Journal of Network Intelligence
Taiwan Ubiquitous Information

Volume **9**, Number **4**, November **2024**

Mechanism of Vision Object Detection Algorithm Combined with CNN

Bin Zhang*, Wan-Hua Chen, Wen-Zhi Cheng

School of Information Engineering,
Hunan University of Science and Engineering, Hunan 425199, China
huseComB@163.com, laipiwaner@163.com, chengwenzhi@126.com

*Corresponding author: Bin Zhang
Received January 16, 2024, revised April 30, 2024, accepted August 4, 2024.

Abstract. *Object recognition has been studied extensively because of its wide range of applications. However, in real-world scenes, the identified objects often change in many aspects such as angle and occlusion, which leads to inaccurate recognition. Especially for some small samples, the recognition effect is more limited. To address this issue, a convolutional neural network model is proposed to optimize the Faster region. The three-stage framework of small sample recognition is designed in detail, and the concept of space pyramid pool network and multi-task loss function are introduced to optimize the performance of this model. Experimental results show that the proposed model achieves 0.569 recognition accuracy after optimization measures such as module transformation, which is an overall increase of 3.3% compared with the original model. To sum up, the research can still maintain a high recognition accuracy when facing the real image transformation scene.*
Keywords: convolutional neural network; small sample; fast algorithm; deformable module; object detection

1. **Introduction.** The progress of the world economy has led to the development of various industries towards intelligence. Artificial intelligence technology has gradually become a popular research trend and is widely applied in various fields such as agriculture, medical treatment, and criminal investigation. Among them, computer vision technology is an important branch in intelligence research, which has fully integrated into people's daily lives. The most typical belongs to Object Detection (OD) technology, such as identification detection, facial detection, or detection of illegal items during security, which has significant development prospects that cannot be ignored [1]. In the current era of intelligent construction, it will inevitably gradually sweep across various industries, bringing great convenience to people's work and life. In some product promotions, when property rights issues are involved, this technique can be used to identify and eliminate them. It can not only analyze and predict the market, but also protect one's own rights and interests, making it an important development path in e-commerce [2]. In terms of transportation, it can also be applied to fields such as intelligent navigation, making it the best choice to assist in traffic management. At the same time, it can also reduce human work pressure, achieve automation, and reduce work costs [3].

Target detection is fundamental to logo detection, and the task of both is to frame the target region of the input image and to recognize it. The earliest logo detection techniques are based on sliding machine window learning, which require feature extraction for each pixel in the image and classification using classifiers such as support vector machines.

As a result, the entire research has focused on feature extraction methods and classifier selection, which has undoubtedly generated a huge amount of work. Therefore, target detection methods based on Deep Learning (DL) networks are proposed after continuous research in the academic world. The key to DL network detection methods is that the improvement of their accuracy relies on training in large-scale data sets. With the rapid development of network technologies, DL has also gradually become a core technology in the field of computer research, but it still needs further analysis and improvement in the dependence on the amount of data. The small samples marking detection has become a heavy difficulty in the research, especially in the face of extremely high background complexity and too large labeling discrepancy in real-world operational scenarios. In the real-world application scenarios, objects will be affected by Angle transformation, scale, and occlusion, which will degrade the accuracy of the model. In traditional studies, keypoints are usually extracted by artificial definitions. This significantly reduces the recognition efficiency. Therefore, the research should focus on this problem. In addition, the identification detection itself is a small sample, and it is difficult to obtain a large data set. Therefore, studies should be targeted to optimize for such small sample characteristics. To address this problem, the study proposes a regional convolutional neural network detection model based on the Faster algorithm, which aims to construct a small-sample oriented detection system based on the concept of transfer learning. And the background richness at the data level can be solved, which improves the model expressiveness of the algorithm. The model's ability of scale transformations can be strengthened through the deformable modules.

The research mainly includes the following content. Firstly, it mainly introduces the structural framework design for small sample data. Then, the current research status and common models in this field are introduced. Next, the modeling and optimization process of the faster Region-based Convolutional Neural Network (R-CNN) model selected for this study is introduced. The next step is to conduct simulation performance experiments on the designed system using the FlickrLogo-32 dataset. The final section provides a detailed summary and analysis of the experimental data.

2. **Related Work.** OD is a relatively important research direction in computer vision analysis. Cheng believed that the quality of 3D data samples had a significant impact on the effectiveness of computer detection, especially for multi-view data. Based on this, their study applied the entropy-based multi-perspective information quantification model to the data evaluation system. Their model included three modules: hierarchical data, feature generation, and quantitative calculation. Finally, the effectiveness of the model was verified through simulation experiments [4]. An et al. believed that the required accuracy in the current detection field far exceeded the performance that traditional detection algorithms could achieve. They believed that un-optimized DL networks had significant shortcomings in nonlinear modeling and pooling repeatability. Therefore, a global CNN model with multiple parameter exponential linear units was proposed in their study as a detection system. And learning parameters were introduced in this experiment to represent piecewise linearity and exponential nonlinearity. Finally, simulation experiments were conducted on datasets such as ImageNet, demonstrating the stability and other advantages of this system [5]. Ma et al. found that the imbalance between the detection background and the target object was a bottleneck in aviation detection. They proposed a model for OD in aerial visual scenes based on the YOLOv3 network. To achieve the effect of balancing the median frequency, a feature pyramid module was also embedded in this experiment. This system had achieved corresponding improvements in detection accuracy and speed [6]. Leira et al. applied target detection technology, combined with

Kalman filter and tracking algorithm of constant speed motion model, to achieve target detection. And the global nearest neighbor algorithm was introduced in this experiment to achieve data association, and its effectiveness was verified through experiments [7].

Sujith and Sasikala believed that there were many bottlenecks in current video tracking and detection. Therefore, they combined a hybrid tracking model with a crowd behavior detection system and limited it using the minimum bounding rectangle in visual detection algorithms. And support vector machines and optimization algorithms were introduced in the experiment to achieve detection and tracking of dense populations [8]. Yi et al. believed that target detection played an important role in agricultural pest control. Taking the most harmful grasshopper plague as an example, this study proposed a random probability-based regional CNN model. In this experiment, target detection technology was combined to achieve the detection, localization and analysis of grasshoppers. By detecting the size, shape, and other features of the target, image classification was carried out and integrated to obtain the final confidence level. Simulation experiments showed that the system had significantly improved the network robustness and stability [9]. Rafique et al. applied CNN to visual detection. Through its segmentation technology, feature extraction was achieved during data preprocessing. Based on this, discrete cosine transform and wavelet transform features were analyzed, and genetic algorithms were introduced to achieve efficient OD function [10]. Rajjak and Kureshi applied DL networks to OD in high-resolution videos. A regional CNN model was proposed to solve the low learning efficiency caused by network depth, which was divided into different training blocks and achieved higher accuracy in detection [11].

Numerous studies have shown that DL techniques are widely used for object detection, but suffer from shortcomings such as duplicate pooling. Based on this, this study proposes an optimized fast R-CNN model for small sample data and introduces transformation modules and deformable modules to optimize and improve it for better target detection results.

3. Design of CNN-based visual OD optimization algorithm and construction of model framework.

In the real OD work, there is often a large amount of unlabeled data, resulting in unsatisfactory detection model performance. Motivated by this, this study focuses on small sample data and uses common DL networks to design detection models.

3.1. Introduction to the concept of relevant sample design based on the basic detection framework of transfer learning.

DL networks are commonly used in OD, and their classification effect is significant. However, this effect is based on massive annotated data. In practice, however, it is costly to obtain such data. Therefore, the research introduces the concept of transfer learning to carry out the small sample OD design. Figure 1 shows the concept of Transfer learning.

Two basic concepts are included in transfer learning, the first is the domain, which consists of a feature space and a marginal probability distribution. The former is essentially the joint conceptual distribution, which represents the probability that all conditions hold simultaneously. The second basic concept is the task, which is generated based on a specific domain and consists of a label space and a prediction function that realizes the prediction of a label for a particular sample in the domain. Migration learning refers to the correspondence between a source domain and a target domain. When there are far more of the former than the latter, it is often necessary to utilize migration learning to help the target domain model to introduce knowledge about the source domain. Migration learning is the basis for constructing a small-sample logo detection framework.

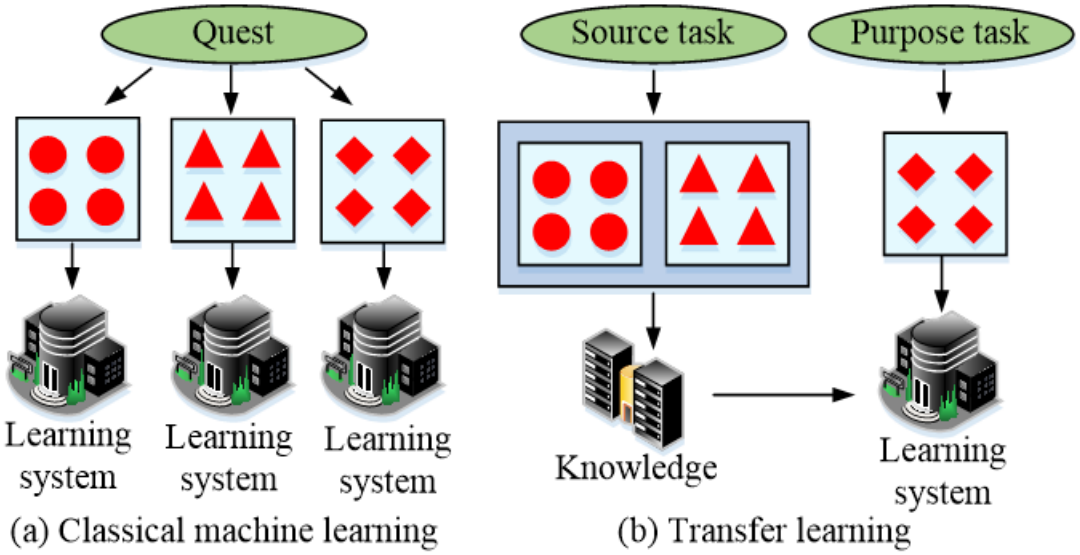


Figure 1. Migration learning diagram

Because its fine-tuning of the already trained magic knowledge, in contrast to zero-weight training, can significantly increase the generalization ability of the model and improve the efficiency as well as the operation accuracy. The framework mainly consists of three parts: initial modeling, synthetic modeling, and refinement modeling. The initialization modeling is to enhance the initial learning parameter starting point of the model, using ImageNet, the world's largest publicly available image recognition database, and embedding it into a base detection model for debugging. Synthetic data are used to further train the initial model to learn certain weight data in real samples to further enhance the model accuracy. The third training for real samples makes the model more idealized. As can be seen, the training phase of the synthetic samples is the most important. There are several related basic concepts that need to be understood before proceeding further with their design, including representation templates, background images, logo template transformations, and logo synthesis. The logo template is the primary part of the artificial data set synthesis and refers to the synthesized image for each type of logo. Background images refer to image components other than logos, i.e., environmental features, and commonly include pixel-based background opaque representation templates, as well as background transparent logo templates. In order to increase the richness of the training dataset, it is often expanded by data improvement. However, such an approach is not effective. In real detection, logos are often hindered from recognition due to changes in light intensity, distortions, etc. So the study performs template transformations on the dataset by affine transformation, evolutionary transformation, etc., to enhance the data diversity. Sign synthesis refers to the fusion of signs with the background to help them adapt to the background environment.

3.2. **OD strategy and design of small sample identification framework.** This section provides a further design of the synthetic data. The smooth acquisition of synthetic samples is a key to modeling success. The process of generating composite images can be roughly divided into selecting identification templates, selecting background images, template conversion, and image synthesis. The identification templates can be divided into transparent and opaque templates for the background. Transparent backgrounds can enhance their robustness relative to cluttered backgrounds, but are prone to damaging

the inherent attributes of the data. Pixel level opaque backgrounds have a stronger ability to retain detailed features, but due to this, some labels may be broken [12]. Based on the analysis of the strengths and weaknesses of each template, a template with pixel level background transparency was selected. Background information refers to the contact data generated between the target object and other scenes. The background image contains a large amount of contextual information, which greatly affects the detection accuracy. The study mainly uses images from the FlickrLogos-32 dataset without identifying information as background images. In addition, some diverse image data are selected on the Flickr database website. The augmentation of the training data also has a positive effect on the detection accuracy. Research has abandoned traditional expansion methods and instead opted for a template conversion method to deal with that objects are obscured and rotated during detection. Formula (1) calculates the rotation transformation of the image.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

In Formula (1) above, (x, y) represents the original pixel coordinates. (x', y') represents the pixel coordinates after rotation transformation. θ represents the rotation angle. However, integer coordinates may overlook some target pixel points, resulting in empty pixels. Therefore, the concept of reverse mapping is studied and improved in Formula (2).

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{2}$$

Prior to this, this model also needs to confirm the area where target object is located to achieve the rotation coordinate point confirmation in Formula (3).

$$\begin{cases} x = x'\cos\theta - y'\sin\theta + f_1 \\ y = y'\cos\theta + x'\sin\theta + f_2 \\ f_1 = \frac{w'-1}{2}\cos\theta + \frac{h'-1}{2}\sin\theta + \frac{w-1}{2} \\ f_2 = \frac{h'-1}{2}\cos\theta + \frac{w'-1}{2}\sin\theta + \frac{h-1}{2} \end{cases} \tag{3}$$

In Formula (3) above, h and w represent the height and width of the original image, respectively. h' and w' represent the height and width after transformation. Template transformation mainly includes color, affine, and Gaussian transformations. Wherein, Formula (4) is the calculation of Affine transformation [13].

$$I^* = PIR_xR_y \tag{4}$$

In Formula (4) above, I represents the input image data. I^* represents affine transformed image data. P stands for affine transformation. R_x, R_y represent the rotation matrix. Formula (5) is the calculation of color transformation.

$$c^* = rc \tag{5}$$

In Formula (5) above, c and c^* respectively represent the data before and after color conversion of the image. r represents a color change. In this system, each transformation template does not influence each other and is expanded randomly. Due to the translation invariance of CNN, research only focuses on two-dimensional transformations. The image converted from the identification template will be arbitrarily embedded into background image, that is, identification synthesis. The concept of Poisson fusion is introduced in the

study to enhance the compatibility between the background and the template in Figure 2 [14].

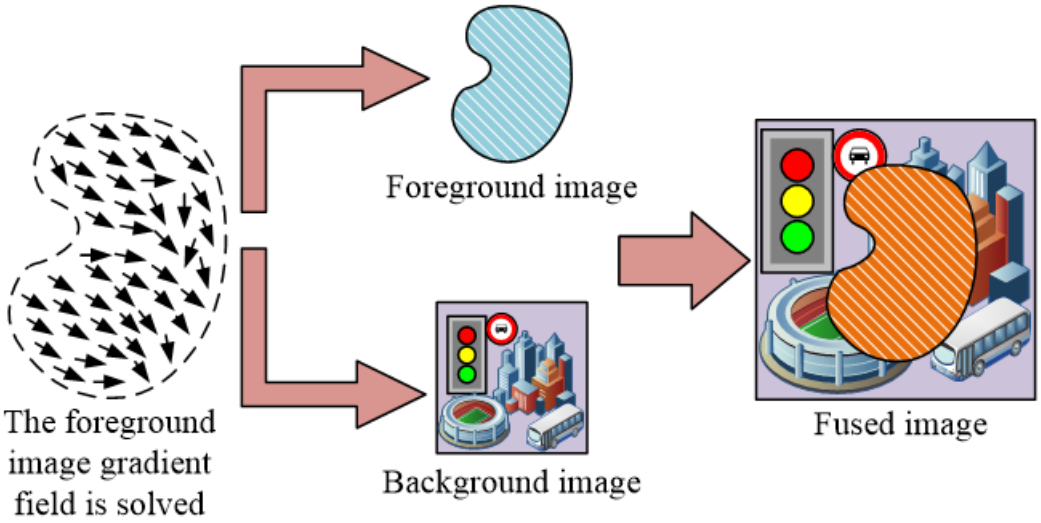


Figure 2. Concept diagram of Poisson fusion

Poisson fusion roughly consists of three parts. Firstly, the gradient field of the foreground image is solved to obtain an unmodified pixel gradient field. Then, the complete gradient field of the object waiting to be reconstructed is obtained using Poisson reconstruction. Finally, based on the first two data, its divergence is obtained [15]. Poisson fusion is easy to operate. And further research has added optimized boundary algorithms to reduce the computational burden, accelerate the calculation speed, and achieve higher model detection efficiency.

3.3. **Establishment of faster R-CNN detection optimization model.** CNN is a classical DL model commonly used in artificial intelligence. This model has no limit on the number of network layers and utilizes techniques such as local perception and pooling sampling to achieve feature extraction and matching. Region CNN (R-CNN) further achieves OD by introducing concepts such as linear regression, which has significantly improved accuracy and efficiency compared to the original network [16]. The workflow of R-CNN consists of four stages, namely selecting candidate boxes, feature extraction, data classification, and regression adjustment in Figure 3.

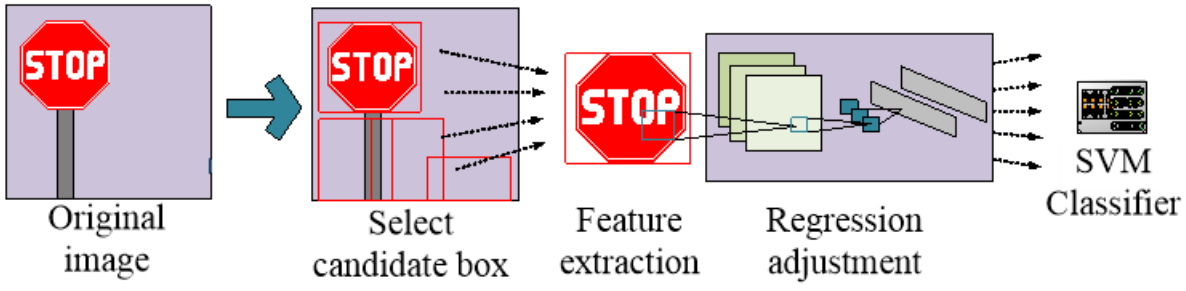


Figure 3. R-CNN working flow chart

The candidate boxes selection adopts Selective Search (SS) algorithm. The feature vectors obtained after feature extraction are imported into the classifier to identify objects. Finally, it is necessary to return to the fully connected layer for fine-tuning. This method is relatively simple in terms of box selection and more accurate in feature extraction. However, for the two different tasks of classification and regression, the temporal and spatial properties of the model will become more complex. FAST is introduced in this research, and the classification and regression steps are combined by using multi-task loss function to enable them to train. And the SSP-Net concept is introduced into ROI pooling layer in this research. The basic concept of this approach is to perform the box selection step on feature map formed after feature extraction. This can convert multiple convolution processes into one-time convolutions, greatly improving computational efficiency [17]. Faster R-CNN selects the region recommendation networks, integrating independent models into a single learning framework. And it hands over all algorithm detection work to DL network for processing. The workflow roughly includes four steps, starting with establishing a basic network. Then, a Region Proposal Network (RPN) is used to select candidate boxes. Subsequently, a fixed length conversion output is performed on pooling layer. Finally, there are classification and regression steps in Figure 4.

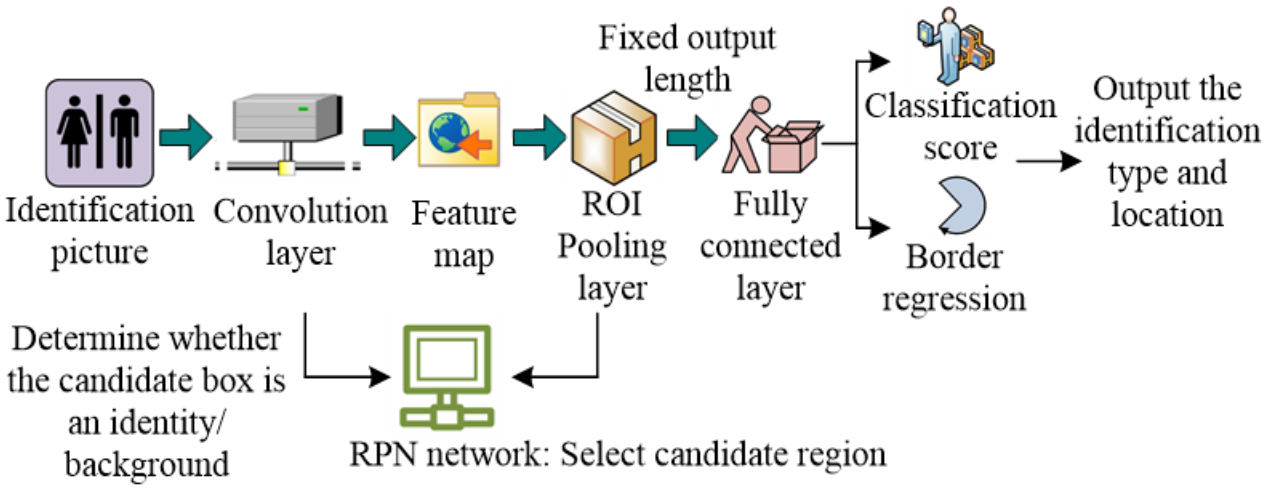


Figure 4. Work flow of faster R-CNN

The basic network is used for feature extraction, including 13 convolution layers, activation function layers, and 4 pooling layers. The convolutional kernel size is 3, and the pooling layer's kernel size is 2. The selected images are transmitted to RPN through feature maps in Figure 5.

The independent feature maps all use three different proportions of pixels to generate nine types of anchor points. The anchor structure includes foreground and background, which has offsets in four directions. The offset is calculated based on the ground reality with the highest overlap. After bounding box regression, this algorithm specifies a limit value, compares candidate boxes one by one, and compares boxes that are greater than the limit value. And the actual ground conditions are calculated through intersection and comparison, achieving further screening. The input data of the pooling layer are the candidate boxes' mapping values on feature map. The overall feature map is cut into sub windows and subjected to maximum pooling processing. Finally, a specific size

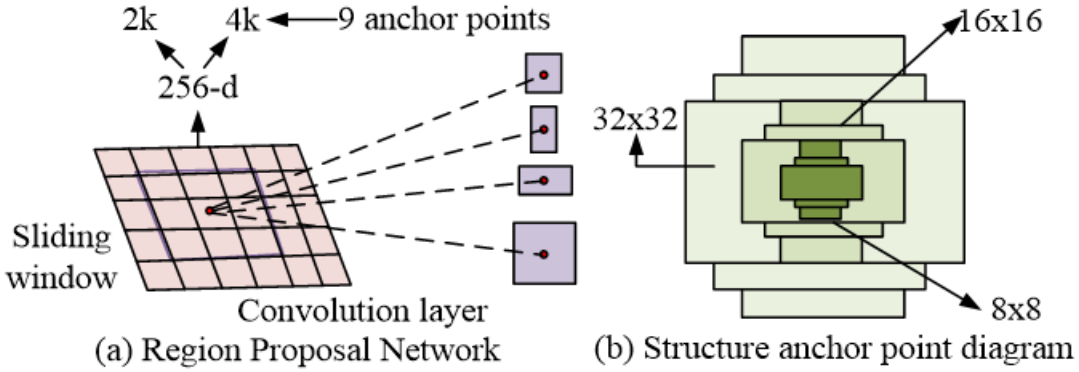


Figure 5. RPN and anchor point schematic diagram

feature map is output, entering the fully connected layer. The loss function calculates the classification and the regression loss in Formula (6).

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{6}$$

In Formula (6) above, i represents the anchor index within the trained identification image. p_i represents the probability that the anchor point is the correct target. t_i is the vector size of the candidate box at four coordinates, that is, the offset size estimated by the anchor point. t_i^* represents the coordinate vector size of the candidate box for the positive anchor point. L_{cls} and L_{reg} represent classification loss and regression loss, respectively. N_{cls} and N_{reg} represent the anchor point number when classifying and regressing, respectively. The classification loss represents the logarithmic consumption of correct identification and incorrect identification. Formula (7) represents two types of losses.

$$\begin{cases} L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1-p_i^*)(1-p_i)] \\ L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \end{cases} \tag{7}$$

In Formula (7) above, R represents the Smooth L_1 function. The classification loss function is a typical two classification cross-entropy function. After the logarithmic loss calculation of all anchors, the sum divisor operation is performed. Before training, N_{cls} is 256, and after training, it becomes 128. Formula (8) is the function expression [18].

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{8}$$

After the anchor completes the regression loss calculation, it needs to be multiplied by p^*, which takes the value of 0 or 1. Only when the target is the foreground target, losses need to be considered. The calibration box has a central position, and Formula (9) is the coordinate calculation required for the regression loss.

$$\begin{cases} t_x = (x - x_a)/w_a, & t_y = (y - y_a)/h_a \\ t_w = \log(w/w_a), & t_h = \log(h/h_a) \\ t_x^* = (x^* - x_a)/w_a, & t_y^* = (y^* - y_a)/h_a \\ t_w^* = \log(w^*/w_a), & t_h^* = \log(h^*/h_a) \end{cases} \tag{9}$$

In Formula (9) above, x and y represent the horizontal and vertical positions of the center of the bounding box. w and h represent the width and height of the box, respectively. Subscripts a and $*$ correspond to the coordinates of anchor bounding box and the ground truth bounding box, respectively. This model needs to accurately recognize objects with style changes. Therefore, this study introduces spatial transformation networks for further improvement. A spatial transformation network is usually inserted between two network layers and consists of a positioning network and a spatial transformation module. The former is used to build the structure of the Affine transform, and the training data for the spatial transform is the network parameter values. The latter generates the parameter value θ of Affine transformation, after which the data can be transmitted to the next network layer [19].

This research introduces variable convolution modules and pooling modules, aiming to enhance the OR ability and model learning ability of different geometric shapes. The central idea of both modules is to use the offset method for sampling. Compared with traditional fixed mode, the network sensitivity has been greatly improved, and it also has a certain strengthening effect on feedback transmission. When the detection object undergoes transformation, the convolutional kernel can also achieve automatic adjustment, as shown in Figure 6 [20].

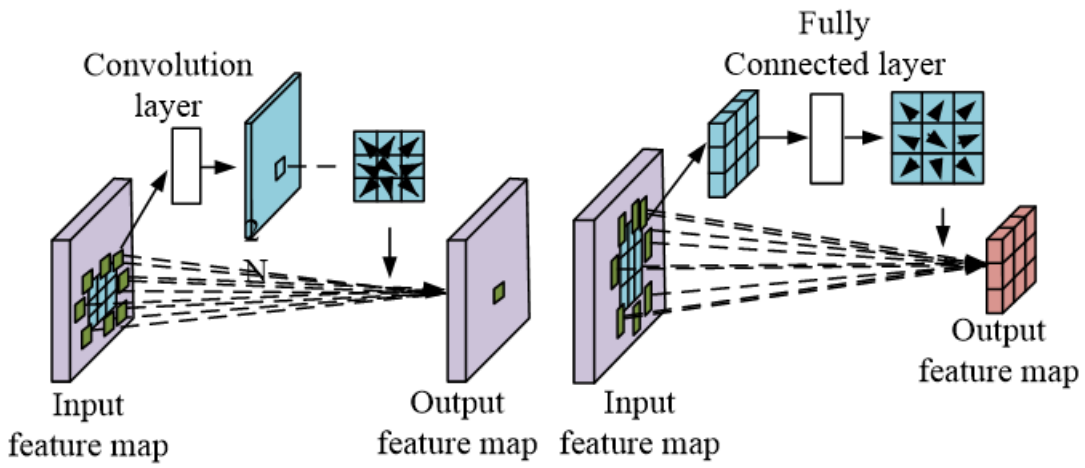


Figure 6. Structure diagram of deformable network

Taking the convolution kernel with size 3 as an example, initially, the feature maps obtained by the convolution module need to be sampled through the grid, and Formula (10) is used for its calculation.

$$R = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\} \tag{10}$$

In Formula (10) above, R represents the grid. In classical convolutional networks, when it is necessary to convolution the random point p_0 in the feature map, Formula (11) is used for its calculation.

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n) \tag{11}$$

In Formula (11) above, $y(p_0)$ represents the convolutional layer output value. p_n represents the offset. The variable convolutional model introduces an offset at all sampling points. In this way, the convolutional kernel can better recognize objects with varying

shapes. The offset is obtained through feature map learning, and both are used as input values to enter the next convolutional layer. Formula (12) is the final output result.

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{12}$$

In Formula (12) above, Δp_n represents the offset value added to all sampling points. Although adding an offset can improve network efficiency, it can also cause coordinates to become decimals. In this study, the concept of interpolation is used to implement the corresponding data validation of decimal coordinates. This is because after the model introduces compensation, the position will change accordingly, and then non-integer phenomenon will appear. The interpolation method can help the model effectively confirm the corresponding value of the non-integer position, as shown in Formula (13).

$$x(p) = \sum_{q} G(q, p) \cdot x(q) \tag{13}$$

In Formula (13) above, $x(p)$ represents the input feature map. w and Δp are the data to be trained. The convolution kernel with size 3 is transformed into a feature map of $2 \times 3 \times 3$ after convolution, representing the compensation value on the horizontal and vertical coordinates. The convolution is transformed into the offset of feature points in various directions through the compensation module. Formula (14) is the output of the traditional pooling layer [21, 22].

$$y(i, j) = \sum_{p_n \in bin(i,j)} x(p_0 + p)/n_{i,j} \tag{14}$$

In Formula (14) above, $n_{i,j}$ represents the drawing size. The deformable pooling module also learns from the fully connected layer by adding an offset in Formula (15).

$$y(i, j) = \sum_{p_n \in bin(i,j)} x(p_0 + p + p_{i+1,j})/n_{i,j} \tag{15}$$

So, the working principles of deformable convolutional modules and deformable pooling modules are similar. The whole process of building the detection model is step-by-step, as shown in Figure 7.

Starting from the understanding of convolutional neural networks, which are classical DL models commonly used in object recognition and computer vision tasks. Techniques such as local perception and pooled sampling are used to extract features and perform feature matching. Relative to the traditional form of image matching, convolutional neural networks have made great progress, but there are still some non-negligible problems, due to the differences in the aspect ratio and spatial location of the detection target. For example, there may be a situation in which the target object may occupy a large part of the picture or is very small, and the differences in its shape, etc., all of which place greater demands on the segmentation area of the convolutional neural network. This has further leads to an excessive computational burden on the model. Therefore, to improve this phenomenon, R-CNN is also born. It reduces the number of regions that need to be segmented by the model through the introduction of concepts such as linear regression, and further improves the accuracy and efficiency of object recognition. In summary, R-CNN is selected as the basic framework for object detection models. The workflow of R-CNN includes four stages: candidate frame selection, feature extraction, data classification, and regression adjustment. Among them, candidate frame selection usually uses SS algorithm, and then the extracted feature vectors are fed into the classifier for object

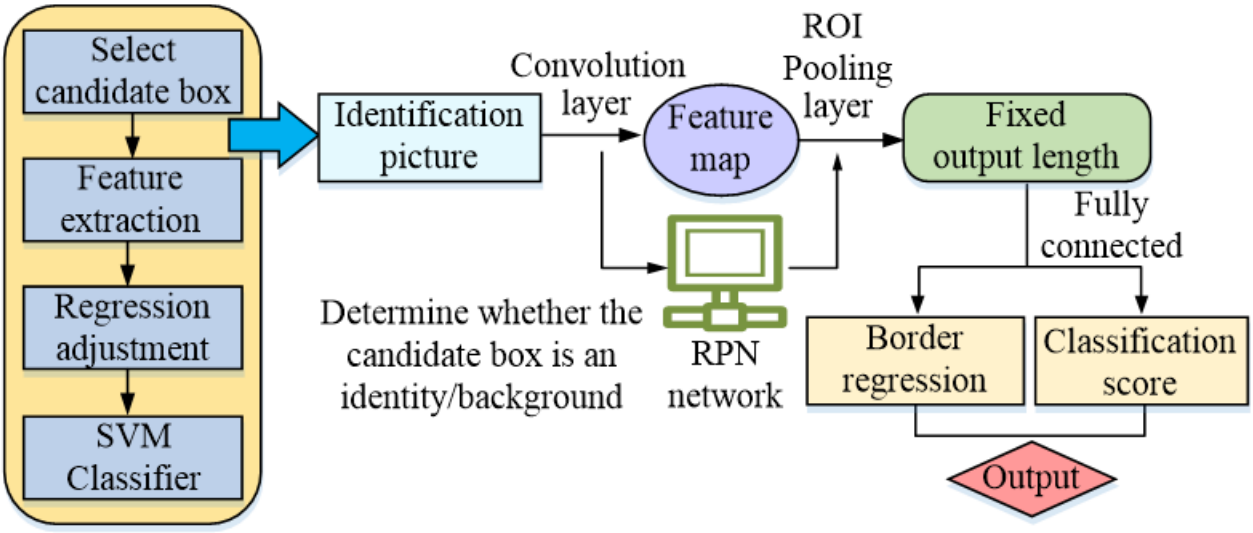


Figure 7. Comparison between the initial model and the final optimization model

recognition and fine-tuned by the fully connected layer. However, traditional R-CNN models use different stages for classification and regression tasks, leading to increased time and space complexity. To improve this problem, an algorithm called fast R-CNN is proposed, which trains the classification and regression steps simultaneously by introducing a multi-task loss function. Region Generating Network (RPN) is also introduced to improve the candidate box framing process to further deepen the optimization called faster R-CNN. The workflow of faster R-CNN is broadly divided into building the base network, candidate box selection using RPN, transforming the output of the candidate boxes at the pooling layer, and the classification and regression steps. Relevant concepts used in the above methods are as follows. (a) Regional Convolutional Neural Network (R-CNN): It utilizes concepts such as linear regression to enhance DL accuracy and efficiency. (b) SS: It is an algorithm used to select candidate frames for the candidate frame selection phase in R-CNN. (c) Fully Connected Layer: It is used to classify and fine-tune the extracted feature vectors. (d) Fast R-CNN: It improves model efficiency and accuracy by introducing a multi-task loss function and training both classification and regression steps simultaneously. (e) Faster R-CNN: An RPN for candidate frame selection and leaves the algorithmic detection is introduced to the DL network. (f) Spatial Transformation Network (STN): OD is improved with a localization network and a spatial transformation module that transmits data to the next layer of the network. (g) Variable Convolution Module and Pooling Module: OD with different geometries and model learning is enhanced by using offsets for sampling. Together, these techniques can improve the accuracy, efficiency, and robustness of the object recognition tasks. Through continuous improvement and introduction of new techniques, the performance of DL models in object recognition and computer vision tasks is continuously improved. The effectiveness of these efforts is demonstrated by improved target detection accuracy and increased efficiency. By introducing a multi-task loss function and a region generating network, the model is able to handle both classification and regression tasks and leave the algorithmic detection to the DL network, which makes the model more simplified and optimized. In addition, the spatial transformation network, the variability convolution module and pooling module further improve the performance and ability of the model to detect objects with different

geometries, which enables the model to successfully complete the task and enhance its robustness even when facing detection in complex scenes.

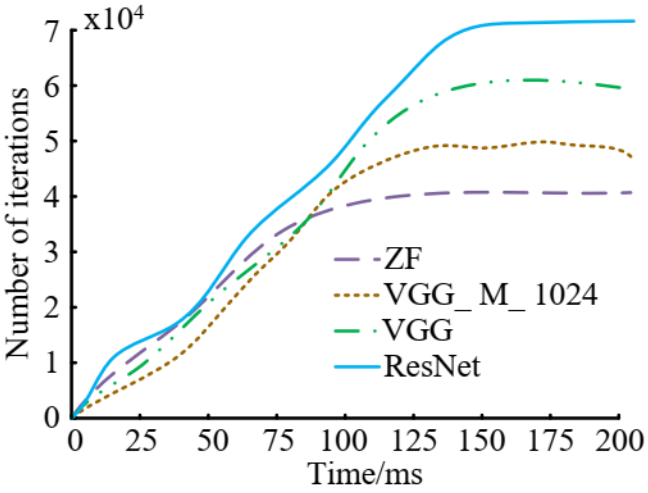
4. **Simulation experiment to verify the effectiveness of faster R-CNN model detection.** CNN has high requirements for datasets, and its training process requires the use of large-scale data information as a basis. The FlickrLogo-32 dataset was selected as the experimental data for this study. It contains a total of 32 categories and 8240 image data, all of which are publicly and fully annotated. The study divided it into three subsets with different purposes, namely a training set of 320 image data in 10 categories, a validation set of 3960 images in 30 categories, and a testing set of 960 images in 30 categories. The experimental operating environment is shown in Table 1.

Table 1. Setting and selection of the experimental operating environment

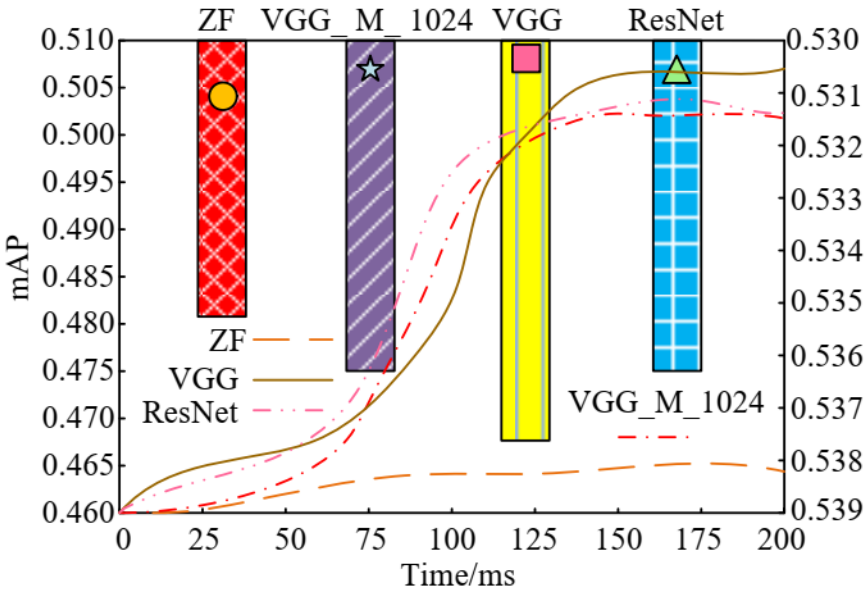
Environment	Name
Operating system	Windows 10 (64 bit)
Visual interface environment	Eclipse integrated development
Programming environment	JAVA
Front-end interface	JavaScript + MVC + SSH
Hardware environment	NVIDIA TITAN GPU
Archive	MySQL 5.5
Model deployment	Tomcat 7.0

4.1. **Comparison and performance experiment of module selection in the original faster R-CNN model.** The study used Mean Average Precision (MAP) as a performance evaluation indicator. The research on OR performance includes two parts: object classification and location identification. However, image categories and identifiers may not be the same, so choosing only classification indicators is incorrect. The research selects the commonly used VOC calculation method, which uses the average accuracy of specific 11 recall rates as the final solution and obtains it from the precision recall rate curve. This experiment was conducted in the NVIDIA TITAN GPU hardware environment. The study selected four different basic deep convolution models of ZF, VGG_ M_ 1024, VGG, and ResNet. Their performance was tested in faster R-CNN. The testing times for training set, validation set, and test set were set to 58ms, 72ms, and 160ms, respectively. Finally, the results in Figure 8 were obtained.

In Figure 8 (a), as the network complexity increases, the iteration number also increases. In Figure 8 (b), the algorithm performance increases with network complexity increasing. However, when the complexity reaches a certain level, the algorithm performance will actually decrease. In training set, the average accuracy of four ZF, VGG_ M_ 1024, VGG, and ResNet networks is 0.462, 0.472, 0.474, and 0.473, respectively. The ZF network has the lowest mAP value in training set. Compared with the best performing VGG network, it has decreased by 0.012. In validation set, the average accuracy of four ZF, VGG_ M_ 1024, VGG, and ResNet networks is 0.504, 0.506, 0.508, and 0.506, respectively. The more complex ResNet network has a lower mAP value than VGG network, which is the same as VGG_ M_ 1024 network. In test set, VGG network's mAP value also reached its highest value, increasing by 0.02 compared to ZF network and 0.01 compared to VGG_ M_ 1024 and ResNet networks. This indicates that when choosing a basic network, it is not only important to consider its complexity, but also the matching degree with data. In summary, the faster R-CNN model under VGG network has the best performance. In addition, when the task is sensitive to time, it is also necessary to choose a reasonable



(a) Different network iteration times



(b) Map values of different data sets for each network

Figure 8. Comparison of different network performance

model based on the actual situation. The above experiments compared the selection of the basic framework for the establishment of the faster R-CNN model, and finally selected the VGG basic network. On this basis, the overall performance of the model is further analyzed, and the real data set and synthetic data set are classified to confirm the validity of the synthetic data set used in the study. The study selected Single Shot MultiBox Detector (SSD) and YOLO models for comparative experiments, and obtained the results in Figure 9.

The dataset used in this experiment was divided into ten detection samples and a dataset with 100 artificially synthesized samples added to the temperament. In Figure (8), the performance of the faster R-CNN is the best for both manual and test datasets. In the real dataset, YOLO performs the worst, with an average accuracy of 0.511. The average accuracy of the faster R-CNN is 0.538, which is 0.027 higher than YOLO and 0.013 higher than SSD model. In the manual dataset, faster R-CNN still performs the best, reaching

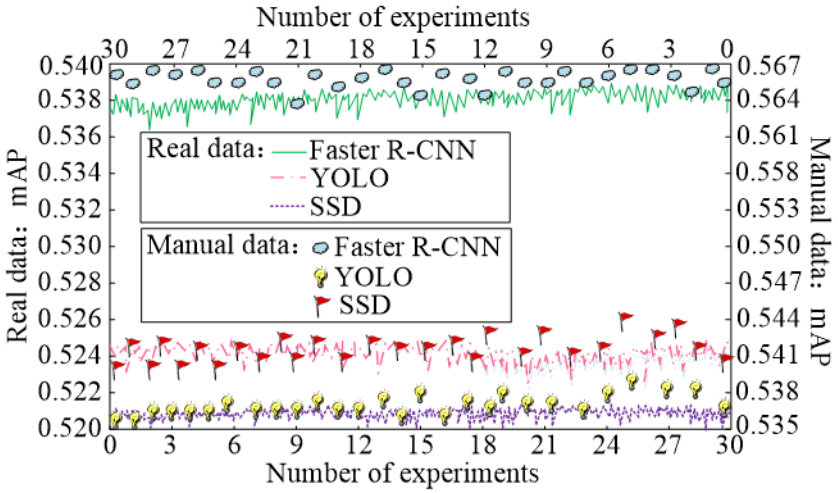


Figure 9. Map index changes of different models in the two data sets

0.569, with an average accuracy of 0.541 for SSD and 0.538 for YOLO, respectively. Compared to these two types of models, the average accuracy of faster R-CNN has been improved by 0.028 and 0.031, respectively. From a horizontal perspective, each model has improved the detection accuracy of artificial data. Faster R-CNN model has improved its average detection accuracy by 3.1 percentage points on artificial dataset, which is better than other models. In real scenes, images may be affected by unknown factors such as illumination intensity, deflection and occlusion. The template change proposed in this study is to solve this problem. Therefore, whether the model can cope with the detection of complex image scenes is the key to judge whether the template transformation is effective. Further experimental analysis was conducted on the differences in detection accuracy caused by different template transformation methods in Figure 10.

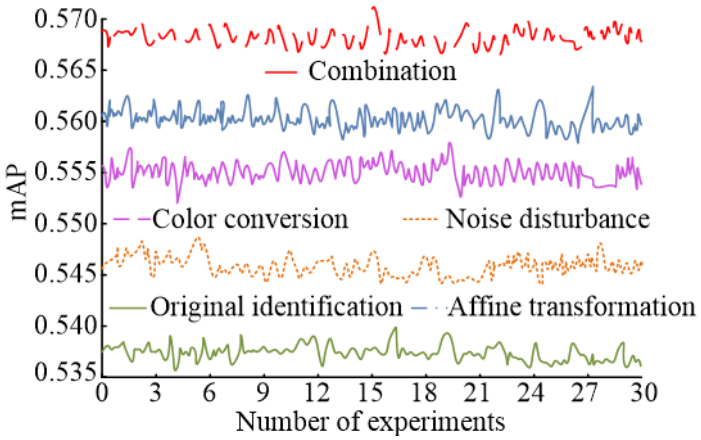


Figure 10. Influence of template transformation on detection accuracy

The original recognition data used in the experiment represent the results obtained in the faster R-CNN model using real datasets. The rest of the template conversion was carried out by controlling variables in scientific control. According to Figure (10), when template transformation is not used, the detection performance of this model is the worst, with an average accuracy of 0.536. The use of template transformation can enhance the detection accuracy of the model to a certain extent. The noise disturbance has the worst effect on improving detection performance, only improving by 0.011. The Affine

transformation and color transformation are better for its improvement effect. Their average accuracy reaches 0.562 and 0.557, respectively. The reason is that in the real world, there are more cases of color and affine changes in objects, with relatively less noise disturbance due to the low number of training sample. Therefore, the corresponding features cannot be fully learned, resulting in poor processing of this aspect. The combined effect of the three models is the best for detection, with an average accuracy of 0.569, which is 3.3% higher than the accuracy value of the original identification.

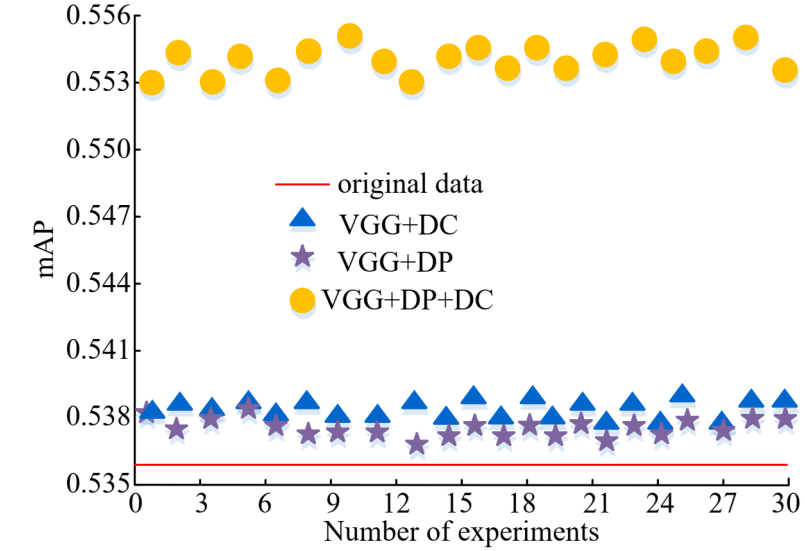
4.2. **Comparative experiments and performance experiments on the selection of various modules in the improved faster R-CNN model.** A scientific control was conducted to further verify the influence of variable convolution module added to the original model on recognizing accuracy. Table 2 shows the specific structure of variable convolution module.

Table 2. Specific structure of VGG 16 network

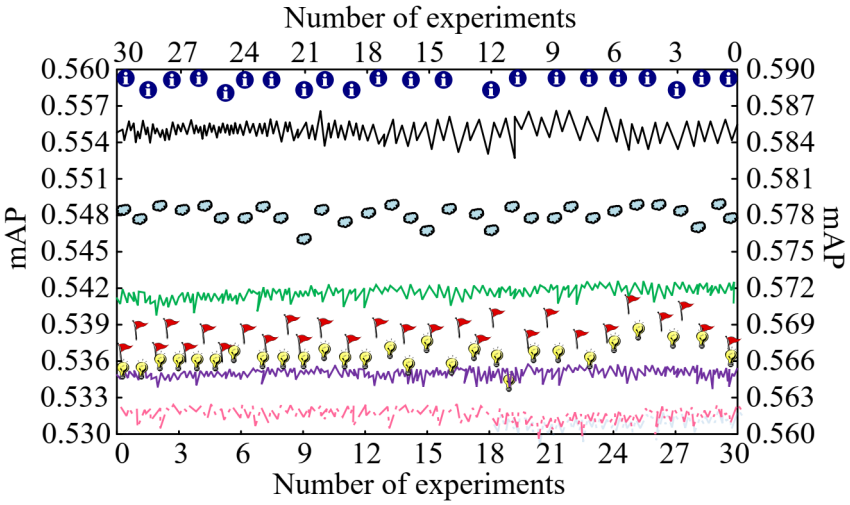
/	Number of layers	Number of convolution kernels	Convolution kernel size	Stride size	Pad size
Group 1	Convolution layer 1	64	3×3×3	1	1
	Convolution layer 2	64	3×3×64	1	1
	Maximum pool layer	1	2×2	2	0
Group 2	Convolution layer 3	128	3×3×64	1	1
	Convolution layer 4	128	3×3×128	1	1
	Maximum pool layer	1	2×2	2	0
Group 3	Convolution layer 5	256	3×3×128	1	1
	Convolution layer 6	256	3×3×256	1	1
	Convolution layer 7	256	3×3×256	1	1
	Maximum pool layer	1	2×2	2	0
Group 4	Convolution layer 8	512	3×3×256	1	1
	Convolution layer 9	512	3×3×512	1	1
	Convolution layer 10	512	3×3×512	1	1
	Maximum pool layer	1	2×2	2	0
Group 5	Convolution layer 11	512	3×3×512	1	1
	Convolution layer 12	512	3×3×512	1	1
	Convolution layer 13	512	3×3×512	1	1
	Maximum pool layer	1	2×2	2	0

The deformable module includes a deformable convolution module and a deformable pooling module. By adding them to VGG, the results in Figure 11 can be obtained.

In Figure 11 (a), the deformable module has a significant improvement in the detection performance of the model. Among them, DC and DP represent deformable convolutional modules and pooling modules, respectively. The improvement effect of the deformable convolutional module is slightly higher than deformable pooling module. The average accuracy of this model with VGG only is 0.535, while the accuracy obtained by adding DC and DP modules reaches 0.548 and 0.546, respectively. Adding two modules at the same time leads to a more significant improvement in model detection accuracy, reaching 0.556. Compared to the original network, it has increased by 1.9 percentage points. Compared to adding two separate modules, it has increased by 0.8% and 1%, respectively. The deformable module can have a positive effect on detection effect. According to Figure 11 (b), the improved algorithm has the best detection performance in both real and artificial



(a) Influence of deformable module on recognition accuracy

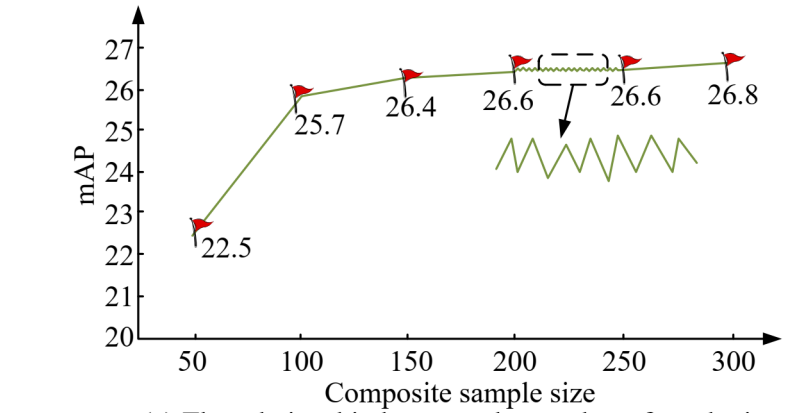


(b) Comparison of recognition accuracy of different network models

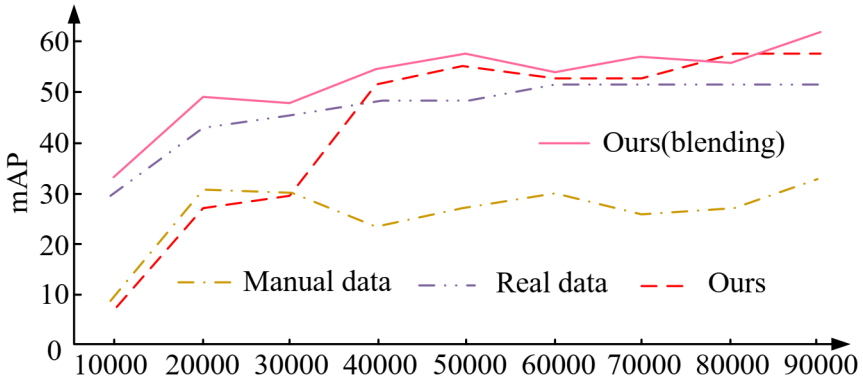
Figure 11. Influence of deformable module on detection accuracy

datasets. From the vertical comparison, the improved algorithm shows an average accuracy improvement of 1.4% and 1.2% compared to the original algorithm in two datasets, respectively. From a horizontal comparison, the accuracy of the improved algorithm in artificially synthesized dataset is improved by 3.5% compared to real dataset. The detection performance of the improved faster R-CNN model has been further improved. Experiments were conducted on the differences in artificial datasets and iterations, leading to variations in detection performance. The results in Figure 12 were obtained.

According to Figure 12 (a), there is a positive correlation between artificial samples and detection accuracy, and the increase in accuracy is quite significant. But when samples reach around 150, their growth rate slows down a lot, or even stops growing. This indicates that relying solely on increasing the amount of manual data does not have a sustained positive guidance effect on the detection effect, which in turn verifies the importance of the stage detection structure. With the increase of the artificial samples, the mAP



(a) The relationship between the number of synthetic samples and the accuracy



(b) The relationship between the number of iterations and the accuracy

Figure 12. Control experiments between the number of synthetic data and the iterations

value of the model reaches 26.8, which has increased by 16.04% compared with the initial value of 22.5. Moreover, the growth rate of the model is faster in the early stage. When the manual samples are 100, the mAP value is increased by 12.45% compared with the samples of 50. According to Figure 12 (b), real sample data are added during 30000 iterations, which shows a significant improvement in accuracy. After 50000 cycles, the rate of improvement gradually flattens out. At the 50,000th iteration, the error between the model value and the real data is 0.58%. Compared with the original value, the error is reduced by 12.41%. Moreover, the mAP value of the model has increased by 44.45% after adding real sample data. This verifies the effectiveness of three-stage identification structure, and a certain number of iterations can achieve the best detection accuracy without the need for additional iterations.

5. **Conclusion.** DL-based computer vision detection technology is currently a popular research trend. This study proposed an optimized faster R-CNN model based on small sample detection, which utilized SSP-Net technology to solve the repeated convolution. At the same time, color transformation modules and deformable modules were introduced in the experiment to further improve it. Finally, the FlickrLogo-32 dataset was selected for performance experiments. The study conducted a scientific control on four basic depth

convolution models, including VGG and ZF. Taking the training set as an example, compared to the other three networks, the VGG network had an average improvement of 0.5% in accuracy. In research design model versus SSD/YOLO model, the average accuracy of the faster R-CNN model has been relatively improved by 3.1%. The recognition accuracy reached 0.569 by adding the transformation module and deformable module, which was improved by 3.3% compared with the original model. The average recognition accuracy of the model was 0.556, a relative improvement of 1.9%. The above experiments have verified the effectiveness of the improved faster R-CNN model in OD. However, in practical applications, there may be a large number of non-rigid transformations. To solve this problem, a more appropriate sample data generation method should be chosen.

Acknowledgment. The research is supported by the Education Department of Hunan Province (project number 20A205); Provincial first-class undergraduate course in 2021, Xiangjiaotong [2021] No. 322932; 2020 Hunan Province New Engineering Research and Practice Project, Xiangjiaotong [2020] No. 9044; The 2020 Innovation and Entrepreneurship Education Center and School Enterprise Cooperation Innovation and Entrepreneurship Education Base Project of Ordinary Universities, Xiangjiaotong [2020] No. 301, 66.

REFERENCES

[1] K.-J. Singh, D.-S. Kapoor, K. Thakur, A. Sharma, and X.-Z. Gao, "Computer-vision based object detection and recognition for service robot in indoor environment," Computers, Materials and Continuum, vol. 17, no. 7, pp. 197-213, 2022.

[2] D. Jana, and S. Nagarajaiah, "Computer vision-based real-time cable tension estimation in Dubrovnik cable: stayed bridge using moving handheld video camera," Structural Control and Health Monitoring, vol. 28, no. 5, pp. 2713-2735, 2021.

[3] D. Jiang, G. Li, C. Tan, L. Huang, and J. Kong, "Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model," Future Generation Computer Systems, vol. 123, no. 1, pp. 94-104, 2021.

[4] X. Cheng, "Visual information quantification for object recognition and retrieval," Chinese Science, vol. 64, no. 12, pp. 2618-2626, 2021.

[5] F.-P. An, J.-E. Liu, and L. Bai, "Object recognition algorithm based on optimized nonlinear activation function-global convolutional neural network," The Visual Computer, vol. 38, no. 2, pp. 541-553, 2022.

[6] Y. Ma, L. Chai, L. Jin, Y. Yu, and J. Yan, "AVS-YOLO: Object detection in aerial visual scene," International Journal of Pattern Recognition and Artificial Intelligence, vol. 36, no. 1, pp. 225004-225004, 2022.

[7] F.-S. Leira, H.-H. Helgesen, T.-A. Johansen, and T.-I. Fossen, "Object detection, recognition, and tracking from UAVs using a thermal camera," Journal of Field Robotics, vol. 38, no. 2, pp. 242-267, 2020.

[8] K.-S. Sujith, and G. Sasikala, "Optimal support vector machine and hybrid tracking model for behavior recognition in highly dense crowd videos," Data Technologies and Applications, vol. 55, no. 1, pp. 19-40, 2020.

[9] D. Yi, J. Su, and W.-H. Chen, "Probabilistic faster R-CNN with stochastic region proposing: towards object detection and recognition in remote sensing imagery," Neurocomputing, vol. 459, no. 1, pp. 290-301, 2021.

[10] A.-A. Rafique, Y.-Y. Ghadi, S.-A. Alsuhibany, S.-A. Chelloug, A. Jalal, and J. Park, "CNN based multi-object segmentation and feature fusion for scene recognition," Computers, Materials and Continuum, vol. 73, no. 3, pp. 4657-4675, 2022.

[11] S. Rajjak, and A.-K. Kureshi, "Multiple-object detection and segmentation based on deep learning in high-resolution video using mask-RCNN," International Journal of Pattern Recognition and Artificial Intelligence, vol. 35, no. 13, pp. 2150038-2150038, 2021.

[12] K. Hoang, A. Pitti, J.-F. Goudou, J.-Y. Dufour, and P. Gaussier, "Active vision: On the relevance of a bio-inspired approach for object detection," Bioinspiration and Biomimetics, vol. 15, no. 2, pp. 025003-0250022, 2020.

[13] H. Qin, Y. Wu, F. Dong, and S. Sun, "Dense sampling and detail enhancement network: Improved small object detection based on dense sampling and detail enhancement," IET Computer Vision, vol. 16, no. 4, pp. 307-316, 2022.
[14] D. Yang, Y. Zhou, W. Shi, D. Wu, and W. Wang, "RD-IOD: Two-level residual-distillation-based triple-network for incremental object detection," ACM Transactions on Multimedia Computing Communications and Applications, vol. 18, no. 1, pp. 18-41, 2022.
[15] M. Fairon, "Double quasi-poisson brackets: fusion and new examples," Algebras and Representation Theory, vol. 24, no. 4, pp. 911-958, 2021.
[16] K. Wang, and M. Liu, "A feature optimized faster regional convolutional neural network for complex background objects detection," IET Image Processing, vol. 15, no. 2, pp. 378-392, 2020.
[17] J. Zhang, K. Wang, Y. He, and L. Kuang, "Visual object tracking via cascaded RPN fusion and coordinate attention," CMES-Computer Modeling in Engineering & Sciences, vol. 132, no. 3, pp. 909-927, 2022.
[18] K.-S. Lim, A.-G. Reidenbach, B.-K. Hua, J.-W. Mason, C.-J. Gerry, P.-A. Clemons, C.-W. Coley, "Machine learning on DNA-encoded library count data using an uncertainty-aware probabilistic loss function," Journal of Chemical Information and Modeling, vol. 62, no. 10, pp. 2316-2331, 2022.
[19] D. Xu, C. Kleineberg, T. VidakovikgOch, and S.-V. Wegner, "Multistimuli sensing adhesion unit for the self-positioning of minimal synthetic cells," Small, vol. 16, no. 35, pp. 2002440-2002448, 2020.
[20] Y. Guo, Z. Mustafaoglu, and D. Koundal, "Spam detection using bidirectional transformers and machine learning classifier algorithms," Journal of Computational and Cognitive Engineering, vol. 2, no. 1, pp. 5-9, 2023.
[21] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," Human-centric Computing and Information Sciences, vol. 9, no. 14, pp. 1-15, 2019.
[22] W.-A. Ke, B. CMC, C. MSH, D. GM, E. SK, and F. SK, "Transfer reinforcement learning-based road object detection in next generation IoT domain," Computer Networks, vol. 193, no. 1, pp. 108078-108078, 2021.