Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 9076857, 13 pages
https://doi.org/10.1155/2020/9076857

Research Article

Research on Object Detection Algorithm Based on Multilayer Information Fusion

Bao-Yuan Chen,[1,2,3] Yu-Kun Shen,[1,2,3] and Kun Sun[1,2,3]

[1]*The Higher Educational Key Laboratory for Measuring & Control Technology and Instrumentation of Heilongjiang Province, Harbin University of Science and Technology, Harbin 150080, China*
[2]*National Experimental Teaching Demonstration Center for Measurement and Control Technology and Instrumentation, Harbin University of Science and Technology, Harbin, China*
[3]*School of Measurement–Control Technology and Communications Engineering, Harbin University of Science and Technology, Harbin, China*

Correspondence should be addressed to Kun Sun; sunkun1982@126.com

Received 6 April 2020; Accepted 15 September 2020; Published 27 September 2020

Academic Editor: Suzanne M. Shontz

At present, object detectors based on convolution neural networks generally rely on the last layer of features extracted by the feature extraction network. In the process of continuous convolution and pooling of deep features, the position information cannot be completely transferred backward. This paper proposes a multiscale feature reuse detection model, which includes the basic feature extraction network DenseNet, feature fusion network, multiscale anchor region proposal network, and classification and regression network. The fusion of high-dimensional features and low-dimensional features not only strengthens the model's sensitivity to objects of different sizes but also strengthens the transmission of information, so that the feature map has rich deep semantic information and shallow location information at the same time, which significantly improves the robustness and detection accuracy of the model. The algorithm is trained and tested in Pascal VOC2007 dataset. The experimental results show that the mean average precision of the objects in the dataset is 73.87%. At the same time, compared with the mainstream faster RCNN and SSD detection models, the mean average precision of object detection algorithm based on DenseNet is improved by 5.63% and 3.86%, respectively.

1. Introduction

Traditional object detection algorithms mainly include object feature extraction, object recognition, and object positioning. Dalal and Triggs proposed the Histogram of Oriented Gradient (HOG) feature for computer vision and image processing in 2005 [1]. The core idea is that the shape of the detected local object can be described by the light intensity gradient or the edge direction distribution [1]. HOG features are widely used in image recognition, especially for pedestrian detection, and a large number of optimization and improvement algorithms have been proposed successively. The detection algorithm based on the Deformable Part Model (DPM), proposed by Felzenszwalb in 2008, is very robust to deformed objects, and at that time it became the core of many algorithms for classification, segmentation, and posture estimation [2]. As an extension of the HOG algorithm, the design idea of DPM is similar to that of HOG, which adopts the improved HOG features, and it combines the SVM classifier and sliding window detection ideas [3]. Aiming at the problem of multiview and deformation of the detected object, a multicomponent and graph-structure-based component model strategy was adopted. In addition, DPM algorithm takes the model category of the sample and the location of the component model as potential variables and uses Multiple Instance Learning to determine automatically. However, these traditional detection algorithms do not achieve satisfactory results. The

object detection algorithm based on deep learning has become the mainstream. Its process can be expressed as the extraction of image deep level features and the object classification and regression based on convolution neural network [4]. The feature extraction backbone is a convolutional neural network, AlexNet, which has won in the ImageNet classification competition [5, 6], and various convolutional neural networks (VGGNet, GoogLeNet, ResNet) proposed by other scholars have brought deep learning into a new stage [7–9]. These convolution extraction and information fusion operations not only have broad applications in the field of image processing, but also play an important role in signal recognition and modulation, optical communication, natural language processing, financial big data, and other fields [10–13]. The object detection algorithms based on deep learning can be roughly divided into two categories: (1) The first category is the object detection algorithms based on the proposal region, which mainly includes two stages. Firstly, a series of proposed regions are generated by using the region proposal network, and then the proposal regions are classified by a classification network and regress the real boundary of the object. Typical examples of this type of algorithms are RCNN, fast RCNN, faster RCNN [14, 15]. (2) The second category is the regression-based object detection algorithms. This type of algorithms does not require the proposal region extraction stage. The single-stage detection algorithm only needs to obtain the position and category information of the object through the network once. Compared with the object detection algorithm based on the proposed region, the detection speed will be greatly improved, which is more suitable for mobile devices [16]. Common first-order detection algorithms include YOLO series, SSD, and DetectNet [17–19].

In the existing two-stage object detection network, faster RCNN is an advanced algorithm, the author presents a proposed region extraction network with shared features [15]. The application of this network makes the performance of the algorithm further improved. However, VGGNet, its backbone network, is an image classification network based on the pretraining of ImageNet [22], which has the feature of location insensitivity [6]. With the continuous subsampling of VGGNet, the information of some smaller objects are filtered out, resulting in the incomplete information of the feature map input into the regional proposal network [7, 8].

In view of these shortcomings, an object detection algorithm based on dense network and multilayer information fusion is proposed. The fusion of different scale features is an important means to improve detection performance [23]. The low-level features have higher resolution and contain more location and detail information [24]. However, due to less convolution, they have lower semantics and more noise [25]. High-level features have stronger semantic information, but the resolution is very low, and the perception of details is poor [26].

This paper combines the multiscale feature fusion, feature sharing, and other ideas to optimize the extraction of image features, strengthen the use of different levels of features, and then improve the performance of the algorithm. In order to obtain different levels of features with rich information, the feature extraction network of this algorithm adopts the dense connection network with fewer parameters and less computation. Then, the features of different levels are fused, extracted, and input into the improved region proposal network. Then, the proposed regions are classified and regressed.

2. Research on Object Detection Algorithm Based on Multilayer Information Fusion

The detection flow of this algorithm is shown in Figure 1, which mainly includes four parts: (1) multilevel feature extraction; (2) multistage feature dimension specification and feature fusion; (3) region of interest extraction, classification, and regression; and (4) calculation of multitask loss. In the first part, the deep convolution neural network is used to extract image features. Compared with other feature extraction networks, DenseNet alleviates the problem of vanishing gradient, strengthens feature propagation, encourages feature reuse, and reduces the number of parameters [27]. This algorithm is based on the two-stage target detection theory. In a series of faster RCNN algorithms, the model is generally not ideal for small-scale object detection. In the second part, considering the multiscale situation of the detected target, in order to realize the utilization of different levels of features, the feature information of different levels generated by DenseNet is normalized in dimensions, and then the normalized features are fused. In the third part, the fused feature is used as input, and the region of interest is extracted by region proposal network; then, the region of interest is predicted by multiple categories and the accurate regression of the border. Finally, the loss function of the whole algorithm is calculated, and the weights and offsets are updated.

2.1. Feature Extraction by DenseNet. Compared with the previously proposed classification network, DenseNet has made structural changes [27]. DenseNet does not continue the stereotyped thinking that residual networks deepen network layers and GoogLeNet widens network structure to improve network performance, but from the perspective of features, through feature reuse and bypass settings, it not only greatly reduces the number of network parameters, but also alleviates the problem of vanishing gradient to a certain extent. The main structure of DenseNet is shown in Figure 2; it is mainly composed of multiple dense blocks and transmission layers. In order to get different levels of image features from the feature fusion network, we take the output of each transition layer as the input of the feature fusion network (C1, C2, C3, C4). The specific structure of dense block network is shown in Figure 3. In order to further optimize the propagation of information flows between layers, dense networks connect all layers. At the same time, in order to ensure the characteristics of forwarding propagation, the input of each layer is the output of all previous layers. Suppose that a dense block has l layers in common, and each layer of network contains a nonlinear transformation $H_i(x)$. The specific transformation is shown in (1)

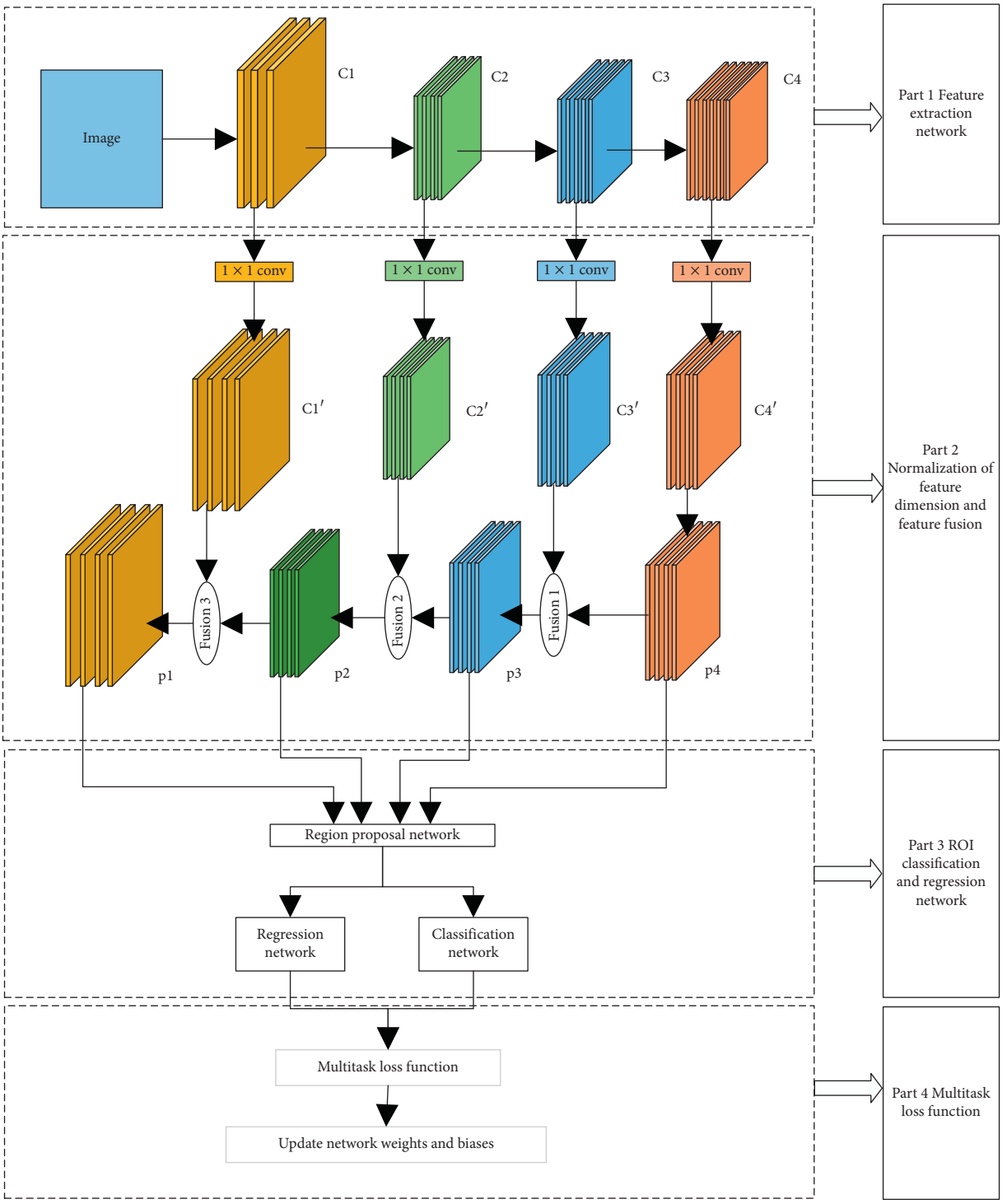


Figure 1: Algorithm structure diagram.

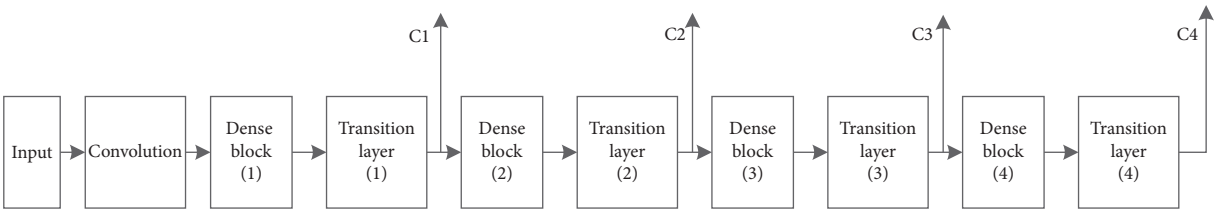


Figure 2: DenseNet structure diagram.

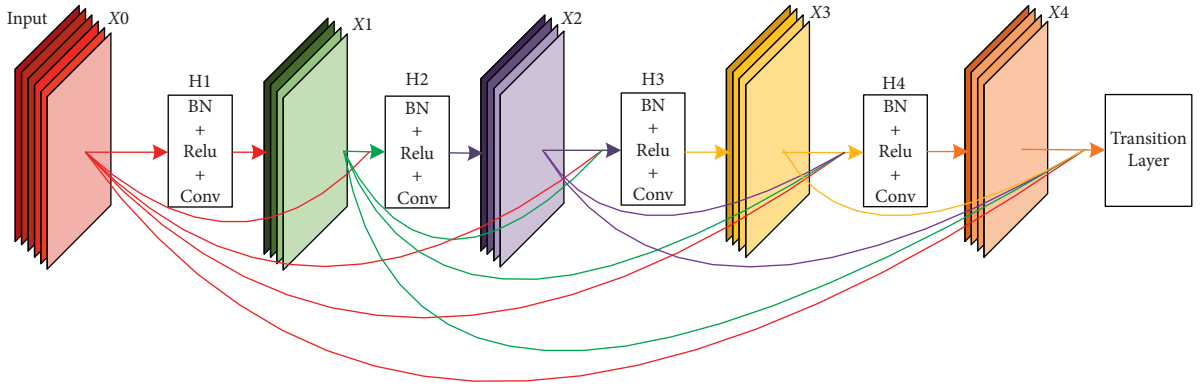


Figure 3: Structure of dense block network (the number of output channels of H (*) convolution kernel is 4).

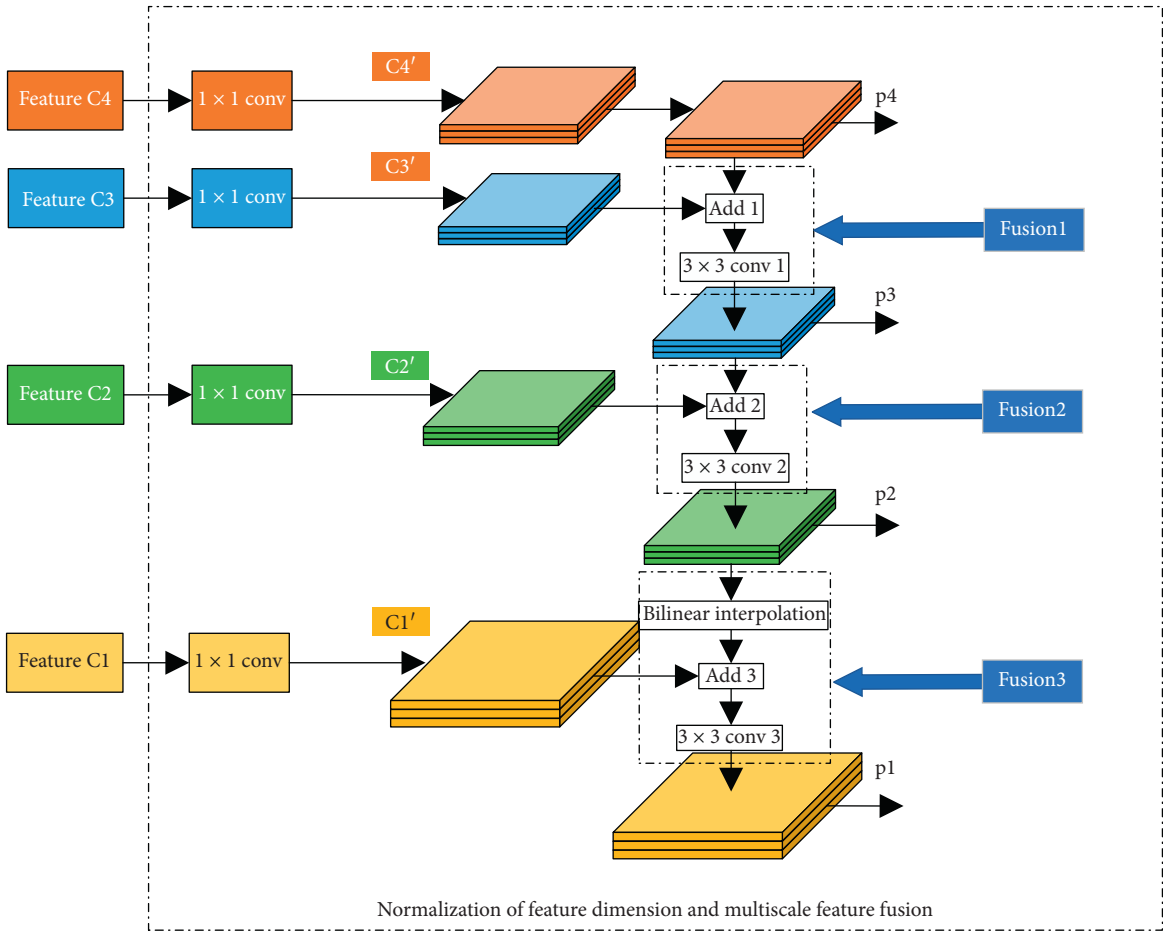


Figure 4: Feature fusion network.

and (2). The characteristic output of the i layer is recorded as X_i. As shown in Figure 3, the input of the i layer is not only related to the output of the layer, but also to the output of all previous layers:

$$H_i(x) = \text{BN} + \text{Relu} + \text{Conv}(3\text{x}3), \quad (1)$$

$$X_l = H\left(\left[X_0, X_1 X_{2\ldots} X_{l-1}\right]\right). \quad (2)$$

2.2. Multiscale Feature Fusion Network. The structure of the feature fusion network in this paper is shown in Figure 4. The image to be detected contains objects of different sizes. In this paper, we use the dense feature extraction network to extract the feature maps of different scales (C1, C2, C3, C4). In order to achieve the effect of feature sharing and more accurate detection, the feature maps of different stages are fused in a pyramid manner and then input to the region proposal network for prediction. The multilevel feature information is fused according to the characteristics of the

structure; besides, the low-resolution, high-level features are connected with the high-resolution, low-level features from the top to the bottom, so that the features of all scales contain the feature information of different sized objects, which to a certain extent increases the perception ability of the detector to the information. Compared with the faster RCNN algorithm, the original faster RCNN only uses the last layer of feature information of the feature extraction network. In this algorithm, the multistage feature fusion is used to extract the proposed region, not only the last P1 stage feature, because the subsequent region proposed network is a sliding window detector with fixed window size. Therefore, sliding in different layers of the fusion network can increase its robustness to target scale changes. In addition, there will be more anchors in the last stage. Increasing the number of mapped anchors will not effectively improve accuracy.

The left structure of Figure 4 is the dimension normalization process of different level features. In this paper, the output of each transition layer (C1, C2, C3, C4) is taken as the input of feature fusion network. The dimension and resolution of the feature map extracted by the backbone network are different. Before fusion, the dimension of different levels of features is normalized. All extracted features are convoluted by 1×1, and the information of different channels is combined linearly to reduce and increase the dimension without damaging the expression ability of the model. Besides, the nonlinear characteristic is added on the premise that the size of the feature graph is kept unchanged. The unified features are $C1'$, $C2'$, $C3'$, and $C4'$. The resolution of $C1'$, $C2'$, $C3'$, and $C4'$ characteristic map is 28×28, 28×28, 28×28, and 56×56, respectively. On the right side of Figure 4 is the process of horizontal and vertical fusion. The specific fusion processes fusion1, fusion2, and fusion3 are shown in the dotted boxes on the right. In the fusion process of $P4'_k$and $C3'_k$, $P4'_k$ and $C3'_k$ have the same feature size, and $P4'_k$ does not need the upsampling process and directly adds $C4'_k$ and $C3'_k$. The fusion operation of $P3'_k$and $C2'_k$ is the same; in the fusion process of $P2'_k$and $C1'_k$, the dimensions of the two sets of feature images are different, and the bilinear interpolation operation is used to restore $P2'_k$to $C1'_k$. The subscript k represents the k-th dimension of the feature, for example, $(C4' = \sum_{k=1}^{256} C4'_k)$. The addition operation calculation of the two features of the k-th dimension is shown in

$$Z_{k(x,y)} = f\left(A_k, B_k\right) = \beta_1 A_{k(x,y)} + \beta_2 B_{k(x,y)}. \tag{3}$$

Formula (3) shows that the corresponding elements of (x, y) positions of feature A_k, B_k are added, and the added results of all positions are taken as the added features. β_1, β_2 are used to balance feature A_k and feature B_k,

In order to eliminate the aliasing effect after fusion, the convolution check of 3×3 is used to convolute each fusion result. The output of the fused network is P1 (28×28 $d = 256$), P2 (28×28 $d = 256$), P3 (28×28 $d = 256$), and P4 (56×56 $d = 256$), and then these fused features are input into region proposal network.

2.3. Region of Interest Extraction, Classification, and Regression Network. This algorithm uses region proposal network to extract region of interest. The essence of RPN network is based on the sliding window and no class target detector [16]. The input of RPN network is the feature map of different sizes returned by the basic network and the output of region of interest. The processing flow of the region proposal network to the feature map is shown in Figure 5. In order to generate candidate regions, the algorithm slides a window on multiple size feature graphs, and the anchor mapping mechanism plays a key role in the network. Anchor points are placed on pictures of different sizes and scales with fixed borders and are used as reference borders in the prediction of later object positions. The region proposal network provides two full connection outputs for each anchor. The first output is the probability of the anchor as the object, so as to judge whether the anchor maps to the real object. This process only judges whether the object is included and does not classify the object specifically. The second output is border regression, which adjusts the anchor point to fit its predicted object better.

The ROI pooling layer takes the proposed region generated by the region candidate network and the feature map extracted by the feature network as the input information to get the fixed size feature map of the proposed region. After entering the ROI pooling layer, it forms the fixed size feature map for full connection operation and uses the normalization function to classify the specific categories, At the same time, the Smooth_{l1} loss function is used to complete the regression operation to obtain the exact position of the object.

2.4. Multitask Loss Function. In the region proposal network, we set a label of two types for each anchor: positive and negative. The positive sample is the anchor point with the highest ratio of the intersection with the real border. If the Intersection over Union with the ground truth is less than 0.3, the anchor point is set as a negative sample [10, 11]. The loss function of RPN network is defined as

$$L\left(\{p_i\}, \{t_n\}\right) = \frac{1}{N_{\text{cls}}} \sum L_{\text{cls}}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{\text{reg}}} \sum p_i^* L_{\text{reg}}\left(t_n, t_n^*\right). \tag{4}$$

Formula (4) is divided into two parts, the first part is the classification loss, and the second part is the regression loss of the object boundary. As shown in (5), p_i is the probability of the anchor point prediction as to the object, and p_i^* is the real label of the dataset. As shown in (6) and (7), t_n represents the four parameterized coordinate vectors of the predicted bounding box, and t_n^* is the vector of the real box matching the positive anchor point:

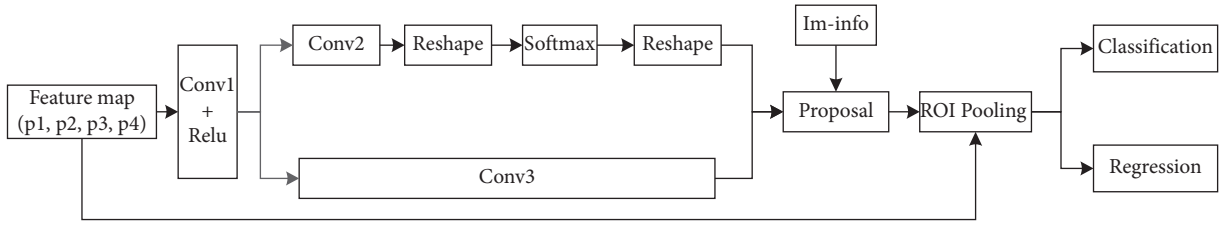


Figure 5: Extraction, regression, and classification process of candidate areas.

$$p_i^* = \begin{cases} 0, & \text{negative label}, \\ 1, & \text{positive label}, \end{cases} \tag{5}$$

$$t_x = \frac{(x - x_a)}{w_a}, \quad t_y = \frac{(y - y_a)}{h_a}, \tag{6}$$

$$t_w = \log\left(\frac{w}{w_a}\right), \quad t_h = \log\left(\frac{h}{h_a}\right), \tag{7}$$

$$t_x^* = \frac{(x^* - x_a)}{w_a}, \quad t_y^* = \frac{(y^* - y_a)}{h_a}, \tag{8}$$

$$t_w^* = \log\left(\frac{w^*}{w_a}\right), \quad t_h^* = \log\left(\frac{h^*}{h_a}\right). \tag{9}$$

x, y, w, and h in formulas (6) and (7) represent the coordinates of the center of the box, the width, and the height, respectively. x, x_a, x^* correspond to the prediction box, anchor point, and real box, respectively. In the classification loss of the first part, $L_{\text{cls}}(p_i, p_i^*)$ is the log loss of two categories (object, nonobject):

$$L_{\text{cls}}(p_i, p_i^*) = -\log[p_i p_i^* + (1 - p_i)(1 - p_i^*)]. \tag{10}$$

In the object box regression prediction in the second part, the least square loss function is usually used. However, the $L2$ loss has a high penalty for larger errors. This article uses the smoothing loss function Smooth_{l1} ($\delta = 3$), which is shown in the following formula:

$$\text{Smooth}_{l1} = \begin{cases} 0.5x^2 \times \frac{1}{\delta^2}, & \text{if } |x| < \frac{1}{\delta^2}, \\ |x| - 0.5, & \text{other}, \end{cases} \tag{11}$$

$$L_{\text{reg}}(t_i, t_i^*) = \text{Smooth}_{l1}(t_i - t_i^*). \tag{12}$$

These two parts of the loss are normalized by N_{cls} (the size of minibatch is 32) and N_{reg} (the number of anchor positions here is 5488) and weighted by a balance parameter λ to achieve the effect of balancing the weight of classification and regression parts.

The region of interest extracted by RPN network also needs to carry out the prediction of specific categories and fine-tuning of prediction frame. It is a process of classification and regression. The regression loss of fine-tuning candidate box is consistent with that of ROI, while the prediction of specific category is expanded from the two categories positive and negative to 20 categories. The specific 20 categories of classification and regression losses are shown in

$$L(\{p\}, \{t_n\}) = \mu \sum L_{\text{cls}}(p, u) + \lambda \frac{1}{N_{\text{reg}}} \sum p^* L_{\text{reg}}(t_n', t_n^*), \tag{13}$$

where $p = (p_{0,\ldots,} p_k)$ is the classification network output to predict the discrete probability distribution of each region of interest; u is the real category of each region of interest; and μ is the equilibrium coefficient of two loss functions, $\mu = 0.5$. t_n' denotes the four parameterized coordinate vectors of the last fine-tuning bounding box. In order to extract the region of interest accurately and carry out specific classification and boundary fine-tuning later, the total loss function of the algorithm is set to the sum of the above two losses (14), so that when training the network, the weights of the two stages can be updated at the same time:

$$L_{\text{total}} = L(\{p_i\}, \{t_n\}) + L(\{p\}, \{t_n'\}). \tag{14}$$

2.5. Specific Steps of the Algorithm

(1) In order to obtain the features with different levels of information, we use DenseNet network to extract the multistage features of the image and take the output of transition layer in DenseNet as the features of different stages of extraction.

(2) Because the feature dimensions of different stages are different, they are normalized. Using the characteristics of 1×1 convolution, the feature dimensions of different stages are normalized to 256, and then the fusion operation shown in Figure 4 is carried out. The specific fusion calculation is shown in

$$Cn' = Cn * \text{conv1}x1_d \quad (n = 1, 2, 3, 4,\ d = 256), \tag{15}$$

$$\begin{aligned}
\text{P4} &= \text{C4}', \\
\text{P3} &= \sum_{d=1}^{256} f_{\text{add}}\left(\text{P4}_d, \text{C3}'_d\right) * \text{conv3} \times 3 \\
&= \sum_{d=1}^{256} \left[\left(\beta_1 \text{P4}_d + \beta_2 \text{C3}'_d\right) * \text{conv3} \times 3\right], \\
\text{P2} &= \sum_{d=1}^{256} f_{\text{add}}\left(\text{P3}_d, \text{C2}'_d\right) * \text{conv3} \times 3 \\
&= \sum_{d=1}^{256} \left[\left(\beta_1 \text{P3}_d + \beta_2 \text{C2}'_d\right) * \text{conv3} \times 3\right], \\
\text{P1} &= \sum_{d=1}^{256} \left[f_{\text{add}}\left(2UP \cdot \text{P2}_d, \text{C1}'_d\right) * \text{conv3} \times 3\right] \\
&= \sum_{d=1}^{256} \left[\left(\beta_1 \cdot 2UP \cdot \text{P2}_d + \beta_2 \text{C1}'_d\right) * \text{conv3} \times 3\right].
\end{aligned} \tag{16}$$

In formula (15), Cn$'$ is the result of dimension normalization of different stage features, and $d = 256$ represents the dimension of normalized features. In formula (16), f_{add} represents the addition operation of the two features proposed in 2.2. β_1, β_2 play the role of weighting and balancing the features, $\beta_1 = \beta_2 = 0.5$.

(3) The region of interest is extracted from the fused feature using the region recommendation network. Rough prediction of the target can only determine whether it contains objects, not specific categories of prediction, and rough regression of the target location. The extraction of region of interest is completed, and then the prediction of specific categories and the precise regression of target location are carried out.

(4) In order to obtain the category and location of the target accurately, we use the multitask loss function designed in 2.4 to calculate the sum of the loss function of the proposed area and the final classification regression loss function. Then, we use the back-propagation algorithm to derive the total loss function, update the weight and bias parameters, and carry out multiple iterations to minimize the loss function.

3. Experiment and Discussion

3.1. Experimental Platform. A: The dataset used in the experiment is Pascal VOC2007. The whole dataset contains 9963 pictures with bounding box for training (5011) and testing (4952). These pictures are divided into four major categories (vehicle, household, animal, and person) and 20 specific categories. The image in the dataset is shown in Figure 6. In this paper, mean average precision is used to evaluate the performance of each algorithm.

B: All experiments were carried out on the Ubuntu 16.04 system. The hardware configuration of the system is Intel Core i5 9400 CPU, NVIDIA GTX1080 GPU, and 16G RAM.

The software platform for this experiment includes Anaconda 3, TensorFlow 1.13.2, Keras 2.1.5, and NumPy 1.17.4.

3.2. Contrast Experiment

3.2.1. Experimental Comparison between Multistage Prediction and Single-Stage Prediction. In the DenseNet of this paper, the 1×1 convolution and 2×2 average pooling operations are performed on the interconnected dense blocks (1) (2) by using the transition layers (1) (2). Although the average pooling operation can reduce the redundancy and increase the receptive field, the local location information is also lost in the process of subsampling, and there is a small deformation in the local area, resulting in the lack of deep feature location information. Therefore, the subsampling process of the transition layers (3) (4) is canceled. In dense blocks (3) (4), the original convolution operation is replaced by dilated convolution, and the size of the convoluted feature map is kept unchanged. In order to obtain image features at different levels for the feature fusion network, the output of each dense block is used as the input of the feature fusion network (C1, C2, C3, and C4).

Table 1 shows the structural parameters of DenseNet. In the process of setting the structure parameters, the number of channels of the initial convolution kernel is set to 16, and the number of channels K of each layer of convolution network of dense blocks is set to 24. In order to improve the simplicity of extraction network, in the transition layers (1) (2), the convolution of 1×1 is used to reduce the dimension, and the number of feature images is reduced to half of the input, and in the transition layers (3) (4), the dimension of dense block output feature is set to a fixed dimension ($d = 256$) by using 1×1 convolution.

In order to solve the problems of slow convergence speed and long training time of gradient descent in model training, this algorithm chooses to load the pretrained model of DenseNet-121 on ImageNet as initialization. ImageNet contains a large number of different types of data which have a variety of shallow and deep characteristics. These features are well extracted by the weights trained on ImageNet. There are commonalities between different image data, including visible and invisible features, so the weights on ImageNet can extract some common features of VOC dataset. At the same time, the loading of the pretrained model can avoid the detection model falling into the local optimum or saddle point.

Because the ImageNet of the pretrained model is similar to the data in the VOC set used in this algorithm, it is unnecessary to retrain the whole network in the subsequent fine-tuning training. We adopt the method of local freezing parameters. The weights of the first and second stages in the pretrained model DenseNet-121 are kept unchanged, and these weights are deployed to the first and second feature

(a)

(b)

Figure 6: Partial image of VOC dataset.

Table 1: The structural parameters of DenseNet-98.

Layers	Output size	DenseNet-98
Convolution	224×224 ($d = 16$)	7×7 conv, stride = 2
Pooling	112×112 ($d = 16$)	3×3 max pool, stride = 2
Dense block (1)	112×112 ($d = 160$)	$[1 \times 1$ conv + 3×3 conv$] \times 6$
Transition	112×112 ($d = 80$)	1×1 conv
Layer (C1)	56×56 ($d = 80$)	2×2 average pool, stride = 2
Dense block (2)	56×56 ($d = 368$)	$[1 \times 1$ conv + 3×3 conv$] \times 12$
Transition	56×56 ($d = 184$)	1×1 conv
Layer (C2)	28×28 ($d = 184$)	2×2 average pool, stride = 2
Dense block (3)	28×28 ($d = 472$)	$[1 \times 1$ conv + 3×3 dilated-conv$] \times 12$
Transition Layer (C3)	28×28 ($d = 256$)	1×1 conv
Dense block (4)	28×28 ($d = 640$)	$[1 \times 1$ conv + 3×3 dilated-conv$] \times 16$
Transition Layer (C4)	28×28 ($d = 256$)	1×1 conv

extraction stages of the proposed algorithm by random screening method. Then the parameters of the third and fourth feature extraction stages are retrained. This ensures that the whole network can extract the features of the image in the VOC dataset more effectively in the subsequent training.

In the process of using GPU to train the model, we use the stochastic gradient descent (SGD) method to update the parameters of the network, because momentum can make the network more robust in training, and the weight attenuation regular term can reduce the training overfitting. In the whole process, the parameter momentum is set to 0.9, and the weight attenuation is 0.001. The total number of iterations is $40K$. In the initial $25K$ iterations, the learning rate parameter is set to 0.003. In the later $15K$ iterations, the learning rate is adjusted to 0.001. The reasons for setting the learning rate in this way are as follows: (1) In the initial stage of training, the model is far away from convergence, and setting a larger learning rate can speed up the training of the model. (2) When the model is close to convergence, the learning rate should be reduced to avoid the model falling into the local minimum.

The selection of batch size has an important impact on the training effect. When the batch size is smaller, the randomness of each batch of data is greater, and the convergence is worse. When the batch size is larger, each batch of training data can better represent the characteristics of the whole data. According to the experience and conclusion of fast RCNN and YOLO series algorithms, we set the batch size to 32.

In the process of training, the loss of verification set is calculated after each round of training. When the loss of verification set is no longer reduced, the training is stopped. This is because when the training set loss decreases and the verification set loss no longer decreases, the model will gradually overfit. The training time of the model is relatively long, and retraining every time will obviously reduce the work efficiency. In addition, the hardware equipment problems in the training process lead to abnormal program exit, which will cause a waste of time. For the parameters of the model to be used repeatedly after training, we will save the training results of the model every other period of time and save the final training results of the model after the

training. In the operation, the saver class of TensorFlow provides a convenient application program interface. We only need to provide a counter (counting once every $5K$ iterations) and storage location, and TensorFlow will automatically generate binary checkpoints file with training parameters.

In the contrast experiment of multistage prediction and single-stage prediction, the training method described above is adopted. In the process of extracting the region of interest, we set four cases to be compared, to verify the impact of multistage prediction and only-one-stage prediction on the detection effect of the algorithm [15].

In the first case, only one anchor box (one area, one length width ratio) is mapped in the final stage of the feature map (P4).

In the second case, only three anchor boxes (one area, three length width ratios) are mapped in the feature map (P4) of the last stage.

In the third case, one anchor box (one anchor frame area per layer, one length width ratio) is mapped on the feature map of four stages, respectively.

In the fourth case, three anchor boxes (one anchor box area per layer, three length width ratios) will be shown in the characteristic diagram of four stages, respectively. The comparison settings of prediction in different stages are shown in Table 2.

Table 3 shows the influence of prediction in different stages on experimental results. The prediction stage of Case 1 and Case 2 are the same as the characteristics of P4 stage. The difference is that the proportion of the two anchor mapping boxes is set by comparison. From the results of Table 3, we can see that the MAP of Case 2 is 0.9% higher than that of Case 1. In Case 3 and Case 4, we still ensure the same prediction characteristics and set different mapping anchor length to width ratios for the two cases. The MAP of Case 4 is 1.5% higher than that of Case 3. Through the analysis of the two comparison results, we conclude that the multiscale anchor mapping frame can better fit the location of the variable length to width ratio targets, thus improving the prediction accuracy of the model.

In Case 1 and Case 3, we set the mapping ratio of anchor points to be the same. The difference is that Case 3 predicts the anchor box in four stages. At the same time, according to the characteristics of different levels of features: deep features are used to predict larger targets, and larger anchor mapping box area is set; shallow features contain rich small target information and clear location information, so a smaller anchor mapping area is set. From Table 3, it can be clearly observed that the map of situation 3 has been improved by 6%; the comparison between Case 2 and Case 4 is similar to that of Cases 1 and 3, and the predicted characteristics are taken as the only variable. Through the observation of the results, there is a 6.6% gap in the map of the two situations.

Through the cross comparison of the four situations, we draw two conclusions:

(1) The length and width of the anchor mapping frame can better deal with the targets with different proportions in reality and improve the recognition accuracy of the algorithm, but the promotion space is limited.

(2) Compared with only the last extracted features, the multistage fusion feature prediction can predict different size targets separately according to the characteristics of the fused features. This operation effectively uses the existing image information, better mining image features, and significantly improves the detection accuracy of the algorithm. From the perspective of application, it is also closer to the actual detection requirements.

Figure 7 shows the average detection accuracy of the algorithm on 20 kinds of objects in VOC2007 dataset. From the detection results, it can be observed that the recognition accuracy of 12 categories exceeds the average level of 20 categories. The algorithm has a high recognition accuracy for cars, buses, bicycles, pedestrians, and other objects in daily life. For sheep, there is a certain gap between the plant detection effect and the average level. Through the analysis of sheep detection effect and sheep image data in dataset, we found that sheep and dog images have similar features, and sheep images lack obvious texture, edge, and color depth features, which leads to the decline of the recognition ability of sheep category. Because of the similarity between plant class and background, there is a situation that plant target is regarded as background.

3.2.2. Comparative Experiments of Different Algorithms. In order to verify the accuracy of the proposed algorithm for target detection, the algorithm is compared with fast RCNN (VGG16), faster RCNN (RPN), and SSD (300) to verify the detection effect of different models. In order to ensure the effectiveness of comparison, all algorithms are trained under the same hardware platform and software framework. The datasets used are all VOC2007 datasets, and the initialization and training mechanism used in Section 3.2.1 are used for the training of the four algorithms. The experimental results of the four algorithms are shown in Table 4. The classical target detection algorithms fast RCNN, faster RCNN, and SSD (300) achieved 68.24%, 70.01%, and 71.19% MAP. They are 5.63%, 3.86%, and 2.8% lower than the algorithm proposed in this paper.

Fast RCNN and faster RCNN only use the deep features of the last stage for prediction. The feature extracted by the two algorithms has incomplete information, because it does not consider that the actual semantic information of small targets has been lost after continuous convolution and pooling operations [28]. Finally, the overall detection performance of the algorithm is affected. In this paper, the independent prediction in different feature layers overcomes the defect of fast RCNN series algorithm that only uses deep features, thus significantly improving the detection accuracy of the model.

SSD (300) algorithm uses the structure of feature pyramid to detect and extract multistage features to locate and detect the target in the image. However, SSD (300) gives up the shallow feature map in order to avoid using low-level

Table 2: Comparison settings of multistage and single-stage prediction.

Different cases	Area of anchor point	Length width ratio of anchor point
Case 1	112×112 (p4)	1 : 1
Case 2	112×112 (p4)	1 : 2, 1 : 1, 2 : 1
Case 3	112×112 (p4)	1 : 1
	56×56 (p3)	1 : 1
	28×28 (p2)	1 : 1
	14×14 (p1)	1 : 1
Case 4	112×112 (p4)	1 : 2, 1 : 1, 2 : 1
	56×56 (p3)	1 : 2, 1 : 1, 2 : 1
	28×28 (p2)	1 : 2, 1 : 1, 2 : 1
	14×14 (p1)	1 : 2, 1 : 1, 2 : 1

Table 3: Experimental results of prediction in different stages.

Prediction of different stages	MAP (%)
Case 1	61.4
Case 2	64.3
Case 3	70.4
Case 4 (proposed)	73.9

features, which are very important for small target detection. From the results of the comparison with SSD (300) algorithm, it can be concluded that the multilayer fusion structure designed in this paper can make the shallow features with higher resolution also have the semantic information of deep features through the continuous downward fusion of deep features. Therefore, in the subsequent prediction, the recognition accuracy of the model has been significantly improved, and the MAP is 73.87%.

3.2.3. Comparison between the Proposed Algorithm and SSD (300) in the Actual Scene. In order to verify the detection effect of the proposed algorithm for complex scenes in real life, the experimental image data are taken from the more common complex scenes in life. It mainly includes the image of the city road downloaded from the Internet, the image of the university campus where the author works, and the image of students participating in extracurricular activities. This part compares the proposed algorithm with SSD (300) in these scenarios. The reason for making a comparison with SSD algorithm is that SSD has higher detection accuracy for objects and uses multiscale feature map to predict the structure. Figures 8(a)–8(f) show the comparison between the proposed algorithm and the original SSD300 algorithm. The left half of the image is the detection effect of the proposed algorithm, and the right half of the image is the detection effect of SSD algorithm.

Figure 8(a) shows the detection effect of the two algorithms on the campus scene of the author's university. It can be observed from the image that both algorithms accurately detect the target in the image, but the proposed algorithm has higher detection accuracy for people and vehicles than SSD algorithm.

Figures 8(b)–8(d) show scenes of urban roads in life. In this scene, the target is complex and variable, and the size of the target is different, which can better measure the detection

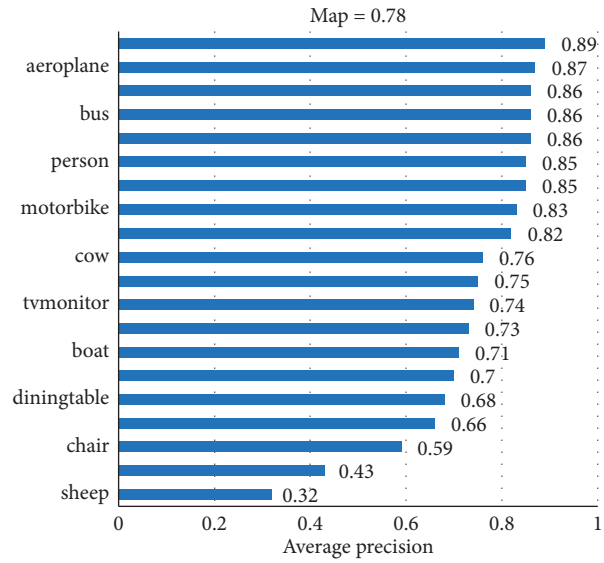


Figure 7: Test results of algorithm on VOC2007 test set.

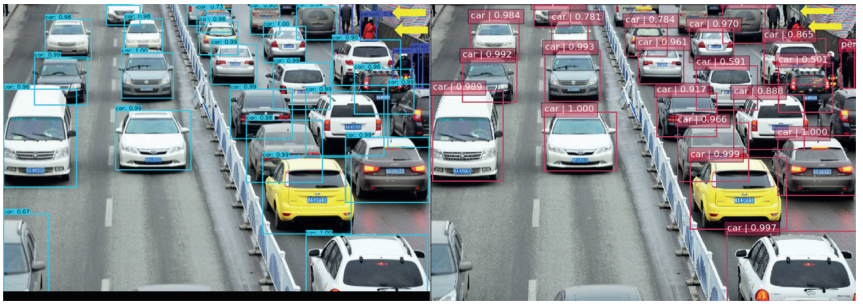
performance of the algorithm. From the contrast images, it can be observed that the proposed algorithm is more sensitive to small targets and can accurately detect the pedestrians and vehicles of remote small targets. In Figure 8(b), our algorithm can more accurately detect pedestrians at the edge of the guardrail. In the comparison of Figure 8(c), the proposed algorithm can detect buses farther away and can detect more targets than SSD (300) algorithm. In Figure 8(d), both algorithms detect some cars, but the proposed algorithm can accurately distinguish and recognize overlapping vehicle targets.

Figures 8(e) and 8(f) show the scene of students' participation in extracurricular activities. From the two recognition results, we can see that both algorithms can detect some humans when the crowd is dense. However, SSD algorithm is not sensitive to overlapping targets and human background information, and the proposed algorithm can accurately detect overlapping targets. In comparison with Figure 8(e), SSD does not accurately detect the three human targets in the front row and identifies two overlapping targets as one target. In Figure 8(f), SSD recognizes the human target in the back of the camera as the background, and the proposed algorithm can identify the back features of the human targets, which improves the detection effect of the algorithm.

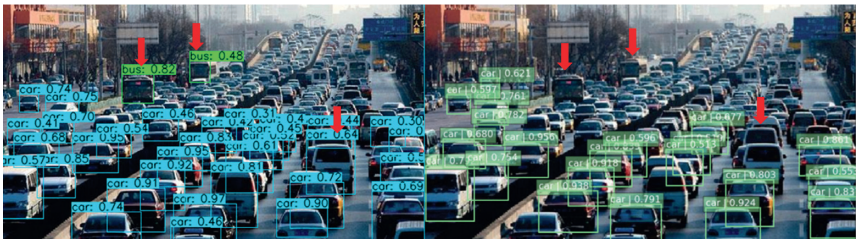
Table 4: Experimental results on the PASCAL VOC2007 dataset.

Model	Fast RCNN (VGG16)	Faster RCNN (RPN)	SSD (300)	Proposal
aeroplane	64.15	67.25	69.94	87.13
bicycle	73.14	75.51	77.34	85.65
bird	65.38	68.96	69.83	72.95
boat	55.25	56.94	59.41	70.71
bottle	51.34	52.71	51.46	66.05
bus	75.93	76.73	78.92	86.10
car	76.55	78.65	79.59	89.50
cat	86.77	89.95	88.62	84.93
chair	49.22	50.21	52.82	59.12
cow	74.34	75.96	77.35	76.33
diningtable	63.55	65.15	67.38	68.32
dog	80.05	81.47	82.49	75.11
horse	79.07	81.21	82.83	86.20
motorbike	73.11	74.23	76.71	82.86
person	78.34	79.14	79.85	85.12
potted plant	42.76	44.76	44.06	43.41
sheep	67.14	66.78	65.33	31.60
sofa	63.02	66.32	67.94	70.26
train	75.32	76.83	78.79	81.99
tvmonitor	70.34	71.43	73.14	73.96
MAP	68.24	70.01	71.19	73.87

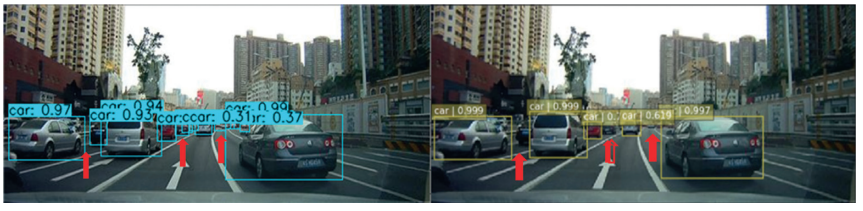
(a)



(b)



(c)



(d)

Figure 8: Continued.

(e)

(f)

Figure 8: The detection results of the proposed method (left side) and the original SSD300 (right side) algorithm in complex scenes of life. Compared with the original SSD algorithm, the proposed method detects more targets and has higher detection accuracy.

From the recognition results of these real scenes, we can get the good detection performance of the proposed algorithm for multiscale targets and small targets, which verifies the superiority and feasibility of the proposed algorithm in structural design.

4. Conclusion and Outlook

In the process of target recognition, the detection of small size target is a prominent problem. In this paper, we propose a detection framework for different scale objects, using DenseNet as the basic feature extraction network; in order to ensure the transmission of feature information, multiscale features are fused. In order to better identify the target, different anchor mapping areas are designed for different sizes of the objects. From the above experimental results, it is obvious that the algorithm in this paper is more efficient compared to other algorithms; it can effectively reduce the amount of computation in each layer of the network and achieve the purposes of reducing redundancy and inhibiting overfitting. The feature information extracted by the convolution neural network is fully utilized, and the transmission of feature information is strengthened. The feature fusion makes the feature map of different sizes simultaneously have the semantic information of high-level features and the location information of low-level features. At the same time, different size anchors should be used for different levels of objects, which improves the detection accuracy of different size objects. Compared with other detection algorithms, the framework has good scalability and can be applied to specific detection scenarios such as aerial images and pedestrians. In the future, the algorithm will continue to be optimized migrating to the direction of semantic segmentation.

Data Availability

The codes used in this paper are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors acknowledge the support from the Fundamental Research Foundation for Universities of Heilongjiang Province, under no. LGYC2018JC045, and the National Natural Science Foundation of China, under Grant nos. 61803128 and 61671190.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, San Diego, CA, USA, June 2005.

[2] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[3] M. Tan, G. Pan, Y. Wang, Y. Zhang, and Z. Wu, "L1-norm latent SVM for compact features in object detection," *Neurocomputing*, vol. 139, pp. 56–64, 2014.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Neural Information Processing Systems*, Toronto, Canada, January 2012.

[6] 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.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *in Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–14, Banff, Canada, April 2014.

[8] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2014.

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

[10] H. Yang, X. D. Zhao, Q. Y. Yao, A. Yu, J. Zhang, and Y Ji, "Accurate fault location using deep neural evolution network in cloud data center interconnection," *IEEE Transactions on Cloud Computing*, p. 1, July 2020.

[11] H. Yang, J. Zhang, Y. Zhao et al., "CSO: cross stratum optimization for optical as a service," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 130–139, 2015.

[12] S. Hong, Y. Wang, Y. Pan et al., "Convolutional neural network aided signal modulation recognition in OFDM systems," in *Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, May 2020.

[13] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[14] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2016.

[15] 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 & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[17] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 6517–6525, Honolulu, HI, USA, July 2017.

[18] W. Liu, D. Anguelov, and D. Erhan, "SSD: single shot MultiBox detector," *Computer Vision—14th European Conference, ECCV*, vol. 9905, pp. 21–27, 2015.

[19] X. Hua, W. Xinqing, W. Dang et al., "Multi-objective detection of traffic scenes based on improved SSD," *Acta Optica Sinica*, vol. 38, no. 12, Article ID 1215003, 2018.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[21] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proceedings of the European Conference on Computer Vision*, pp. 354–370, Amsterdam, Netherlands, September 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on Imagenet classification," in *Proceedings of the 2015 International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.

[23] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition CVPR 2017*, pp. 936–944, Honolulu, HI, USA, July 2016.

[24] X. Hu, X. Xu, Y. Xiao et al., "SINet: a scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1010–1019, 2019.

[25] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[26] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for background subtraction," *Pattern Recognition*, vol. 76, pp. 635–649, 2017.

[27] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, July 2017.

[28] H. Gao, B. Cheng, J. Wang et al., "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018.