Hindawi
Scientific Programming
Volume 2022, Article ID 4037625, 14 pages
https://doi.org/10.1155/2022/4037625

Research Article

A Deep Neural Network-Based Target Recognition Algorithm for Robot Scenes

Lijing Liu

School of Engineering, Hong Kong University of Science and Technology, Hong Kong 999077, China

Correspondence should be addressed to Lijing Liu; liulijing1997@cumt.edu.cn

Received 7 December 2021; Revised 22 December 2021; Accepted 28 December 2021; Published 11 January 2022

Academic Editor: Muhammad Usman

Intelligent robots are a key vehicle for artificial intelligence and are widely employed in all aspects of everyday life and work, not just in the industry. One of the talents required for intelligent robots to complete their jobs is the capacity to identify their environment, which is a crucial obstacle to be overcome. Deep learning-based target identification algorithms currently do not fully leverage the link between high-level semantic and low-level detail information in the prediction step and hence are less successful in recognizing tiny target objects. Target recognition via vision sensors has also improved in accuracy and efficiency because of the development of deep learning. However, due to the insufficient usage of semantic information and precise texture information of underlying characteristics, tiny target recognition remains a difficulty. To address the aforementioned issues, we propose a target detection method based on a jump-connected pyramid model to improve the target detection performance of robots in complex scenarios. In order to verify the effectiveness of the algorithm, we designed and implemented a software system for target detection of intelligent robots and performed software integration of the proposed algorithm model with excellent experimental results. These experiments reveal that, when compared to other algorithms, our suggested algorithm's characteristics have higher flexibility and robustness and can deliver a higher scene classification accuracy rate.

1. Introduction

Intelligent robots have become widely employed in many industries as a result of the fast expansion of the economy in recent years, as well as the rapid emergence of artificial intelligence [1]. The use of these technologies has increased the efficiency of automated manufacturing while also satisfying the demand for services in a variety of sectors, hence improving human life quality [2]. Intelligent robots' "intelligence" is based on their capacity to detect their surroundings and interact with people and objects [3]. Robots can utilize machine vision systems to grasp information in the same way that humans rely largely on their eyes to understand the world and their environment. The capacity to swiftly and reliably locate and distinguish things in pictures is one of the most significant areas of research.

Coverage, surveillance, search, patrolling, monitoring, and pursuit-evasion are only a few of the decisional issues that target detection and tracking involve. The use of intelligent robotic target detection technology offers a wide range of applications [4]. In the sphere of security, security robots may undertake real-time video monitoring of public locations [5] such as residential neighborhoods, supermarkets, banks, and junctions. Tour guide robots [6] in the service industry may detect and identify targets in real time, such as automobiles traveling on campuses or scenic places, pedestrians arriving and exiting, and attraction signs and signage, to provide guests with prompt politeness and advice. In industry, target detection technology [7] for industrial robots may be utilized for tasks like workpiece identification and component damage detection, which not only saves time but also enhances productivity. Figure 1 depicts an intelligent power inspection robot that replaces manual labour to accomplish automatic detection and intelligent analysis of the state of power equipment, therefore enhancing the grid's and equipment's safety. This has enhanced the grid's and electrical equipment's dependability significantly.

Figure 1: Electric power intelligent inspection robot.

The objective of target detection entails both identification and localization [8]. This signifies that all of the target categories to be detected in the image have been recognized, and their locations have been computed. It is critical to verify that the target is fully and precisely recognized and that the target's position is exact enough. In general, the target's class must be recognized, the label must be identified, and the target's position box must be defined by the top left and bottom-right coordinates. Although there are usually just a few target instances in a picture, the number of alternative places and sizes to examine is tremendous. In some detection tasks, the bounding box of the target must be established [9] and also the position of pose information or certain tiny local targets must be detected [10]. Once the location of a water glass has been determined, features such as the orientation of the handle must be sought to offer the positional pose information required to grasp the target.

Many classic target detection approaches [11] work well for fixed targets of a given kind or detection tasks in specific settings, but they are ineffective for detecting many targets in complicated surroundings. At the same time, detection speed is a crucial measure, and real-time target detection is required in many applications, which classical target detection algorithms are unable to satisfy. Deep learning's superior performance in image identification, particularly convolutional neural networks, has made deep learning-based target detection and recognition a prominent study issue in recent years. A deep neural network-based target identification system for robot situations has been developed to provide outstanding target recognition performance and universality. The following are some of the contributions:

(i) Firstly, we address the problems in the application of target detection algorithms for mobile robots

(ii) Secondly, we propose a target detection method based on a jump-connected pyramid model to improve the target detection performance of robots in complex scenarios

(iii) Thirdly, we design and implement a software system for target detection of intelligent robots and perform software integration of the proposed algorithm model with excellent experimental results

(iv) Finally, we verify the effectiveness of the algorithm, experiments were carried out on several different datasets, and the results confirmed the effectiveness of the algorithm

The remainder of our work is organized as follows: Section 2 shows the related works, Section 3 represents the methodology that we have adopted for our work, Section 4 explains the experimental work and graphical representation, and in Section 5, we conclude our work.

2. Related Work

In this section, we discuss works of researchers related to our proposed work.

2.1. Current Status of Research on Target Detection Methods. Target detection has been a major research topic in the field of computer vision. Target detection is used to determine where targets are in space and which category they belong to: pedestrian detection [12], face detection [13], vehicle detection [14], intelligent surveillance [15], and autonomous driving [16]; among other applications, target detection is now widely used in our daily lives [17]. Traditional target identification methods are divided into three parts, as shown in Figure 2: According to this figure, potential regions in a given picture, also known as candidate windows, are selected, and then features from the targeted area of the image are extracted using a features extraction procedure. After that, other categorization methods are used. They categorize the targeted region based on the results of focused detection. These classifiers are also used to train a classifier for classification using the obtained information.

2.1.1. Candidate Area Selection. Targets can appear in any area in the image and be huge or little, and their forms and dimensions are not set in the image in the real world. To traverse the entire image without missing any possible locations, a sliding window approach is used, in which different sizes and aspect ratios are assigned to the windows, causing them to slide across the image from left to right and from top to bottom, and then these windows are used for the subsequent feature extraction work.

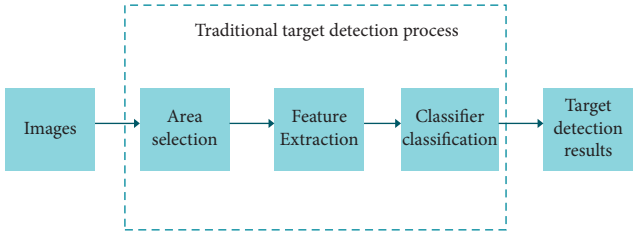

Figure 2: Traditional target detection flow chart.

2.1.2. Feature Extraction. The process of translating raw data into numerical features that can be processed while keeping the information in the original dataset is known as feature extraction. It produces better outcomes than applying machine learning to raw data directly. Many great feature operators have been extracted throughout years of study, such as Scale Invariant Feature Transform (SIFT), Histogram of Oriented Gradient (HOG), and others, in addition to typical features such as color, texture, shape, and gradient.

2.1.3. Classifiers. The technique of guessing the class of given data points is known as classification. Targets, labels, and categories are all terms used to describe classes. The job of estimating a mapping function (f) from input variables (X) to discrete output variables is known as classification predictive modeling (y). In machine learning, a classifier is an algorithm that automatically sorts or categorizes data into one or more "classes." Support Vector Machine (SVM), AdaBoost, and other classifiers are commonly utilized.

2.2. The Current State of Research in Deep Learning for Robotic Target Detection. At present, deep learning-based target detection algorithms fall into two main categories: one is the R-CNN family of target detection frameworks based on combining candidate regions (region proposal); the other is an algorithm that converts target detection into a regression problem.

The two-stage strategy is another name for the candidate region-based approach. This is because the whole network is split into two stages: first the extraction of candidate frames from the region of interest and then regression on the target's class and position. CNNs (convolutional neural networks) are an important part of candidate region-based algorithms. Since then, ResNet [18] and Inception v4 [19] have lowered target detection classification error to less than 4%. The two-stage based target detection method has demonstrated considerable gains in speed and accuracy for the job of picture classification using the new feature extraction model. Because of the completely connected layer of the CNN, the App-Net algorithm model developed by He et al. [20] overcomes the problem that the input pictures must be of the same size. Ross Girshick et al. [21] introduced the R-CNN algorithm concept in 2014. This algorithm is also the heart and soul of the R-CNN algorithm family. The R-CNN method calculates the candidate area input frame for the network model using the selective search [22] algorithm and utilises it as the input for the network model after normalisation. However, while collecting features from candidate frames in candidate areas, the R-CNN method performs a lot of duplicate calculations, slowing down the network model's overall detection performance. Ren et al. [23] introduced the Faster R-CNN target identification method based on the R-CNN, which is a network model that contains an RPN (Region Proposal Network), to address this problem. Instead of using the selective search technique, the RPN maps the candidate frames to the input picture, speeding up the detection process. Target detection has always followed the "region proposal + classification" concept, from R-CNN to Faster R-CNN [24], and this family of algorithmic models has effectively increased the detection accuracy and speed of convolutional neural networks for target detection.

Another method is the regression-based method, also known as the single-stage method. The regression-based detection method divides the feature map into S∗S grids and performs direct bounding box prediction in each grid, followed by category prediction and position regression. The YOLO algorithm [25] is the first algorithm to propose the idea of "single stage." The YOLO algorithm uses a regression-based approach. Unlike the two-stage algorithm model, the YOLO algorithm slices the image into S∗S grids and detects the possible target objects in the center of each grid. Each grid predicts two scales of bounding box information and the corresponding object class information. Instead of using a two-stage candidate region approach, the YOLO algorithm slices the image into a grid format and predicts the target object at the center of the grid at multiple scales. Compared with the candidate region-based target detection method, the YOLO algorithm significantly improves the detection speed of the network model while ensuring the accuracy and basically achieves the requirement of real-time detection. However, although the YOLO algorithm significantly improves the detection speed, the grid mechanism used in the YOLO algorithm is less effective in detecting small targets that fall in the center of the grid in complex scenarios with multiple targets. Moreover, compared to the two-stage target detection based on candidate regions, the localization of bounding boxes is poor, and the accuracy of localization is much lower than that of the Faster R-CNN algorithm model. Moreover, the detection effect is not as good as expected for objects with more regular shapes of the target objects. In response to the problems of the YOLO algorithm model, SSD algorithm model [26] is based on the YOLO algorithm, by combining the ideas of RPN algorithm [27], using the prediction on multiscale feature layers, and using the idea that different scales of the feature map feel different fields, respectively, on the high-level and low-level feature map prediction. The algorithmic model effectively improves the shortcomings of the YOLO algorithm, achieving detection accuracy (mAP) of 73.2% and a detection speed of 59 frames per second. However, SSD does not take small targets into account sufficiently, so detection of small targets is unsatisfactory, and region regression is difficult to converge when there are no candidate regions.

In conclusion, despite the rapid advancement of target detection algorithms, the problem of low detection accuracy

of tiny targets frequently encountered during the work of deep learning-based target detection techniques when applied to real-world settings such as robots requires more investigation.

2.3. Deep Learning Approach. Deep learning's success in picture classification and semantic segmentation [28–30] prompted scientists to apply it to RGB-D data processing. However, the techniques differ in terms of how depth data is sent into the network. The first method is to transmit the depth stream to the neural network as a fourth channel alongside the RGB. The benefit of this technique is that a lot of work has already been done on 2D RGB picture categorization. For this application, converting three-channel input to four channels is pretty simple. As a result, the challenge of how to encode depth data arose. The Horizontal Height Angle (HHA), for example, stores depth information in three channels [31]. The HHA is made up of data derived from depth horizontal disparity, height above ground, and the angle between the pixel's local surface normal and gravity's direction.

Deep neural networks are used in the work in [32] to extract features and categorize RGB-D photos. As a feature extractor, the suggested architecture employs a pretrained convolutional neural network (CNN). The network structure of [33] is as follows: five convolutional layers, three max-pooling layers to minimize the output dimensionality of the first, second, and fifth convolutional layers, two fully connected layers at the network's conclusion, and a softmax layer for classification. As an activation function, the Rectified Linear Unit (ReLU) is utilized. The network was trained for 1000 category classification tasks using the ImageNet dataset [34]. To adapt RGB-D pictures to CaffeNet, Schwarz et al. preprocessed RGB images by merging them with segmentation masks given with the dataset. This method converts RGB pictures to the fixed 227 by 227 dimensions that CaffeNet requires. The same approach is used to process depth pictures. The authors offer a unique approach for depth picture colorization that CaffeNet may use to transform it to a three-channel representation. The data from the RGB and depth photos are then merged and used to classify objects using a Support Vector Machine (SVM).

The work of [35] takes an approach similar to that of [32] and enhances performance only via the use of neural networks. The concept of starting with a pretrained model remains the same, but its design combines two models to do classification. Similarly, the network is divided into two streams, each of which is based on CNN that has been pretrained. The first channel is used to extract features from an RGB picture. The second extract is another collection of characteristics from the same data frame's depth data. To achieve classification, the two sets of features were combined and fed to a fully connected neural network. SafeNet is also used in both streams. Unlike earlier research, however, the authors fine-tune both streams on the RGB-D dataset [36]. The training process starts with the initialization of two streams using pretrained weights from CaffeNet, which was trained on the ImageNet dataset. On the RGB-D dataset, the second step is to train two streams separately. Combining the two streams and training the final classification layers represent the final phase.

2.4. Current State of Research in Small Target Detection. Although deep convolutional neural networks have made great progress in target detection, the detection of small targets still suffers from low detection accuracy. In response, some scholars have proposed new detection network models. In 2016, Bell et al. proposed an Inside-Outside Net (ION) detection model based on the inside and outside information of the region of interest [37]. In 2017, based on the Faster R-CNN network, Lin et al. [38] proposed a Feature Pyramid Network (FPN) with lateral connections, which utilises multiscale features and a top-down structure to achieve target detection. However, FPN only uses the top-level features for detection, ignoring the detailed information that is important for small target detection. To address the problems of the SSD algorithm for small target detection, Fu et al. [39] proposed a Deconvolutional Single Shot Detector (DSSD) algorithm, which changed the base network of the SSD algorithm to ResNet-101 [40], to enhance the feature extraction capability of the network. By combining multi-scale information, the detection accuracy of the model is improved. However, the above network ignores the connection between low-level features and high-level features and does not consider the perceptual field size of the convolutional operation, so that the convolutional operation with the same size of convolutional kernels for objects of different scales cannot extract the object information well.

3. Methodology

3.1. Target Recognition Algorithm for Robot Scenes Using a Deep Neural Network. Mobile robots must be able to properly navigate in complicated situations, detect and track items, avoid obstacles, establish their location, and rebuild 3D visual representations of their surroundings using cameras and other sensors. They may also be asked to provide humanitarian assistance and conduct industrial inspections. A service robot would often be expected to undertake search and rescue, give humanitarian aid in the care of the elderly via monitoring, and deliver timely information about their activities. Another possibility is the growing interest in unmanned aerial vehicles (UAVs) and vision-assisted driving. Traditionally, these functions have been provided by vision-based systems using stereovision or multiview coding. 14 tracking filters such as the Kalman filter, probabilistic data association filter, and multiple sensor fusion and state estimation are commonly used. Simultaneous localization and mapping (SLAM) provide a means of creating an environmental map identifying important obstacles, 3D surface reconstruction, and navigation and understanding the external world for indoor localization and navigational tasks with no external reference support like GPS or wireless location support. The authors of [41] described a real-time approach for reconstructing 3D surfaces from a set of known perspectives using an event-based

camera. Although visual-inertial/odometry (VI/VIO) relies on cameras and inertial measuring units (IMUs) to assess a robot's state (position, orientation, and velocity), it may also be used for other tasks including control, obstacle detection, and avoidance, as well as path planning. References [42, 43] are excellent resources for a thorough examination of SLAM.

3.2. A Small Target Detection Method Based on a Jump Connection Pyramid Model. Because of the camera distance and angle, intelligent mobile robots are in the process of moving, resulting in a large number of small target items that must be identified while moving. For example, the inspection robot must gather information on failure locations, instruments, and equipment, among other things, throughout the inspection process, which necessitates the exact recognition of tiny targets in the picture through implementation. As a result, improving the identification of tiny targets by intelligent service robots in environmental awareness is critical. Target detection based on convolutional neural networks has reached great detection accuracy and detection speed thanks to the widespread usage of deep convolutional neural networks for target detection. Small target identification, on the other hand, remains a hurdle. Small targets are frequently neglected during feature extraction, since they make up such a small amount of the image. Furthermore, conventional deep convolutional neural network target identification methods in the network prediction phase do not fully exploit the link between the semantic information of higher-level features and the detailed information of lower-level features.

We present a tiny target identification approach based on the jump-connected pyramid model to overcome these issues. The majority of the innovation is in two areas: To begin, a jump-connected pyramid model is presented as a way to combine the semantic information of high-level characteristics with the detailed information of low-level features in the network. Different steps of deconvolution are used to downscale the disconnected high-level features to the same size as that of the low-level features, and a 1×1 convolution layer is employed to minimize the dimensionality of the high-level features. Finally, the fused features are subjected to classification and position regression. Second, three parallel transversal network topologies are employed in the network model to better extract feature information corresponding to varied sizes of objects.

3.3. Convolutional Neural Network Fundamentals. The technique in this paper's deep network model is mostly based on convolutional neural networks. As a result, before we go over the overall structure of the proposed network model, as well as the detailed structure of each part and the training of the model, we will go over the principles and composition of convolutional neural networks, forward and backward propagation of convolutional neural networks, and the basic loss function principles. Deep neural networks are built on the foundation of convolutional neural networks (CNNs), which are widely utilized in computer vision, speech recognition, natural language processing, and bioinformatics. Convolutional neural networks, in particular, for computer vision, combine convolution and pooling to efficiently minimize the number of weight parameters, even with the direct input of multidimensional pictures, without requiring a large amount of processing effort. Furthermore, several types of transformation, such as translation, angle, and scale transformation scaling, are extremely invariant to convolutional neural networks. As a result, the target detection technique proposed in this study uses a convolutional neural network structure.

Convolutional neural networks work by mimicking the way humans process information, combining underlying features of an image through multiple layers of convolutional pooling to form higher-level features that represent more abstract information such as categories. Convolutional neural networks consist of a convolutional layer, a pooling layer, and a fully connected layer. The network structure consists of a convolutional layer, a pooling layer, and a fully connected layer. In detail, a convolutional neural network generally starts with alternating convolutional and pooling layers, with the last few layers near the output layer being the fully connected layers, as shown in Figure 3.

Because the layers of the convolutional neural network are completely linked, there are a lot of training parameters, which restricts the depth and complexity of the network model. The information links in each space of the picture, on the other hand, are confined. To obtain global information about an image, it is not necessary to have information about the perceptual field of the entire image but only about a portion of it, and then the global information can be obtained by combining the information of all the local perceptual fields, reducing the number of training parameters even further.

3.4. General Framework of the Model. To solve the current problem of low accuracy of small target detection based on deep learning, we propose a small target detection method based on the jump-connected pyramid model, whose overall structure is shown in Figure 4.

Our model uses VGG16 as the feature extraction network and adds a global receptive field (GRF) module to the feature extraction structure to extract the global feature information of the network. In the prediction stage of the network model, a Skip Feature Pyramid Network is used to fuse the higher-level semantic feature information with the lower-level feature detail information, and nonmaximum suppression is used to obtain the final prediction results. As shown in the figure, the input of the network model is a fixed size color image, and convolution 4_3, convolution 5_3, convolution 6_2, and fully connected layer 7 are the convolutional layers of the VGG16 based network at different scales.

3.5. Jump Connection Pyramid Module. Figure 5 depicts the proposed jump pyramid paradigm in this work. Figure 5(a) depicts the YOLO algorithm's model structure, which predicts using just the final layer of feature layers, with the benefit of increased detection speed but the downside of

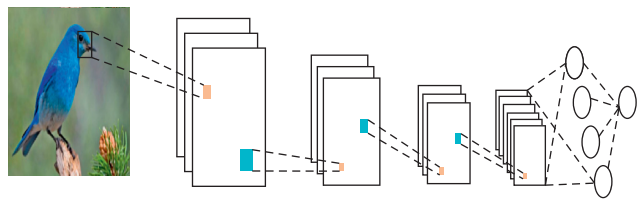

Figure 3: CNN feature structure.

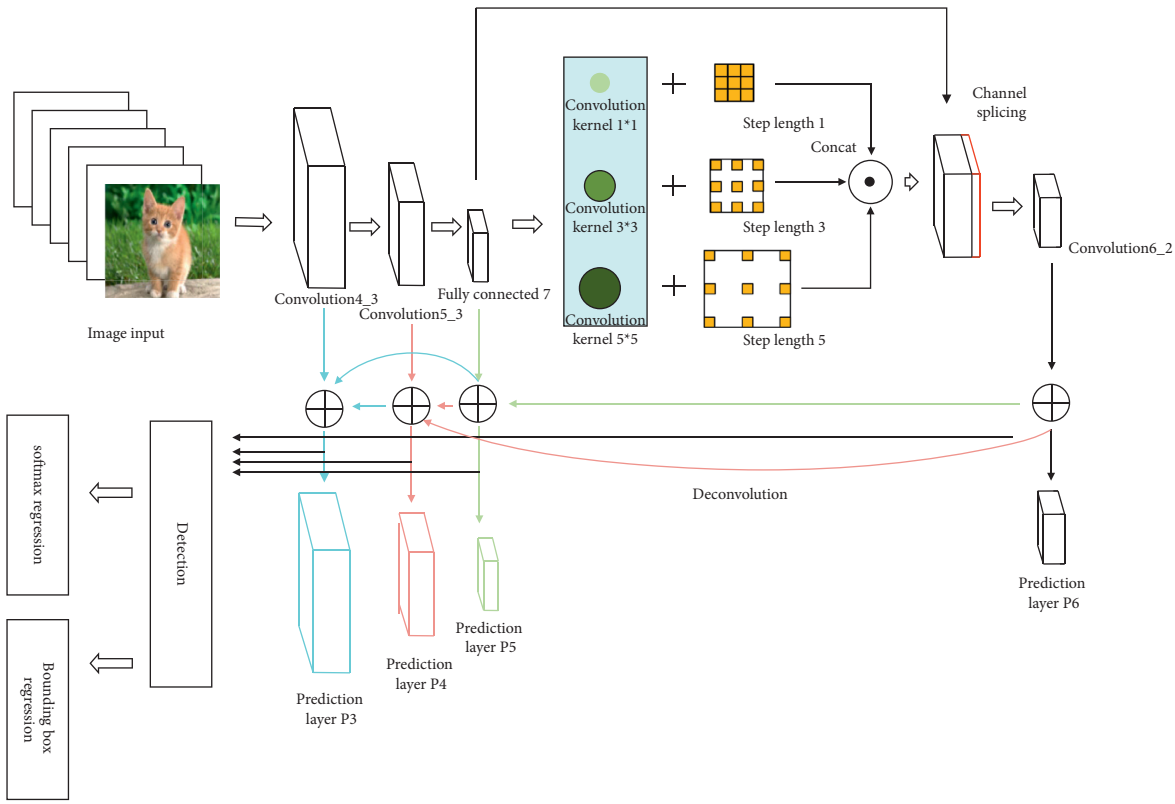

Figure 4: Block diagram of a small target detection model based on a jump connection pyramid.

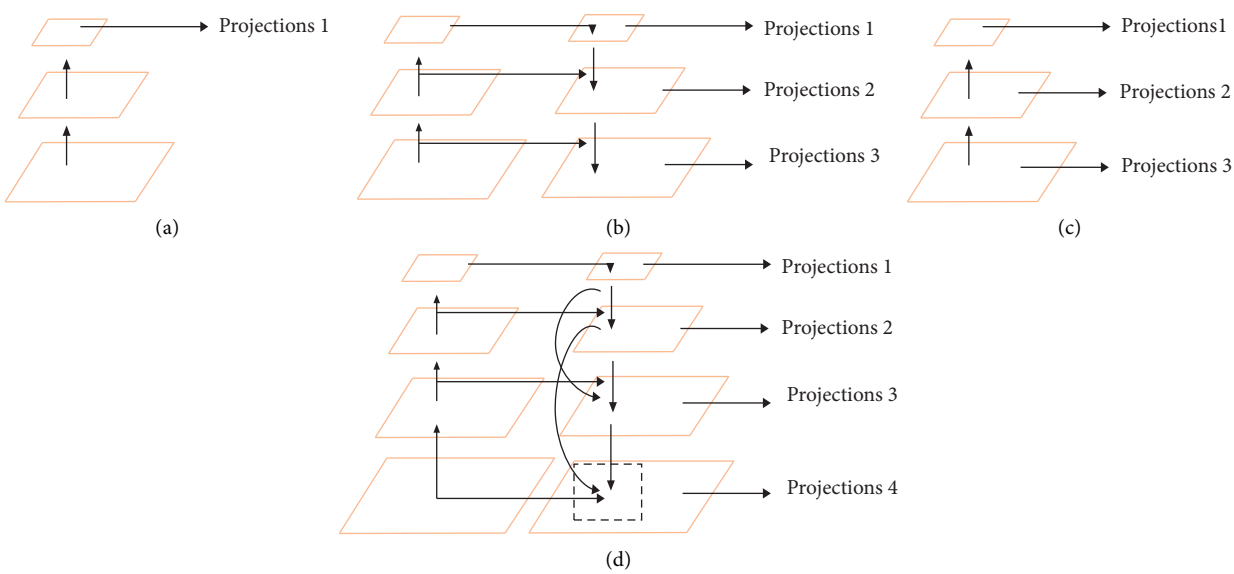

Figure 5: Different structural forms of predictive networks.

poorer detection accuracy. Figure 5(b) is an enhancement to the single-layer feature layer prediction model in Figure 5(a), which predicts higher-level feature layers and hence improves detection outcomes. A top-down pyramid model with distinct feature layers is shown in Figure 5(c). This algorithmic approach takes into account information from nearby feature maps but overlooks the link between higher-level semantic information and lower-level specific feature information. Figure 5(d) shows a jump-connected pyramid model that employs different levels of deconvolution for upsampling and a pixel-by-pixel summation approach to fuse the information between nonadjacent feature maps to address these issues.

In deep convolutional neural networks, the deeper the network model is the more abstract feature information is contained in the image feature layers. For the detection of small targets, the detailed features contained in the lower layer feature maps are equally important for target prediction. Therefore, a jump-connected pyramid model is proposed to fuse the information from the upper and lower feature maps. Furthermore, this allows the use of scale information from different feature layers.

The specific fusion method is shown in Figure 6. First, the high-level feature maps in the selected feature extraction network are passed through a 256-channel 3×3 convolution kernel, changing the number of channels in the different feature layers to the same to facilitate subsequent fusion calculations. After obtaining the same number of feature layers, the adjacent feature maps are upsampled using a 2×2 deconvolution operation with a step size of 2. Nonadjacent feature maps are also upsampled using a 4×4 deconvolution step. The specific deconvolution operation is shown in the following equation:

$$o = \left[\frac{i - f + 2p}{s}\right] + 1, \tag{1}$$

where i is the size of the input feature map; f is the size of the convolution kernel; s is the step size of the deconvolution; and p is the number of pixels filled. The resulting feature map is used as the fused feature map by summing over each pixel.

3.6. Global Feel Wild Module. The top-down network structure is commonly used in current models based on deep convolutional neural networks. This structure neglects the different perceptual fields for different sizes of objects. To address the above problem, a parallel aggregation structure is proposed to enhance the global feature extraction of the overall model by using different step sizes of null convolution and different sizes of convolution kernels. The structure of the global perceptual field module is shown in Figure 7, which can effectively extract features from objects of different scales and sizes.

In the network model, we use the structure of null convolution to improve the perceptual field of the convolutional neural network. Assuming an input feature map of x, a filter of w, and a sampling step of r, for each coordinate i of the output feature map y, the expression for the null convolution is as follows:

$$y[\mathrm{i}] = \sum_k x[i + r \cdot k] w[k]. \tag{2}$$

For a null convolution with a sampling step of r and a convolution kernel size of $k \times k$, the perceptual field size is calculated as follows:

$$k_e = k + (k - 1)(r - 1). \tag{3}$$

The change in perceptual field size can be seen as $k \times k \longrightarrow k_e \times k_e$.

The global perceptual field module's detailed structure is as follows. To begin, by altering the number of channels in the feature map, a 1×1 convolutional layer is utilized to lower the computational burden of the feature model. The visual feature information is then retrieved at various scales utilizing convolution kernels of 1, 3, and 5 sizes, as well as cavity convolution of 1, 3, and 5 sizes. The feature maps are stitched together in channels after that. Using a 1×1 convolution kernel, the number of channels is altered to the same as that of the original feature map, and the corresponding pixels of the original feature map are overlaid. This not only enhances the feature extraction of tiny target objects but also increases the information in the global perceptual field.

3.7. Loss Functions. To balance the problem of large differences in the number of positive and negative samples in the dataset, this section uses negative sample mining to solve the problem of imbalance in the extreme foreground-background categories. In the training process of the network model, instead of using all negative sample bounding boxes and randomly selecting negative sample bounding boxes, the loss of negative samples is ranked and the ratio of positive to negative samples is 3 : 1. Before the final prediction, the bounding boxes generated by the network prediction are bifurcated to filter out the foreground and background. This reduces the number of negative samples. The loss function of the network is shown in the following equation:

$$\begin{aligned} L(\{p_i\}, \{x_i\}, \{c_i\}, \{t_i\}) = {} & \frac{1}{N_{\mathrm{conv}}} \left(\sum_i l_b(p_i, [l_i^* \geq 1]) + \sum_i [l_i^* \geq 1] \cdot l_r(x_i, g_i^*) \right) \\ & + \frac{1}{N_p} \left(\sum_i l_m(c_i, l_i^*) + \sum_i [l_i^* \geq 1] \cdot l_r(t_i, g_i^*) \right), \end{aligned} \tag{4}$$

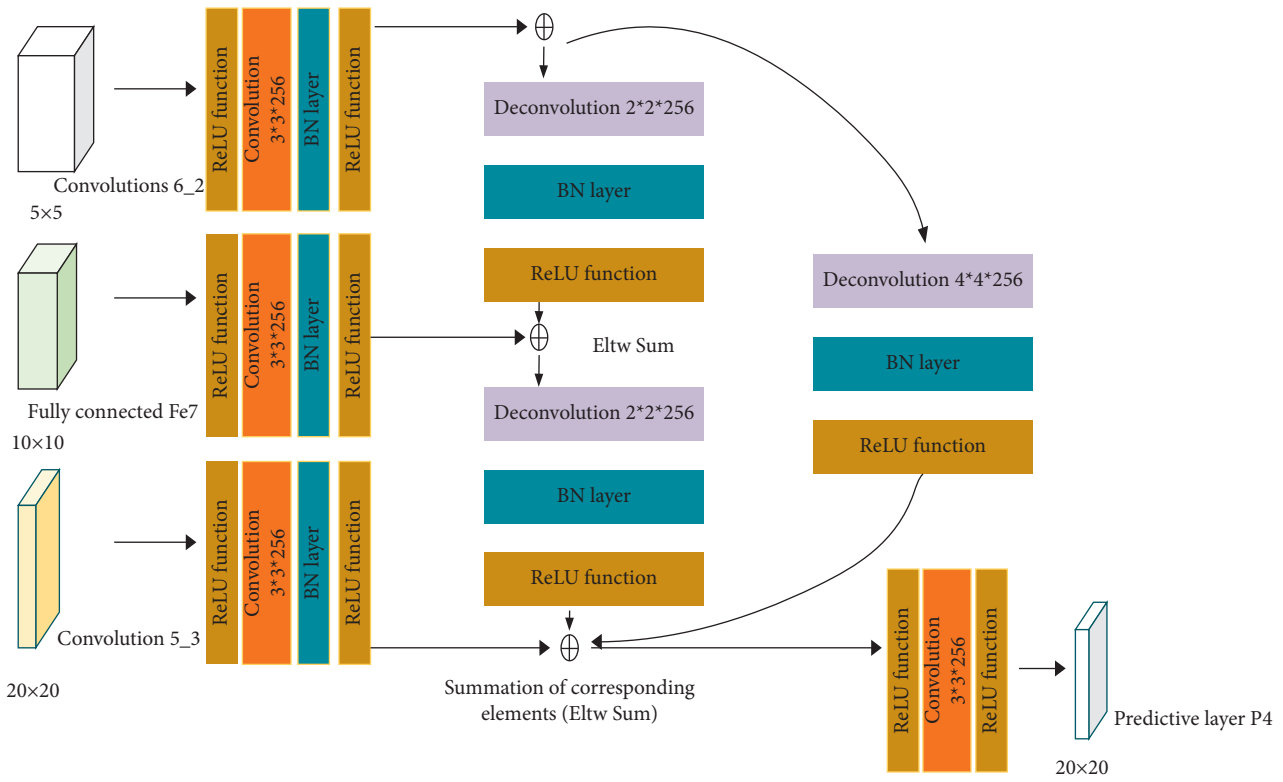

Figure 6: The detailed structure of the jump-connected pyramid.

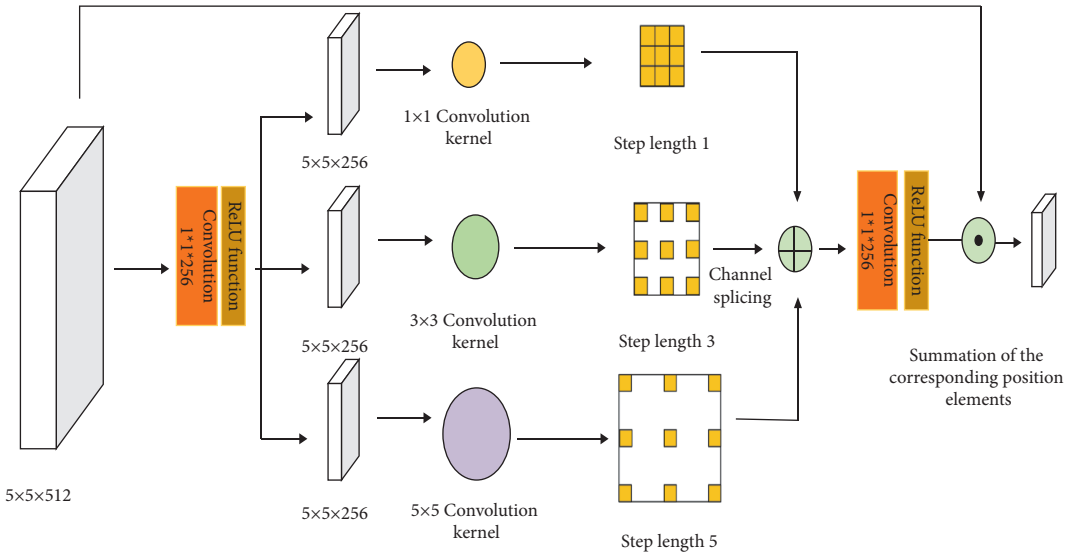

Figure 7: Global Feel Wild module structure.

where i is the index of the bounding box in each training batch of the training set; l_i^* is the corresponding category label of each image annotation in each batch of images; g_i^* is the coordinate label information corresponding to each image annotation; p_i and x_i denote the presence and absence of the target object and the corresponding coordinate information in the bounding box predicted by the network; c_i and t_i are the classes of the objects in the predicted target bounding box and the corresponding coordinate information; N_{conv} and N_p are the numbers of positive sample enclosing boxes in the feature extraction network and the prediction network, respectively; l_b is the cross-entropy loss of the binary classification of the output of the feature extraction network, that is, the determination of whether there is a target in the enclosing box; l_m is the confidence level for the multiclassification task. Similar to the Fast R-CNN algorithm, l_r is the smoothed L1 regression loss. The corresponding loss of the model is only calculated if

the prediction is true when $l_i^* \geq 1$ in the enclosing box. The specific loss function for one of the position loss functions l_r is as follows:

$$l_r(x, g^*, l^*) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} [l^* \geq 1] \text{smooth}_{L1}\left(x_i^m - \widehat{g}_j^m\right),$$

$$\widehat{g}_j^{cx} = \frac{\left(g_j^{cx} - d_i^{cx}\right)}{d_i^w},$$

$$\widehat{g}_j^{cy} = \frac{\left(g_j^{cy} - d_i^{cy}\right)}{d_i^h},$$

$$\widehat{g}_j^{w} = \log\left(\frac{g_j^w}{d_i^w}\right),$$

$$\widehat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right), \tag{5}$$

where (cx, cy, w, h) are the predicted center coordinates of the bounding box and the corresponding width and height, respectively. $(\widehat{g}^{cx}, \widehat{g}^{cy}, \widehat{g}^{w}, \widehat{g}^{h})$ denote the center coordinates of the image annotated coordinates of the center of the bounding box and the corresponding width and height, $(d^{cx}, d^{cy}, d^{w}, d^{h})$ denote the center coordinates of the default bounding box as well as the width and height, and $(x^{cx}, x^{cy}, x^{w}, x^{h})$ denote the center coordinates of the predicted bounding box as well as the width and height.

4. Experiments and Analysis of Results

The algorithmic model in this research was pretrained on the ILSVRCCLS-LOC dataset and employed the VGG16 network as the basis feature extraction network. The method in this part was tested on the PASCAL VOC and MS COCO datasets, respectively, to ensure that the algorithm model was effective. The PASCAL VOC and MS COCO databases provide 20 and 80 categories, respectively, with each category having its own category information and location label information.

4.1. Experimental Analysis of the PASCAL VOC Dataset. The network model was trained on the PASCAL VOC2007 and PASCAL VOC2012 datasets and tested on the PASCAL VOC2007 dataset. Figure 8 shows some example images of the PASCAL VOC dataset. The algorithm model presented in this section was trained for a total of 140K iterations. The learning rate was set to 10^{-3} for the first 80K iterations, decreasing to 10^{-4} for 80K to 100K iterations, 10^{-5} for 100K to 120K iterations, and 10^{-6} for 120K to 140K iterations.

As shown in Table 1, the experimental results of the algorithm model in this section on the PASCAL VOC2007 dataset are compared with those of the mainstream methods. As can be seen from the table, the final detection results vary depending on the size of the input images. The algorithm model obtained the detection accuracy on the PASCAL VOC2007 dataset by calculating the average accuracy of the test set. When the input image size is 320×320, the accuracy of the algorithm model in this section is 80.1%, with a speed of 31.2 frames per second. When the input image is 512×52, the average accuracy of the detection is 81.9% and the speed is 18.2 frames per second. Our algorithmic model is 1% more accurate than the STDN algorithm with the highest accuracy, but the speed of detection is 10.4 frames per second lower. The accuracy of the algorithm model and the speed of detection of the algorithm model are highly dependent on the size of the input image. When the size of the input image is large, the number of corresponding pixels in the image will increase, and the corresponding targets in the image will occupy more pixels, which will increase the computational consumption of the feature extraction phase of the network model and affect the detection efficiency of the network model. The algorithm model in this section improves the detection accuracy of the network model with less reduction in rate and also validates the effectiveness of our proposed algorithm model in improving the accuracy of small targets.

Table 2 shows the results of this part for the PASCAL VOC2007 test set. The average accuracy rates for each of the 20 categories are shown in the table. This method's detection rate for tiny targets is substantially higher than that of other standard detection methods, as seen in the table. For all categories, the average accuracy is 1% greater than the best algorithm. Small objects such as birds, sheep, and plants had greater accuracy than other networks by 2.5 percent, 3.2 percent, and 2.7 percent, respectively. The experimental findings in this study demonstrate the algorithm's usefulness.

4.2. Design of a Target Detection System for Intelligent Mobile Robot Scenarios. Intelligent mobile robots perform the task of target detection through three main processes: sensing, decision-making, and control. The intelligent mobile robot uses sensors on its body to obtain information about the external environment, which is then processed and transmitted to the robot's decision-making system, where the decision-making process makes the appropriate decision and the final decision command controls the mobile robot to complete the corresponding task.

In order to better validate the effectiveness of the proposed scene target detection algorithm for mobile robot environment awareness, we have developed and designed a prototype software system for realistic scene detection using C++ MFC and Python technology. The software is organized as follows: firstly, a requirements analysis is carried out to determine the specific functions to be implemented; then, a prototype target detection system based on the C++ MFC software environment is designed and implemented. Compared to MATLAB and Java, the MFC development environment is more efficient and allows for simpler and more user-friendly programming of the window interface while still meeting the design requirements. The flow chart of the system is shown in Figure 9.

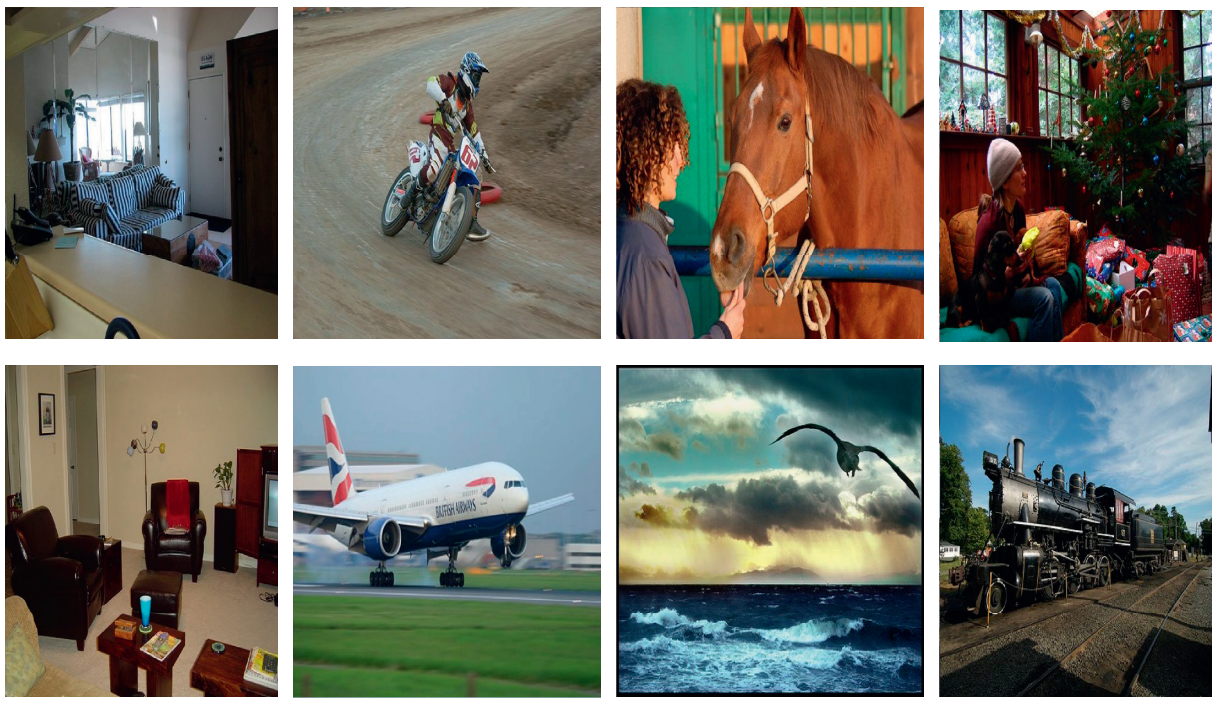

Figure 8: Example image of a PASCAL VOC dataset.

Table 1: Detection results of different network models for PASCAL VOC2007.

Methods	Basic network	Accuracy	Detection speed	Image size
Faster	VGG16	73.2	7	1000 × 600
Faster	Residual-101 [10]	76.4	2.4	1000 × 600
R-FCN	Residual-101	80.5	9	1000 × 600
DSOD300	DS/64-192-4	77.7	17.4	300 × 300
YOLOv2	Darknet-19	78.6	40	544 × 544
SSD300	VGG16	77.5	46	300 × 300
DSSD32	Residual-101	79.5	9.5	321 × 321
STDN321	DenseNet-169	79.2	41.5	321 × 321
Ours320	**VGG16**	**80.1**	**31.2**	**320 × 320**
SSD512	VGG16	78.6	19	512 × 512
DSSD513	Residual-101	81.5	5.5	513 × 513
STDN513	DenseNet-169	80.9	28.6	513 × 513
Ours512	**VGG16**	**81.9**	**18.2**	**512 × 512**

Bold values represent the experimental results of our method.

4.3. Recognition of the Target. We use the RTX 2080ti as the equipment, TensorFlow as the experimental framework, and the COCO database to test the improved YOLO approach. Because the experimental scenario is an office, the COCO dataset was used for a thorough examination. If you need to add a specific target for identification, just upload the relevant information and change the network structure as needed. Figure 8 displays the implementation's final result. There are 90 categories in the dataset, with many small targets, many single-picture targets, and noncentral distribution for the bulk of the items. It is better suitable for daily usage and more difficult to detect. Despite the poor quality of the experimental apparatus, as illustrated in Figure 8, good experimental results are obtained. The algorithm can analyze 416 × 416 images at 29 frames per second with up to 55.3 percent mAP@0.5, which is similar to RetinaNet but four times faster.

4.4. Experimental Analysis of the MS COCO Dataset. The MS COCO dataset tests were undertaken to further evaluate the efficiency of the algorithmic model presented in this section. The MS COCO dataset offers more categories and training pictures than the PASCAL VOC dataset, and it contains data from a variety of complicated scenarios. Table 3 shows the detection results for the MS COCO dataset. The findings for the underlying feature network and several picture sizes are shown. For an input picture

Table 2: PASCAL VOC2007 test results for different categories.

Methods category	Faster	ION	MR-CNN	YOLOv2	SSD300	SS512	STDN321	STDN513	Ours320	Ours 512
Aero	76.5	79.2	80.3	86.3	79.5	84.8	81.2	86.1	84.5	**88.5**
Bike	79	83.1	84.1	82	83.9	85.1	88.3	**89.3**	85.4	86.4
Bird	70.9	77.6	78.5	74.8	76	81.5	78.1	79.5	80.1	**84**
Boat	66.5	65.6	70.8	59.2	69.6	73	72.2	74.3	73.8	**75.8**
Bottle	52.1	54.9	68.5	51.8	50.5	57.8	54.3	61.9	60	**69.4**
Bus	83.1	85.4	88	79.8	87	87.8	87.6	88.5	87.7	**88.9**
Car	84.7	85.1	85.9	76.5	85.7	88.3	86.7	88.3	88.2	**89.2**
Cat	86.4	87	87.8	**90.6**	88.1	87.4	88.7	89.4	89	89.5
Chair	52	54.4	60.3	52.1	60.3	63.5	63.5	**67.4**	63.8	66.7
Cow	81.9	80.6	85.2	78.2	81.5	85.4	83.2	85.5	84.7	**86.4**
Table	65.7	73.8	73.7	58.5	77	73.2	79.4	**79.5**	77.2	73.2
Dog	84.8	85.3	87.2	**89.3**	86.1	86.2	86.1	86.4	86	87.6
Horse	84.6	82.2	86.5	82.5	87.5	86.7	**89.3**	89.2	86.4	88.2
M. bike	77.5	82.2	85	83.4	83.9	83.9	88	**88.5**	86.7	87.5
Person	76.7	74.4	76.4	81.3	79.4	82.5	77.3	79.3	82.5	**84.9**
Plant	38.8	47.1	48.5	49.1	52.3	55.6	52.5	53	56.1	**58.3**
Sheep	73.6	75.8	76.3	77.2	77.9	81.7	80.3	77.9	81.3	**84.9**
Sofa	73.9	72.7	75.5	62.4	79.5	79	80.8	**81.4**	80.4	78.3
Train	83	84.2	85	83.4	87.6	86.6	86.3	86.6	88.5	**87.8**
Tv	72.6	80.4	81	68.7	76.8	80	82.1	**85.5**	79.8	80.8
mAP	73.2	75.6	78.2	76.8	77.5	79.5	79.3	80.9	80.1	**81.9**

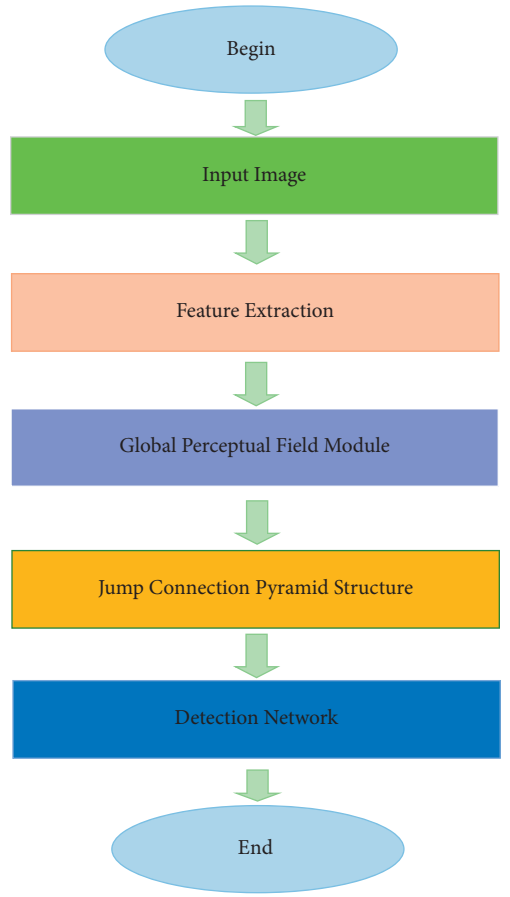

Figure 9: Flow chart of small target detection for robot scenes.

Table 3: MS COCO dataset test results.

Methods	Basic network	AP0.5 : 0.95	$AP_{0.5}$	AP0.75	APS	APM	APL	AR1	AR10	AR100	ARs	AR_M	AR_L
Faster	VGG16	21.9	42.7	—	—	—	—	—	—	—	—	—	—
ION	VGG16	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.2	10.1	37.7	53.6
R-FCN	Residual-101	29.2	51.5	—	10.3	32.4	43.3	—	—	—	—	—	—
DSOD	DS/64/192/4	29.3	47.3	30.6	9.4	31.5	47	27.3	40.7	43	16.7	47.1	65
YOLOv2	Darknet	21.6	44	19.2	9	28.9	41.9	24.8	37.5	39.8	14	43.5	59
SSD300	VGG16	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
DSSD321	Residual-101	28	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42	62.6
STDN321	DenseNet	28	45.6	29.4	7.9	29.7	45.1	24.4	36.1	38.4	12.5	42.7	60.1
Ours320	**VGG16**	**28.2**	**47.7**	**29.1**	**10.3**	**31.4**	**43.7**	**25.8**	**38.9**	**41.2**	**16.9**	**47.2**	**61**
SSD512	VGG16	28.8	48.5	30.3	10.9	31.8	43.5	26.1	39.5	42	16.5	46.6	60.8
DSSD513	Residual-101	33.2	53.3	35.2	13	35.4	51.1	28.9	43.5	46.2	21.8	49.1	66.4
STDN513	DenseNet	31.8	51	33.6	14.4	36.1	43.4	27	40.1	41.9	18.3	48.3	57.3
Ours512	**VGG16**	**33.1**	**52.3**	**32.4**	**15.6**	**34.6**	**42.7**	**28.3**	**42.6**	**45.6**	**25.9**	**50.8**	**60.1**

Bold values represent the experimental results of our method.

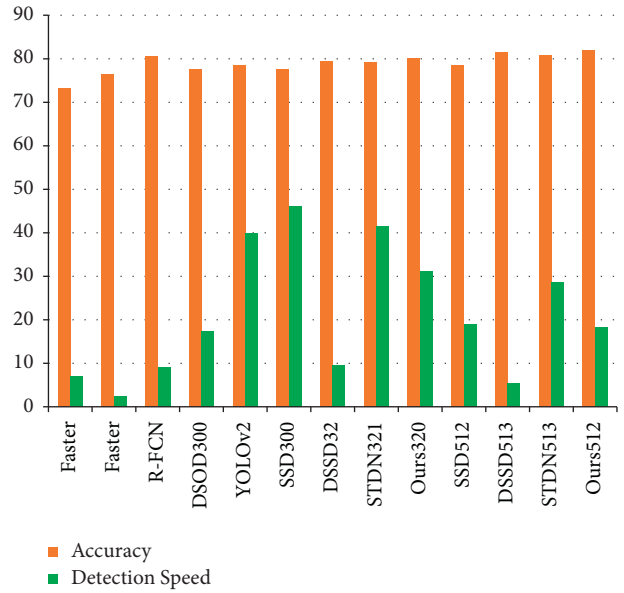

Figure 10: Detection results for PASCAL VOC2007.

size of 320 × 320 and an evaluation metric of AP0.5 : 0.95, the detection method in this section has a 28.2% accuracy. The detecting algorithm's accuracy is 33.1% when the input picture size is 512 × 512 pixels. The detection algorithm in this section has a recall and accuracy of 2.4 percent and 1.2 percent, respectively, greater than the STDN algorithm model, where APS denotes the detection accuracy of tiny targets. The MS COCO dataset has more categories and more sophisticated visual information than the PASCAL VOC dataset. As a result, the detection accuracy is quite poor on average, and there are disparities in the detection results based on different criteria. The maximum detection accuracy was likewise attained for the metrics examined for the detection of tiny objects. As a consequence, the experimental findings support the efficacy of the method presented in this study.

Figure 10 compares the accuracy and detection of all of the approaches we tested for PASCAL VOC2007. The accuracies of Ours320 and Ours512 are 80.1% and 81.9%, respectively, according to this graph, which are the greatest accuracies of all approaches. Similarly, these two approaches had the greatest detected speed among the chosen methods, with 31.2 and 18.2 seconds for Ours320 and Ours512, respectively.

Figure 11 shows the comparison among our selected 10 methods, that is, Faster, ION, MR-CNN, YOLOv2, SSD300, SSD512, STDN321, STDN513, Ours320, and Ours512, respectively, against method categories. From this figure, it is clear that Ours512 is better than other methods that we have selected.

Figure 12 depicts a comparison of the results obtained from the MS COCO dataset test. It is obvious from this figure that Ours320 and Ours512 are superior to the other approaches we considered.

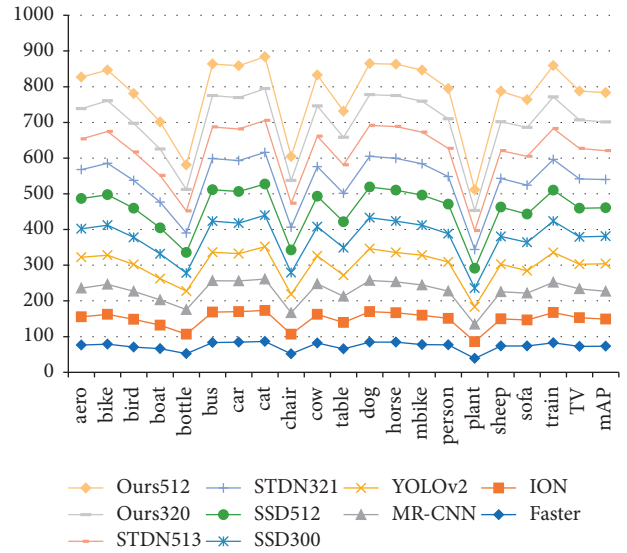

Figure 11: Test results for different categories.

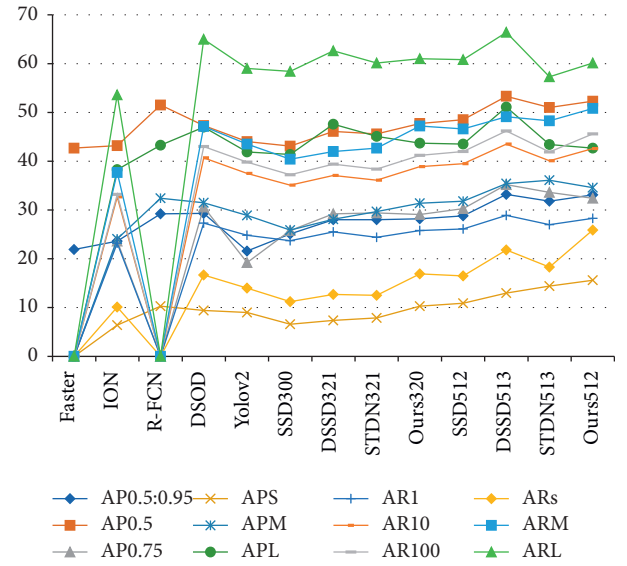

Figure 12: MS COCO dataset test results.

5. Conclusion

Target detection and placement based on deep learning algorithms has been a popular issue in the field of pattern recognition in recent years. People not only want to be free of easy and repetitive tasks, but they also want robotic intelligence to be able to satisfy the needs of humans autonomously, lessening the load on families and society and enabling a variety of intelligent services. We examine challenges in the application of target identification algorithms for mobile robots in this research, with the goal of improving intelligent mobile robots' perception of external environmental information and making intelligent inspection robots more suited to actual surroundings. A jump-connected pyramid model is used to suggest a target detection approach. The high-level feature map semantic information in a deep learning-based target algorithm model is extremely abstract for the target's features, but the low-level feature map information includes comprehensive information. To merge many layers of high-level semantic feature information with the detailed information of low-level feature maps, a jump-connected pyramid structure is proposed. Furthermore, the global feature information is recovered utilizing different sizes of convolution kernels and varied step sizes of complete convolution in the network model to better extract feature information related to objects at different scales. Experiments were carried out on numerous different datasets to validate the algorithm's performance, and the findings verified the algorithm's effectiveness. Furthermore, these findings suggest that the proposed model performs much better than previous algorithm models in terms of tiny target identification accuracy.

Data Availability

The datasets used in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The author declares that he has no conflicts of interest.

References

[1] R. Sarc, A. Curtis, L. Kandlbauer, K. Khodier, K. E. Lorber, and R. Pomberger, "Digitalisation and intelligent robotics in value chain of circular economy oriented waste management - a review," *Waste Management*, vol. 95, pp. 476–492, 2019.

[2] G. Ren, T. Lin, Y. Ying, G. Chowdhary, and K. C. Ting, "Agricultural robotics research applicable to poultry production: a review," *Computers and Electronics in Agriculture*, vol. 169, Article ID 105216, 2020.

[3] H. Hassani, E. S. Silva, S. Unger, M. TajMazinani, and S. Mac Feely, "Artificial intelligence (AI) or intelligence augmentation (IA): what is the future?" *A&I*, vol. 1, no. 2, pp. 143–155, 2020.

[4] Y. Tang, M. Chen, C. Wang et al., "Recognition and localization methods for vision-based fruit picking robots: a review," *Frontiers of Plant Science*, vol. 11, p. 510, 2020.

[5] E. E. Joh, "Private security robots, artificial intelligence, and deadly force," *U.C. Davis L. Review*, vol. 51, p. 569, 2017.

[6] C. Kahraman, M. Deveci, E. Boltürk, and S. Türk, "Fuzzy controlled humanoid robots: a literature review," *Robotics and Autonomous Systems*, vol. 134, Article ID 103643, 2020.

[7] A. Rogowski and P. Skrobek, "Object identification for task-oriented communication with industrial robots," *Sensors*, vol. 20, no. 6, p. 1773, 2020.

[8] L. Liu, M. Pietikäinen, J. Qin, O. Wanli, and V. G. Luc, "Efficient visual recognition," *International Journal of Computer Vision*, vol. 128, no. 8, pp. 1997–2001, 2020.

[9] Y. Xue and Y. Li, "A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 8, pp. 638–654, 2018.

[10] S. Moradi, P. Moallem, and M. F. Sabahi, "Fast and robust small infrared target detection using absolute directional mean difference algorithm," *Signal Processing*, vol. 177, Article ID 107727, 2020.

[11] H. Zhu, S. Liu, L. Deng, Y. Li, and F. Xiao, "Infrared small target detection via low-rank tensor completion with top-hat

regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 2, pp. 1004–1016, 2019.

[12] Y. Tian, L. Ping, and X. Wang, "Pedestrian detection aided by deep learning semantic tasks," 2015, https://arxiv.org/abs/1412.0069.

[13] G. Levi and T. Hassner, "Age and gender classification using convolution neural networks," in *Proceedings of the .2015 IEEE Conference on Computer. Vision and Pattern Recognition Workshops*, pp. 34–42, CVPRW, Boston, MA, USA, 2015.

[14] Y. Zhou, Q. L Liu, L. Shao, and M. Mellor, "DAVE: a unified framework for fast vehicle detection and annotation," 2016, https://arxiv.org/abs/1607.04564.

[15] L. T. Nguyen-Meidine, E. Granger, M. Kiran, and L. A. Blais-Morin, "A comparison of CNN-based face and head detectors for real-time video surveillance applications," 2018, https://arxiv.org/abs/1809.03336.

[16] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," 2016, https://arxiv.org/abs/1611.07759.

[17] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: a survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84–100, Jan. 2018.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[19] C. Szegedy, S. Ioffe, V. Vincent, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4278–4284, AAAI Press, February 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, pp. 1904–1916, 2014.

[21] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with r∗ cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1080–1088, Santiago, Chile, December 2015.

[22] P. Bharati and A. Pramanik, "Deep learning techniques-R-CNN to mask R-CNN: a survey," *Computational Intelligence in Pattern Recognition*, Springer, Berlin, Germany, pp. 657–668, 2020.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[24] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: an improved faster RCNN approach," *Neurocomputing*, vol. 299, pp. 42–50, 2018.

[25] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, 2019.

[26] W. Liu, D. Anguelov, D. Erhan et al., "SSD: Single Shot Multibox detector," *European Conference on Computer Vision*, Springer, Berlin, Germany, pp. 21–37, 2016.

[27] Y. Xiao, X. Wang, P. Zhang, F. Meng, and F. Shao, "Object detection based on faster R-CNN algorithm with skip pooling and fusion of contextual information," *Sensors*, vol. 20, no. 19, p. 5490, 2020.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, Curran Associates, Inc., Dutchess County, NY, USA, 2012.

[31] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proceedings of the European Conference on Computer Vision*, pp. 345–360, Springer, Berlin, Germany, July 2014.

[32] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1329–1335, Seattle, WA, USA, May 2015.

[33] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 675–678, Orlando, FL, USA, November 2014.

[34] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[35] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 681–687, Hamburg, Germany, September2015.

[36] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 1817–1824, Shanghai, China, May 2011.

[37] S. Bell, C. L. Zitnick, K. Bala, and G. Ross, "Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2874–2883, Las Vegas, NV, USA, June 2016.

[38] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[39] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. Berg, "Dssd: Deconvolutional single shot detector," 2017, https://arXiv:1701.06659.

[40] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: revisiting the ResNet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[41] H. Rebecq, G. Gallego, and E. Mueggler, "EMVS: event-based multi-view stereo-3d reconstruction with an event camera in real-time," *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1395–1414, 2018.

[42] G. N. Desouza and A. C. Kak, "Vision for mobile robot navigation: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[43] C. Cadena, L. Carlone, H. Carrillo et al., "Past, present, and future of simultaneous localization and mapping: toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.