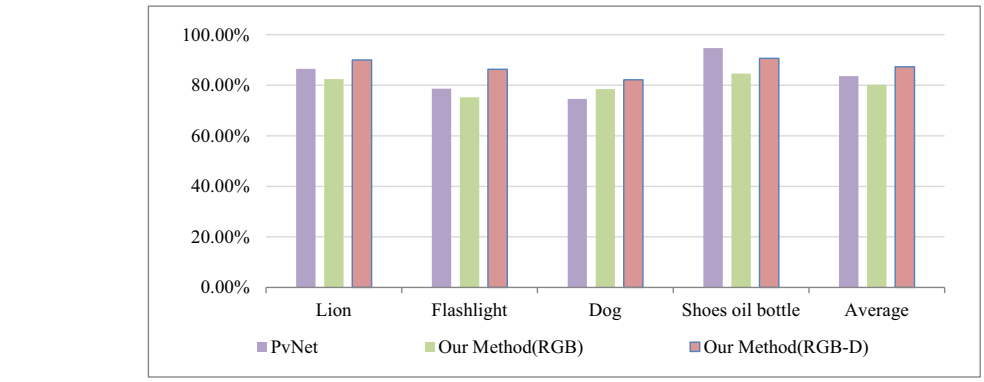


Fig. 13 Comparison of recognition rate (ADD) between this method and PvNet method

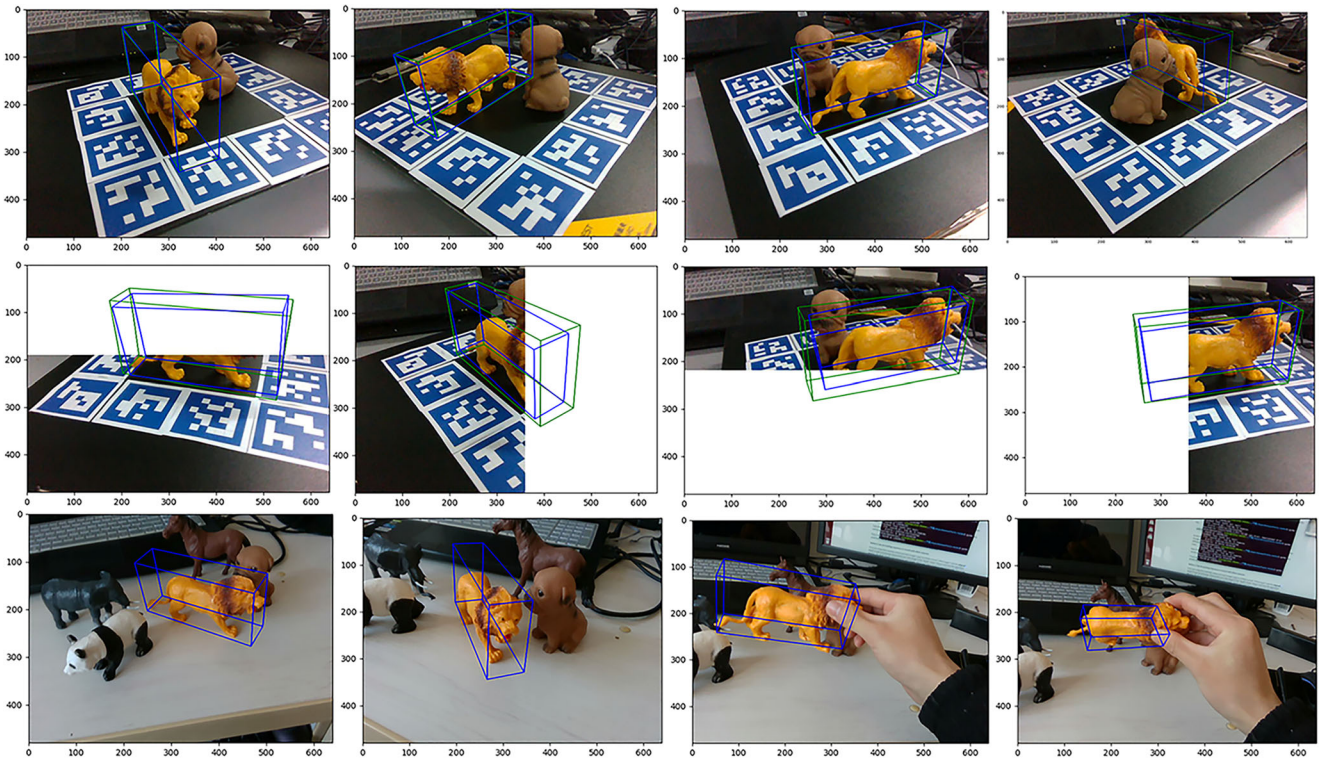


Fig. 14 Visualization of experimental results

RGB image proposed in this paper. In general, the key point detection of pixel-by-pixel distance voting for joint semantic segmentation proposed in this paper has achieved the expected effect. At the same time, the method of extending 2D key points to 3D combined with the depth image is feasible and effective.

In order to verify the recognition effect of the method in this paper on truncated data, the best trained lion model based on RGB-D is selected. The images on its verification set are appropriately truncated, and the obtained model is used to identify the lion model in a random scene without marking. The visualization of the result is shown in Fig. 14.

The first row shows the effect of the RGB-D-based 3D target detection model on the verification set to detect and visualize. The predicted 3D bounding box basically coincides with the real labeled box. The second row shows the detection effect of the model under different levels of truncation. It is predicted that the 3D bounding box and the real bounding box have a certain degree of offset, but in general it can still reflect the 3D space position occupied by the target. The last row shows the detection effect of the model in unlabeled scenes. In most cases, it can successfully detect the 3D pose of the target. The above experimental results show that our method in this paper can obtain good results on the data set made by ourselves, and it can handle truncated images well at the same time.

6 Conclusion

In this paper, a key point voting model based on joint semantic segmentation of RGB images is proposed. On this basis, depth image branching channels are added to enhance the accuracy of semantic segmentation. Meanwhile, the 2D key point detection is extended to three-dimensions by predicting the pixel-by-pixel depth distance difference. The pose is optimized by the ICP algorithm, and the 3D target detection of indoor scene based on RGB-D images is completed. Finally, the reliability of the method is proved by analyzing and comparing various evaluation metrics and model visualizations. The validation set, truncated validation set and unlabeled three cases are tested and visualized. The superiority and good generalization performance of the method in this paper are further proved, so as to better solution the problems of small individual target, multiple types and truncation in indoor scenes.

Acknowledgments This work was supported by grants of the National Natural Science Foundation of China (Grant Nos.52075530, 51575407, 51505349, 61733011, 41906177); the Grants of Hubei Provincial Department of Education (D20191105); the Grants of National Defense PreResearch Foundation of Wuhan University of Science and Technology (GF201705) and Open Fund of the Key Laboratory for Metallurgical Equipment and Control of Ministry of Education in Wuhan University of Science and Technology (2018B07,2019B13).

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Proces Syst 25(2):1097–1105
2. Jiang D, Li G, Sun Y, J.Hu, Yun J, Liu Y (2021) Manipulator grabbing position detection with information fusion of color image and depth image using deep learning. J Ambient Intell Humaniz Comput 12(12):10809–10822
3. Wang L, Li R, Sun J, Liu X, Zhao L, Seah H, Quah C, Tandianus B (2019) Multi-view fusion-based 3D object detection for robot indoor scene perception. Sensors 19(19):4092
4. Kuang H, Wang B, An J, Zhang M, Zhang Z (2020) Voxel-FPN: multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds. Sensors 20(3):704
5. Sun Y, Weng Y, Luo B, Li G, Tao B, Jiang D, Chen D (2020) Gesture recognition algorithm based on multi-scale feature fusion in RGB-D images. IET Image Process 14(15):3662–3668
6. Rahman M, Tan Y, Xue J, Shao L, Lu K (2018) 3D object detection: learning 3D bounding boxes from scaled down 2D bounding boxes in RGB-D images. Inf Sci 476:147–158
7. Tan C, Sun Y, Li G, Jiang D, Chen D, Liu H (2020) Research on gesture recognition of smart data fusion features in the IoT. Neural Comput & Applic 32(22):16917–16929
8. Jiang D, Li G, Tan C, Huang L, Sun Y, Kong J (2021) Semantic segmentation for multiscale target based on object recognition using the improved faster-RCNN model. Futur Gener Comput Syst 123:94–104
9. Arnold E, Dianati M, Temple R, Fallah S (2020) Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2020.3028424
10. Wu Y, Qin H, Liu T, Liu H, Wei Z (2019) A 3D object detection based on multi-modality sensors of USV. Appl Sci 9(3):535–543
11. Lepetit V, Moreno N, Fua P (2009) EPnP: An accurate on solution to the PnP problem. Int J Comput Vis 81(2):155–166
12. Guan Y, Li W, Zhang B, Han B, Ji M (2020) Multi-label classification by formulating label-specific features from simultaneous instance level and feature level. Appl Intell 9:1–16
13. Bay H (2006) SURF: speeded up robust features. Comput Vision Image Understanding 110(3):404–417
14. Li G, Li J, Ju Z, Sun Y, Kong J (2019) A novel feature extraction method for machine learning based on surface electromyography from healthy brain. Neural Comput Applic 31(12):9013–9022
15. Ma C, Hu X, Xiao J, Du H, Zhang G (2020) Improved ORB algorithm using three-patch method and local gray difference. Sensors 20(4)
16. Li W, Wang J, Qi (2010) Spin-image surface matching based target recognition in laser radar range imagery. Chinese Physics B 19(10): 281–288
17. Guo Y, Sohel F, Bennamoun M, Lu M, Wan J (2013) Rotational projection statistics for 3D local surface description and object recognition. Int J Comput Vis 105(1):63–86
18. Shih J, Chen H (2009) A 3D model retrieval approach using the interior and exterior 3D shape information. Multimed Tools Appl 43(1):45–62
19. Li G, Jiang D, Zhou Y, Jiang G, Kong J, Manogaran G (2019) Human lesion detection method based on image information and brain signal. IEEE Access 7:11533–11542
20. Prakhya S, Liu B, Lin W, Jakhetiya V, Guntuku S (2017) B-SHOT: a binary 3D feature descriptor for fast Keypoint matching on 3D point clouds. Auton Robot 41(7):1501–1520
21. Guo Y, Bennamoun M, Sohel F, Lu M, Wan J (2016) A comprehensive performance evaluation of 3D local feature descriptors. Int J Comput Vis 116(1):66–89
22. Kamranian Z, Sadeghian H, Nilchi A, Mehrandezh M (2020) Fast, yet robust end-to-end camera pose estimation for robotic applications. Appl Intell 3:1–19
23. Li G, Tang H, Sun Y, Kong J, Jiang G, Jiang D, Tao B, Xu S, Liu H (2019) Hand gesture recognition based on convolution neural network. Clust Comput 22(Suppl.2):2719–2729. https://doi.org/10.1007/s10586-017-1435-x
24. Tejani A, Kouskouridas R, Doumanoglou A, Tang D, Kim T (2017) Latent-class hough forests for 6 DOF object pose estimation. IEEE Trans Pattern Anal Mach Intell 40(1):119–132
25. Jfa B, Lm B, Zhi B (2020) A registration method of point cloud to CAD model based on edge matching. Optik 219:165223
26. Zhang T, Yang Y, Zeng Y, Zhao Y (2020) Cognitive template-clustering improved linemod for efficient multi-object pose estimation. Cogn Comput 12(4):834–843
27. Zhai S, Shang D, Wang S, Dong S (2020) DF-SSD: an improved ssd object detection algorithm based on densenet and feature fusion. IEEE Access 8:24344–24357
28. Jiang D, Li G, Sun Y, Kong J, Tao B (2019) Gesture recognition based on skeletonization algorithm and CNN with ASL database. Multimed Tools Appl 78(21):29953–29970
29. Liu F, Fang P, Yao Z, Fan R, Pan Z, Sheng W, Yang H (2019) Recovering 6D object pose from RGB indoor image based on two-stage detection network with multi-task loss. Neurocomputing 337: 15–23
30. Guo J, Xing X, Quan W, Yan D, Gu Q (2021) Efficient center voting for object detection and 6D pose estimation in 3D point cloud. IEEE Trans Image Process: 1–1
31. Omachi S, Omachi M (2007) Fast template matching with polynomials. IEEE Trans Image Process 16(8):2139–2149
32. Liu Y, Zhou B, Han C, Guo T, Qin J (2020) A novel method based on deep learning for aligned fingerprints matching. Appl Intell 50(2):397–416
33. Spratling M (2019) Explaining away results in accurate and tolerant template matching. Pattern Recogn 104:107337
34. Huang L, He M, Tan C, Jiang D, Li G, Yu H (2020) Jointly network image processing: multi-task image semantic segmentation of indoor scene based on CNN. IET Image Process 14(15):3689–3697
35. Liao S, Li G, Li J, Jiang D, Jiang G, Sun Y, Tao B, Zhao H, Chen D (2020) Multi-object intergroup gesture recognition combined with fusion feature and KNN algorithm. J Intell Fuzzy Syst 38(3):2725–2735
36. Lu X, Tatarczak A, Lyubopytov V, Monroy I (2017) Optimized eight-dimensional lattice modulation format for IM-DD 56 Gb/s optical interconnections using 850 nm VCSELs. J Lightwave Technol 35(8):1407–1414
37. Gai R, Chen N, Yuan H (2021) A detection algorithm for cherry fruits based on the improved YOLO-v4 model. Neural Comput Appl: 1–12
38. Cao M, Jia W, Zhao Y, Li S, Liu X (2018) Fast and robust absolute camera pose estimation with known focal length. Neural Comput Applic 29(5):1383–1398

39. Peng S, Zhou X, Liu Y, Lin H, Huang Q, Bao H (2020) PVNet: pixel-wise voting network for 6dof object pose estimation. IEEE Trans. Pattern Anal Mach Intell: 1–1
40. Duan H, Sun Y, Cheng W, Jiang D, Yun J, Liu Y, Liu Y, Zhou D (2021) Gesture recognition based on multi-modal feature weight. Concurrency Comput: Pract Exper 33(5):e5991. https://doi.org/10.1002/cpe.5991
41. Eldar Y, Lindenbaum M, Porat M, Zeevi Y (1997) The farthest point strategy for progressive image sampling. IEEE Trans Image Process A Publ IEEE Signal Process Soc 6(9):1305–1315
42. Saha S, Mou L, Qiu C, Zhu X, Bovolo F (2020) Unsupervised deep joint segmentation of multitemporal high-resolution images. IEEE Trans Geosci Remote Sens 58(12):8780–8792
43. Liu H, Wu W, Wang X, Qian Y (2018) RGB-D joint modelling with scene geometric information for indoor semantic segmentation. Multimed Tools Appl 77(17):22475–22488
44. Huang J, Liu B, Fu L (2020) Joint multi-scale discrimination and region segmentation for person re-ID. Pattern Recogn Lett 138: 540–547
45. Liu Y, Jiang D, Duan H, Sun Y, Li G, Tao B, Yun J, Liu Y, Chen B (2021) Dynamic gesture recognition algorithm based on 3D convolutional neural network. Comput Intell Neurosci 2021: 4828102–4828112
46. Oscadal P, Heczko D, Vysocky A, Mlotek J, Novak P, Virgala I, Sukop M, Bobovsky Z (2020) Improved pose estimation of aruco tags using a novel 3D placement strategy. Sensors 20(17):4825
47. He W, Cai J, Xiong G, Zhou K (2018) Improved reversible data hiding using pixel-based pixel value grouping. J Light Electronoptic 157:68–78
48. Liu T, Li G, Nie X, Wang H, Zhang D (2021) Enhancement of contour smoothness by substitution of interpolated sub-pixel points for edge pixels. IEEE Access: 1–1
49. Lu J, Wang Z, Hua B, Chen K (2020) Automatic point cloud registration algorithm based on the feature histogram of local surface. PLoS One 15(9):e0238802
50. JuHyok U, Lu P, Kim C, Ryu U, Pak K (2020) A new LSTM based reversal point prediction method using upward/downward reversal point feature sets. Chaos, Solitons Fractals 132:109559
51. Vasquez J, Perez S, Travieso C, Alonso J (2013) Meteorological prediction implemented on field-programmable gate array. Cogn Comput 5(4):551–557
52. Tian J, Cheng W, Sun Y, Li G, Jiang D, Jiang G, Tao B, Zhao H, Chen D (2020) Gesture recognition based on multilevel multimodal feature fusion. J Intel Fuzzy Syst 38(3):2539–2550
53. Lu L, Li H, Ding Z, Guo Q (2020) An improved target detection method based on multiscale features fusion. Microw Opt Technol Lett 62(9):3051–3059
54. Cheng Y, Li G, Yu M, Jiang D, Yun J, Liu Y, Liu Y, Chen D (2021) Gesture recognition based on surface electromyography-feature image. Concurr Comput: Pract Exper 33(6):e6051.https://doi.org/10.1002/cpe.6051
55. Liu Y, Jiang D, Tao B, Qi J, Jiang G, Yun J, Huang L, Tong X, Chen B, Li G (2022) Grasping posture of humanoid manipulator based on target shape analysis and force closure. Alexandria Eng J 61(5):3959–3969. https://doi.org/10.1016/j.aej.2021.09.017
56. Chen Y, Guo B, Shen Y, Wang W, Suo X, Zhang Z (2020) Using efficient group pseudo-3d network to learn spatio-temporal features. SIViP 15(2):361–369
57. Weng Y, Sun Y, Jiang D, Tao B, Liu Y, Yun J, Zhou D (2021) Enhancement of real-time grasp detection by cascaded deep convolutional neural networks. Concurr Comput: Pract Exp 5(33): e5976. https://doi.org/10.1002/cpe.5976
58. Ma S, Guo P, You H, He P, Li H (2021) An image matching optimization algorithm based on pixel shift clustering RANSAC. Inf Sci 562:452–474
59. Zhang S, Li S, Zhang B, Peng M (2020) Integration of optimal spatial distributed tie-points in RANSAC-based image registration. Eur J Remote Sensing 53(1):67–80
60. Jiang D, Zheng Z, Li G, Sun Y, Kong J, Jiang G, Xiong H, Tao B, Xu S, Liu H, Ju Z (2019) Gesture recognition based on binocular vision. Cluster Comput 22(Supple.6):13261–13271. https://doi.org/10.1007/s10586-018-1844-5
61. He Y, Li G, Liao Y, Sun Y, Kong J, Jiang G, Jiang D, Liu H Gesture recognition based on an improved local sparse representation classification algorithm. Cluster Comput 22(Supple.5):10935–10946. https://doi.org/10.1007/s10586-017-1237-1
62. Xie P, Zhang L, Du C, Wang X, Zhong W (2021) Space target attitude estimation from ISAR image sequences with key point extraction network. IEEE Signal Processing Lett: 1–1
63. Liao S, Li G, Wu H, Jiang D, Liu Y, Yun J, Liu Y, Zhou D (2021) Occlusion gesture recognition based on improved SSD. Concurr Comput: Pract Exper 33(6):e6063. https://doi.org/10.1002/cpe.6063
64. Luo B, Sun Y, Li G, Chen D, Ju Z (2020) Decomposition algorithm for depth image of human health posture based on brain health. Neural Comput & Applic 32(10):6327–6342
65. Du S, Xu G, Zhang S, Zhang X, Yue G, Chen B (2020) Robust rigid registration algorithm based on pointwise correspondence and correntropy. Pattern Recogn Lett 132:91–98
66. Xiao F, Li G, Jiang D, Xie Y, Yun J, Liu Y, Huang L, Fang Z (2021) An effective and unified method to derive the inverse kinematics formulas of general six-DOF manipulator with simple geometry. Mech Mach Theory 159:104265
67. Liu X, Jiang D, Tao B, Jiang G, Sun Y, Kong J, Tong X, Zhao G, Chen B (2022) Genetic algorithm-based trajectory optimization for digital twin robots. Front Bioeng Biotechnol 9:793782. https://doi.org/10.3389/fbioe.2021.793782
68. Liu Y, Jiang D, Yun J, Sun Y, Li C, Jiang G, Kong J, Tao B, Fang Z (2022) Self-tuning control of manipulator positioning based on fuzzy PID and PSO algorithm. Fronti Bioeng Biotechnol 9: 817723. https://doi.org/10.3389/fbioe.2021.817723