IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 17, 2019, accepted January 30, 2019, date of publication February 4, 2019, date of current version February 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897283

A PSO and BFO-Based Learning Strategy Applied to Faster R-CNN for Object Detection in Autonomous Driving

GANG WANG, JINGMING GUO, YUPENG CHEN, YING LI, AND QIAN XU
[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

Corresponding author: Yupeng Chen (yeppchan@hotmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 61602206.

ABSTRACT Recently outstanding object detection results are achieved by the faster region-based convolutional network (faster R-CNN). Particularly, high-quality detection proposals are obtained by the region proposal network (RPN). Nevertheless, part of the parameters in RPN is assigned by prior knowledge. Therefore the underfitting problem is likely to appear on the training model of RPN. In other words, the generalization ability of RPN is not enough. Increasing parameters is an effective solution to this problem. Thereupon a strengthened RPN (SRPN) is designed to expand the exploring space of RPN. Acquiring the optimal parameter values of SRPN is a non-deterministic polynomial-time hard problem, which can be solved by swarm intelligence algorithms. Thereafter a particle swarm optimization (PSO) and bacterial foraging optimization (BFO)-based learning strategy (PBLS) is introduced to optimize the classifier and loss function of SRPN. In SRPN, a novel multi-level extracting network is created to improve the feature sampling ability. Moreover, the mathematical model of the smooth L_1 loss function is improved to boost the fitting ability. Additionally, support vector machine (SVM) method is applied to enhance the classifier learning capability. PBLS is applied to SRPN (PBLS_SRPN). The parameters of SVM and the improved loss function are optimized by the BFO and PSO methods, respectively. Then, the performance of SRPN is further promoted. The excellent results are obtained by our proposed methods on PASCAL VOC 2007, 2012, MS COCO, and KITTI data sets. Consequently, PBLS_SRPN is effective for object detection in autonomous driving.

INDEX TERMS Deep learning, object detection, region proposal, learning strategy, autonomous driving.

I. INTRODUCTION

Nowadays autonomous driving is a key technique in the automotive industry. Many major car manufacturers and IT companies are working on building autonomous driving systems. The tasks of autonomous driving can be divided into 4 categories which are perception, location, planning and controlling. In other words, the driving scenes are captured by autonomous driving system through the sensors (LiDar, Radar, camera, etc.) to control the driving behaviors. Autonomous driving technique has evolved rapidly. The perception task realizes the detection of the surrounding objects [1], [2]. For example, the pedestrian, bike, car and bus are the important detection contents. Because the surrounding objects can be captured by the camera, therefore the object detection methods for images [3], [4] are the focuses of the current research.

In machine vision field, two kinds of methods are developed for processing the object detection problems. These two categories are hand-engineered feature based methods [5]–[7] and deep learning network based methods [8]–[11]. The famous hand-engineered feature based methods include Scale-Invariant Feature Transform (SIFT) [12], Histograms of Oriented Gradient (HOG) [13] and Deformable Part Models (DPM) [14]. Distinctive invariant features are achieved from images based on the SIFT method. A reliable matching between different views of an object or scene can be performed through SIFT features. Even the gradients or edge

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang.

positions of local object are not known, the appearance and shape of local object still can be characterized rather well via the distribution of local intensity gradients or edge directions in HOG. The HOG or SIFT features are used as image descriptors to search the regions through a class-specific maximum response in DPM.

In recent years, a great success has been got through the deep learning network based methods. Object detection methods Multi-column deep neural networks [15] and Over-Feat [16] have obtained superior improvements in speed and accuracy compared with traditional object detection methods. Because the traditional methods such as DPM and HOG mainly depend on the prior knowledge to design the model, then generalization of these methods is not good. However, deep learning network based methods with the same model can be suitable for various objects detection. Moreover, the deep learning methods with high quality region proposals [17], [18] have achieved excellent localization and classification accuracy. The main stream object detection methods with region proposals are presented as follows.

Recently, Region-based Convolutional Neural Network (R-CNN) [19] has got outstanding detection results. Furthermore the Fast Region-based Convolutional Network (Fast R-CNN) [20] is improved to strengthen the ability of object detection based on R-CNN. The extracting ability is strengthened by using the region of interest (RoI) pooling method on the output features. Moreover, the performance of Fast R-CNN region proposals is promoted by Faster R-CNN [21]. The generation speed of region proposals is accelerated by replacing the Selective Search (SS) [22] with RPN. The RPN is a kind of fully convolutional network (FCN) [23]. RPN can be trained end-to-end for generating detection region proposals.

At present, excellent results are achieved by the deep learning network based methods. Most of the parameters in these methods are defined by the prior knowledge. The parameters can be divided into 2 categories. The first category is deep learning network related parameters. The training process of the deep learning network is mainly controlled by these parameters. The second category is object detection method related parameters. The processing results are decided by these parameters. In order to promote the performance of processing results, these parameters need to be learned.

Four problems are not settled in the mentioned above object detection methods. Firstly, the output feature maps of convolutional network are used to generate region proposals in RPN. Nevertheless, the multi-level convolutional information is not contained in the output feature maps. Consequently, the performance of small object detection for Faster R-CNN is seriously affected. Secondly, the mathematical model of smooth L_1 loss function is not reasonably developed. Therefore, the fitting capability of smooth L_1 loss function is poor. Thirdly, the probability of each class is calculated by the softmax method to classify the objects. The softmax method is suitable for solving multiple classification problems. However, the classifier is applied to distinguish the positive and negative anchor boxes in RPN. Obviously, these are binary classification problems. Additionally, the learning ability of softmax is poor. Thereupon, softmax method is not our optimal choice. Fourthly, the parameters of RPN network are designed by the prior knowledge. However, the parameters of model should be adaptive to the training samples in the machine learning filed. Thereafter underfitting problem is likely to appear on the training model of RPN. That is to say, the generalization ability of RPN is not enough.

In this paper, a novel learning strategy is developed. Five improvements are designed in PBLS_SRPN. Firstly, a novel multi-level extracting network (MLEN) is created in this work. The lower feature maps and the higher feature maps are integrated with the output features in MLEN. In this way, the effect of feature sampling method in our MLEN is enhanced. Secondly, the mathematical model of smooth L_1 loss function is improved in PBLS_SRPN. Therefore, the fitting ability of smooth L_1 is strengthened. Thirdly, because support vector machine (SVM) [24] method is more suitable for solving binary classification problems. Besides, the parameter number of SVM is more than softmax. Thereupon the learning ability of SVM is better. As a result the softmax method is replaced by SVM method to accomplish the classification task. Fourthly, BFO is applied to optimize the parameters of the SVM, termed BFO-SVM. Specially, the improved swarming equation is proposed to promote the optimization effect of BFO. Consequently, the classification performance of PBLS_SRPN is enhanced by using BFO-SVM. Fifthly, the parameters of RPN network and the improved loss function are optimized by the PSO method. Thereupon, the generalization ability of RPN is boosted.

Section 2 shows the related work to our PBLS_SRPN. Section 3 presents the basic concept of Faster R-CNN. Section 4 describes the improvements of PBLS_SRPN method. Section 5 shows the experiment results and discussions. Finally, section 6 draws some conclusions for this paper.

II. RELATED WORK

Several state of the art deep leaning based methods [25]–[27] are introduced in this chapter. R-CNN is developed to improve the object detection mean average precision (mAP). The relationship between image classification and object detection is created by R-CNN which contains three stages for object detection. Firstly, SS method is used to generate around 2k region proposals. Secondly, convoluntional features of each region proposal are extracted based on pre-trained CNN [28], [29]. Thirdly, SVM is applied to classify the convoluntional features. However, computation time is increased by using these steps.

A bounding box regression scheme is applied to improve the object detection results in Multi-region CNN (MR-CNN) [30]. In order to strengthen the detection results, the bounding boxes are evaluated twice. In addition, the semantic segmentation-aware features are extracted through multi-region deep CNN. However, the region proposals generation method is not optimal. Therefore, the calculation speed is the computing bottle neck of MR-CNN method. Both inside and outside the region of interest information is exploited in the object detector Inside-Outside Net (ION) [31]. The information of features is extracted based on skip pooling [32] at multiple levels of abstraction and scales. The region proposals generation effect is need to be upgraded.

The object detection and handling region proposal generation tasks are jointly developed in HyperNet [33]. The object detection results are good on PASCAL VOC 2007 and 2012 based on HyperNet. However, the loss function of HyperNet is not optimal. A vision-based method FRCNN+Or [34] is proposed by building upon a deep convolutional neural network that can reason simultaneously about the location of objects in the image and their orientations on the ground plane. The same set of convolutional layers is used for the different tasks involved, avoiding the repetition of computations over the same image. The goal of Mono3D [35] is to perform 3D object detection from a single monocular image in the domain of autonomous driving. The focus of this method is on proposal generation. The results of Mono3D are significantly better than other monocular approaches. A regionlet [36] is a base feature extraction region defined proportionally to a detection window at an arbitrary resolution. These regionlets are organized in small groups with stable relative positions to delineate fine-grained spatial layouts inside objects. The worlds of 3D and 2D object is connected in DPM-VOC + VP [37] by building an object detector which leverages the expressive power of 3D object representations while at the same time can be robustly matched to image evidence. This method provides consistently better joint object localization and viewpoint estimation than the state-of-the-art multi-view and 3D object detectors on various benchmarks. Our novel PBLS_SRPN is compared with the methods mentioned above. We can find that the performance of our proposed PBLS_SRPN is excellent in the experiment results.

III. LITERATURE REVIEW

A. FASTER R-CNN

Faster R-CNN method is a state of the art object detection method. The basic knowledge is introduced in this section. The convolutional features of image are processed by RPN. Next, the region proposals are generated as output of the RPN. Moreover, the object score is computed for each region proposal. A sliding window [38] is applied to the output features of top-level convolutional layer to generate reference region proposals. Therefore, the quality of region proposals is decided by the top-level features. Specially, these features are sent to the box-regression layer (*reg*) and the box-classification layer (*cls*) for information processing. In other words, the framework of RPN is accomplished by a $n \times n$ convolutional layer and two 1×1 convolutional layers.

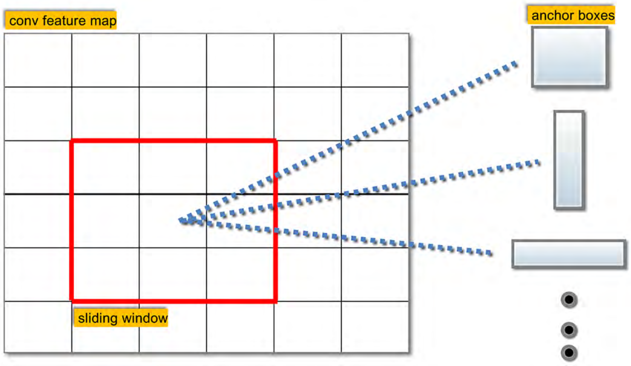


FIGURE 1. The structure of anchor boxes.

Multiple region proposals are predicted at each sliding-window location simultaneously. We assume that the maximum possible proposals for each location are k. Therefore the $4k$ outputs of *reg* layer encode the coordinates of k rectangular proposals, and the $2k$ scores of *cls* layer are outputted. Particularly, the probability of object or not object for each proposal is estimated by every score. Moreover, an anchor is adjusted by the scale and aspect ratio. The anchor is centered at the sliding window (Figure 1). We presume that the size of a convolutional feature map is $W \times H$, and then the total number of anchors is WHk. In order to diminish the cost of computation, the anchor-based method is built on a pyramid of anchors. Multiple scales and aspect ratios are applied to classification and bounding box regression.

The expression for the loss function in RPN is presented in (1).

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

where i represents the number for an anchor box and p_i means the predicted probability for the ith anchor box. The value of p_i^* is 0 if the label of anchor box is negative, while if the label of anchor box is positive when the value of p_i^* is 1. Variable t_i represents the computation results for the predicted bounding box. Item t_i^* means the ground-truth box for a positive anchor box. The variable N_{cls} is the mini-batch size. The parameter N_{reg} is the number of anchor boxes. The term λ represents a balancing factor. The loss function L_{cls} is a log loss for classifying object or not object. The formulation of L_{cls} is showed as follows:

$$L_{cls}\left(p_i, p_i^*\right) = \begin{cases} -\log p_i & \text{if } p_i^* = 1 \\ -\log(1-p_i) & \text{if } p_i^* = 0 \end{cases} \quad (2)$$

The loss function L_{reg} is described as follows:

$$L_{reg}\left(t_{i,} t_i^*\right) = R\left(t_i - t_i^*\right) \quad (3)$$

where R represents the robust loss function (smooth L_1). Variable $p_i^* L_{reg}$ represents that the expression $\lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}$

$\left(t_i, t_i^*\right)$ is valid when the anchor boxes are positive $\left(p_i^* = 1\right)$. The outputs of the *reg* and *cls* layers contain $\{t_i\}$ and $\{p_i\}$. The parameters of regression expression are showed as follows:

$$\begin{aligned} t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a \\ t_w &= \log(w/w_a), t_h = \log(h/h_a) \\ t_x^* &= \left(x^* - x_a\right)/w_a, t_y^* = \left(y^* - y_a\right)/h_a \\ t_w^* &= \log\left(w^*/w_a\right), t_h^* = \log\left(h^*/h_a\right) \end{aligned} \tag{4}$$

where x, y, w and h represent center coordinates, width and height for region proposals. Parameters x_a, x and x^* are defined as the anchor box, predicted box and groundtruth box.

The main methods for training the RPN are back propagation (BP) and stochastic gradient descent (SGD). The ''image-centric'' is used as the training sampling strategy in the RPN. In each mini-batch of a single image, a lot of positive and negative anchor boxes are generated.

The number of negative samples is more than the number of positive samples, thus part of the anchor boxes is not used to train the network. The loss function is computed by 256 anchor boxes from a mini-batch. Moreover, the number of positive samples is the same as that of negative samples. If the number of positive anchor boxes is less than 128, then the negative anchor boxes are used as the positive ones. The layers of RPN are initialized by using the pre-trained ImageNet [29], [39] model.

B. SOFTMAX CLASSIFIER

In convolutional neural network, the softmax classifier [40] is used frequently. The general expression for logistic regression is softmax which can solve multi-class classification problems. For each value of $j = 1, \ldots, k$, the probability $P(y = j|x)$ is estimated based on our hypothesis for a test input x.Therefore a k dimensional vector is calculated through our hypothesis with k estimated probabilities. The formulation of $h_\theta(x)$ is showed as follows:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(j)} = 1 \,|x^{(i)}; \theta) \\ p(y^{(j)} = 2 \,|x^{(i)}; \theta) \\ \cdot \\ \cdot \\ \cdot \\ p(y^{(j)} = k \,|x^{(i)}; \theta) \end{bmatrix} \tag{5}$$

The above equation is transformed as follows:

$$h_\theta(x^{(i)}) = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \cdot \\ \cdot \\ \cdot \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{6}$$

where $\theta_1, \theta_2, \ldots, \theta_k \in R^{n+1}$ are the factors of our model. Because the each item is multiplied by $\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}$, thus each item is normalized.

C. INTERSECTION-OVER-UNION

If an anchor box is a positive example, then this anchor box overlaps an object a lot. Contrarily an anchor box is a negative example, thus this anchor box covers an object a little. Nevertheless if an anchor box partially covers an object, thereupon it is hard to evaluate an anchor box. Intersection-over-Union (IoU) [41] measure is developed to settle this problem. The definition of IoU is $(w \cap b)/(w \cup b)$. The w and b present the anchor boxes and ground truth boxes. The grid search method is applied to set the value of IoU for distinguishing the positive and negative samples.

D. PARTICLE SWARM OPTIMIZATION

The PSO method is one of the optimization methods [42]. This method simulates the social behavior of birds. The mathematical formulation of PSO is presented as follows. In a N-dimensional region, the ith particle is $x_i = (x_{i1}, \ldots, x_{in}, \ldots, x_{iN})$. The best position for the ith particle is $p_i = (p_{i1}, \ldots, p_{in}, \ldots, p_{iN})$, which obtains the optimal fitness result and is named as *pbest*. In particles, the symbol *gbest*represents the global best results among all searching particles. The expression $v_i = (v_{i1}, \ldots, v_{in}, \ldots, v_{iN})$ represents the velocity of the ith particle. The update strategy for the velocity and position of $t + 1$ iteration particle are illustrated in (7).

$$\begin{aligned} v_{in}(t+1) &= wv_{in}(t) + c_1 r_1(p_{in} - x_{in}(t)) \\ &\quad + c_2 r_2(p_{gn} - x_{in}(t)) \\ x_{in}(t+1) &= x_{in}(t) + v_{in}(t+1) \end{aligned} \tag{7}$$

where t means the iteration number; the $v_{in}(t)$ represents the velocity of the ith particle on the nth dimension in tth iteration; factors r_1 and r_2 are random values in [0, 1]; the weight w is the inertia coefficient; the variables c_1 and c_2 are learning rates. When the fitness threshold or the maximum iteration number is satisfied, then the iterations of PSO method stop.

E. SUPPORT VECTOR MACHINE

The SVM method is a famous classifier in machine learning filed. Moreover, a deeply theoretical basis is contained in SVM. Simultaneously, global optimal solutions can be found by SVM with a small amount of training samples. SVM method is widely used in object detection, object classification, non-linear regression and pattern recognition.

A linear model is applied to generate the non-linear class boundaries by taking some non-linear mapping input vectors into a multi-dimensional space. In the multi-dimensional space, an optimal separating hyper plane is created. Therefore the advantage of SVM is to search a maximum hyper plane which separates the decision classes. Consequently, a non-linear relationship is discovered by SVM between the inputs and outputs in multi-dimensional space.

Linear kernel function (8), sigmoid kernel function (9), the RBF kernel function (10) and polynomial kernel function (11) are showed as follows:

$$\mathrm{K}\left(x_i, x_j\right) = x_i^T x_j \tag{8}$$

$$\mathrm{K}\left(x_i, x_j\right) = \tanh\left(\gamma x_i^T x_j + r\right) \tag{9}$$

$$\mathrm{K}\left(x_i, x_j\right) = \exp\left(-\gamma \parallel x_i - x_j \parallel^2\right), \gamma > 0 \tag{10}$$

$$\mathrm{K}\left(x_i, x_j\right) = \left(\gamma x_i^T x_j + r\right)^d, \gamma > 0 \tag{11}$$

where d, r and γ are kernel parameters. In order to promote the performance of anchor boxes, the SVM classifier is applied to replace the softmax classifier in RPN.

F. BFO ALGORITHM

The *E.coli* foraging process is simulated by BFO [43] algorithm which contains chemotaxis, swarming, reproduction and elimination-dispersal 4 step operations. The *E.coli* bacteria are showed in Figure 2 by using microscope.

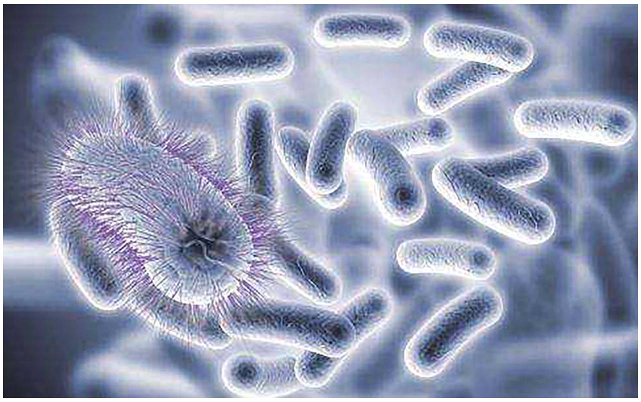


FIGURE 2. ***E.coli* bacteria.**

Bacterial foraging strategy is simulated by this phase. Firstly, the direction for bacteria is changed for a period of time based on tumbling. Next, a short distance is moved by bacteria. The bacteria will keep swimming when the bacteria find rich nutrients. We suppose θ is the position of a bacterium and $\theta^i(j, k, l)$ means the ith bacterium in the jth chemotaxis, kth reproduction, lth elimination-dispersal procedure. The expression for tumbling is presented as follows:

$$\varphi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}} \tag{12}$$

where $\Delta(i), i = 1, 2, \ldots, S$ is a random variable. The range for the element $\Delta_m(i), m = 1, 2, \ldots, p$ in $\Delta(i)$ is $[-1, 1]$. The parameter$S$ means the amount of bacteria. The formulation for bacterial position updating is described as follows:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\varphi(i) \tag{13}$$

where $C(i), i = 1, 2, \ldots, S$ is a distance for moving during the swimming phase.

The communication between bacteria is simulated by swarming process. The chemical substances are released by bacteria to attract other bacteria when they find high amounts of nutrients. The bacteria will repel each other when they are in danger. The swarming action is described as follows:

$$\begin{aligned} &Jcc(\theta, \theta^i(j,k,l)) \\ &= \sum_{i=1}^{S}\left[-d_{\text{attract}} \exp\left(-w_{\text{attract}} \sum_{f=1}^{p}(\theta_f - \theta_f^i)^2\right)\right] \\ &\quad + \sum_{i=1}^{S}\left[h_{\text{repellant}} \exp\left(-w_{\text{repellant}} \sum_{f=1}^{p}(\theta_f - \theta_f^i)^2\right)\right] \end{aligned} \tag{14}$$

where $\theta = [\theta_1, \ldots, \theta_p]^T$ is a bacterium in the swarming stage. The variable θ_f^i is the f th component of the ith bacterium position θ^i. The communication value $Jcc(\theta, \theta^i(j,k,l))$ between bacteria is added to the fitness function result in the chemotaxis phase j; Variable p represents the amount of problem dimension; Parameter S means the number of bacteria; Factors $d_{attract}$, $w_{attract}$, $h_{repellant}$ and $w_{repellant}$ are the attraction or repulsion force. The fitness result for ith bacterium is presented as follows:

$$J(i,j,k,l) = J(i,j,k,l) + Jcc(\theta, \theta^i(j,k,l)) \tag{15}$$

The reproduction process is executed after N_c chemotactic operations. The variable S is assumed as a positive even integer. The number of bacteria population with sufficient nutrients is S_r. These bacteria are reproduced with no mutations.

$$S_r = \frac{S}{2} \tag{16}$$

The accumulated cost is described by the health of a bacterium. If the nutrients of a bacterium are decreased, then the value of accumulated cost is increased. In other words the bacterium cannot reproduce if the bacterium is not health. The bacteria are sorted through descending order based on their health in this phase. The S_r least healthy bacteria are eliminated. In addition, the other S_r healthiest bacteria are reproduced.

The rising temperature may kill a lot of bacteria. In other words, the adverse environment can cause the death of bacteria. This process is expressed through dispersing some bacteria with a small probability P_{ed}. Moreover, some randomly generated bacteria are applied to replace the death ones.

IV. OUR APPROACH

A. OVERVIEW

In this paper, the convolutional layers of VGG-16 [44] are represented as Conv1, Conv2, Conv3, Conv4 and Conv5. The features of input image are extracted by using the pre-trained VGG-16 model.

Five improvements are presented in this section. In section B, a novel multi-level extracting network is proposed to strengthen the output features of VGG-16. In section C, the mathematical model of smooth L_1 loss function is improved. In section D, the softmax method is

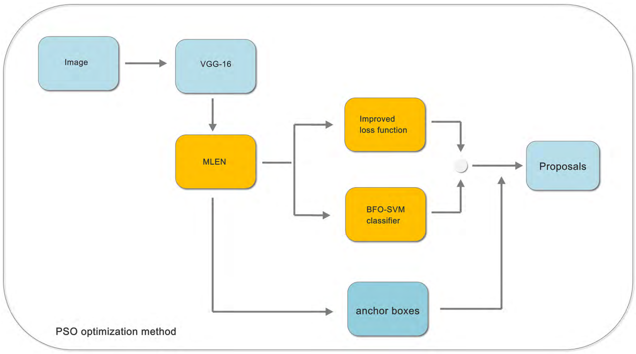


FIGURE 3. The framework of PBLS_SRPN network.

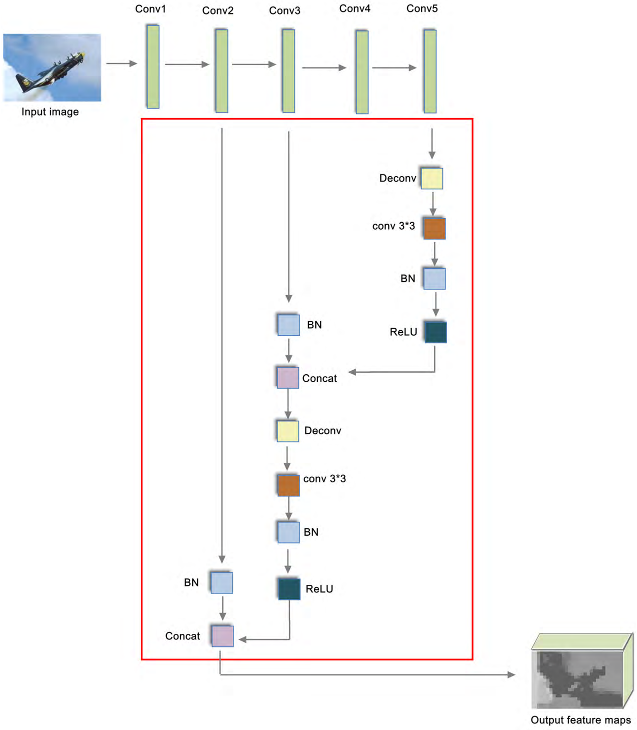


FIGURE 4. The framework of multi-level extracting network.

replaced by SVM method to accomplish the classification task. In section E, BFO is applied to optimize the parameters of the SVM, termed BFO-SVM. In section F, the parameters of RPN network and the improved loss function are optimized by the PSO method.

B. A NOVEL MULTI-LEVEL EXTRACTING NETWORK

In RPN, the output feature maps of the VGG-16 are used to generate region proposals. Nevertheless, the semantic information of output feature maps is poor. The reason is that the highly semantic information is not contained in the output feature maps of the VGG-16. Particularly, the rich information of small objects is kept in the lower level feature maps. Consequently, the performance of small object detection for Faster R-CNN is seriously affected. A novel multi-level extracting network (MLEN) is proposed in this paper to solve the problems mentioned above. The framework of MLEN is illustrated in Figure 4.

From Figure 4 we can see that the features of multi-level are extracted in our MLEN. Because the size of feature maps in lower level is large and the semantic information of feature maps in higher level is rich, then the output features combining with lower and higher level features are strengthened in our MLEN. Additionally, different strategies are applied to different level convolutional layers. Deconvolutional layer, 3×3 convolutional layer, batch normalization layer and concatenation layer are used to sample different level features. At the end, the lower feature maps and the higher feature maps are compressed into a uniform space. In this way, the effect of feature sampling method of our MLEN is enhanced. Because the features of Conv1 are large, then these features are not contained in our MLEN.

C. IMPROVED LOSS FUNCTION

From (3) we can see that the regression loss function is $R\left(t_i - t_i^*\right)$. The definition of robust loss function (smooth L_1) is presented as follows:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \textit{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise}, \end{cases} \tag{17}$$

The smooth L_1 loss function is designed through the integration of L_1 loss and L_2 loss. The equations of L_1 loss and L_2 loss are showed as follows:

$$\begin{aligned} L_1(x) &= \alpha\,|x| \\ L_2(x) &= \beta x^2 \end{aligned} \tag{18}$$

where α and β are the influence factors. The graph of L_1 loss and L_2 loss is presented as follows:

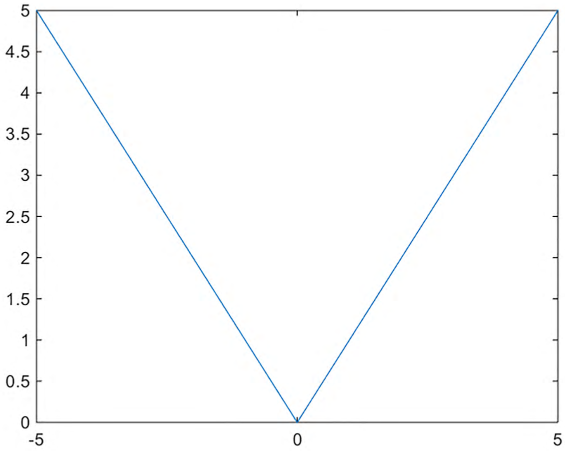


FIGURE 5. The graph of L_1 loss.

For convenience, the value of α and β is set to 1 in Figure 5 and Figure 6. We can see that the descending speed of L_1 loss is faster than that of L_2 loss. In other words, the ability of convergence for L_1 loss is better than L_2 loss. However the complexity of L_1 loss is less than L_2 loss. Therefore, L_1 loss that is less sensitive to outliers than the L_2 loss.

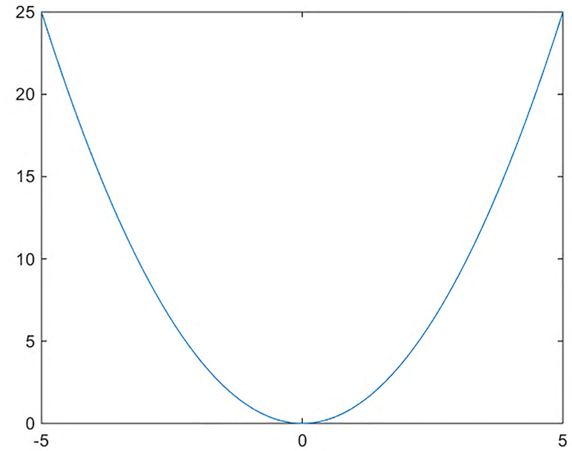


FIGURE 6. The graph of L_2loss.

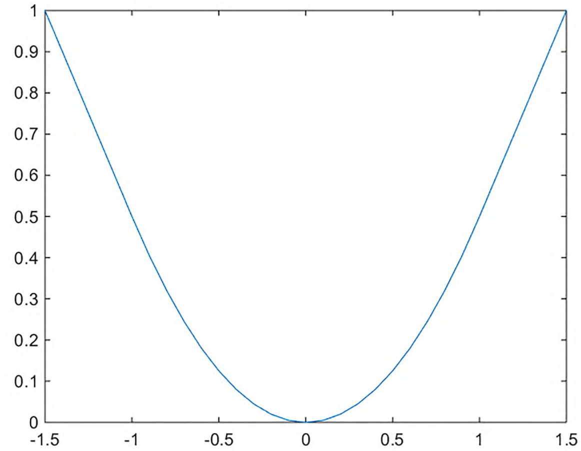


FIGURE 7. The graph of smooth L_1loss.

From Figure 7 we can see that smooth L_1 is the L_2 loss in the range $(-1, 1)$, while smooth L_1 is the L_1 loss in the rest of range. In other words, the smooth L_1 contains both the advantage of L_1 loss and L_2 loss. Specially, the smooth L_1 is the special case of Huber loss. The definition of Huber loss is presented as follows:

$$H(x) = \begin{cases} 0.5x^2 & \mathit{if}\ |x| < \alpha \\ \alpha(|x| - 0.5\alpha) & \text{otherwise}, \end{cases} \tag{19}$$

From (19) we can find that if the value of x is in the range $[-\alpha, \alpha]$, then the L_2 loss plays a major role. In other words, the fitting ability of Huber loss is strengthened. But the Huber loss is more sensitive to outliers than before. In contrary, L_1 loss occupies a leading position in Huber loss. The descending speed of Huber loss is enhanced. Nevertheless, the under fitting problems are prone to occur. Because the (19) only contains the linear function and quadratic function, thus the fitting ability is not enough. Specially, the bounding box regression problems are complexity in RPN, therefore (19) need to be improved.

$$H(x) = \begin{cases} 0.5(x^2 + x^4) & \mathit{if}\ |x| < \alpha \\ \alpha(|x| - 0.5\alpha) & \text{otherwise}, \end{cases} \tag{20}$$

In (20), the square of L_2 loss is added to promote the ability of fitting. In this way, the performance of L_2 loss is strengthened in range $(-\alpha, \alpha)$. In addition, the L_1 loss is kept in (20).

$$H(x) = \begin{cases} \beta x^2 + \theta x^4 & \mathit{if}\ |x| < \alpha \\ \alpha(|x| - \beta\alpha) & \text{otherwise}, \end{cases} \tag{21}$$

Moreover, the value 0.5 is replaced by variable β. The reason is that the 0.5 may not be the optimal value. Additionally, variable θ is set as the influence factor of x^4. As a result, our improved (21) includes the advantage of L_1 loss and L_2 loss. Specially, the fitting ability is improved by introducing the variable x^4. The variablex^4 is introduced to the smooth L_1 loss function according to our experience. From the experimental results we can see this improvement is effective. Simultaneously, variables α, β and θ can be set by the optimization method. In other words, the fitting ability of smooth L_1 loss function is promoted.

D. SOFTMAX CLASSIFIER IS REPLACED BY SVM CLASSIFIER

Softmax classifier is applied to RPN of Faster R-CNN method. The expression of softmax classifier $h_\theta(x)$ is presented in (6). Each probability of class label is achieved based on corresponding estimated value. The diagram of softmax classifier is presented in Figure 8.

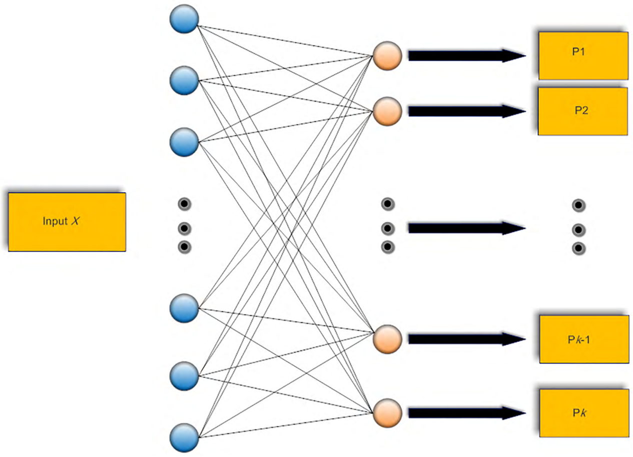


FIGURE 8. The diagram of softmax classifier.

From (6) we can see that the softmax classifier is mainly used for multi-class classification problems. However, RPN is applied to distinguish the foreground region boxes and the background region boxes. In other words, the classification task of RPN is a binary classification problem. In order to solve binary classification problem, the (6) is transformed to keep two items. The modified expression is presented as follows:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 \,|x^{(i)}; \theta\,) \\ p(y^{(i)} = 2 \,|x^{(i)}; \theta\,) \end{bmatrix} \tag{22}$$

$$=> h_\theta(x^{(i)}) = \frac{1}{\sum_{j=1}^{2} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \end{bmatrix} \tag{23}$$

The item $\sum_{j=1}^{2} e^{\theta_j^T x^{(i)}}$ equals to $e^{\theta_1^T x^{(i)}} + e^{\theta_2^T x^{(i)}}$, then the (23) is changed as follows:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 \,|x^{(i)}; \theta\,) \\ p(y^{(i)} = 2 \,|x^{(i)}; \theta\,) \end{bmatrix} \tag{24}$$

$$=> h_\theta(x^{(i)}) = \begin{bmatrix} \dfrac{e^{\theta_1^T x^{(i)}}}{e^{\theta_1^T x^{(i)}} + e^{\theta_2^T x^{(i)}}} \\ \dfrac{e^{\theta_2^T x^{(i)}}}{e^{\theta_1^T x^{(i)}} + e^{\theta_2^T x^{(i)}}} \end{bmatrix} \tag{25}$$

Furthermore, the (25) is modified as follows:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 \,|x^{(i)}; \theta\,) \\ p(y^{(i)} = 2 \,|x^{(i)}; \theta\,) \end{bmatrix} \tag{26}$$

$$=> h_\theta(x^{(i)}) = \begin{bmatrix} \dfrac{1}{1 + e^{-(\theta_1^T - \theta_2^T)x^{(i)}}} \\ 1 - \dfrac{1}{1 + e^{-(\theta_1^T - \theta_2^T)x^{(i)}}} \end{bmatrix} \tag{27}$$

Moreover, variable θ^T is used to represent $\theta_1^T - \theta_2^T$. Therefore the following expression is achieved.

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 \,|x^{(i)}; \theta\,) \\ p(y^{(i)} = 2 \,|x^{(i)}; \theta\,) \end{bmatrix} \tag{28}$$

$$=> h_\theta(x^{(i)}) = \begin{bmatrix} \dfrac{1}{1 + e^{-\theta^T x^{(i)}}} \\ 1 - \dfrac{1}{1 + e^{-\theta^T x^{(i)}}} \end{bmatrix} \tag{29}$$

We can see that the (29) is the formulation of classic logistic regression classifier. In other words, the softmax classifier is the extension of logistic regression classifier in the area of multi-class classification. However, the fitting ability of logistic regression classifier is not optimal. This issue can lead to poor classification results. In order to solve this problem, SVM classifier with RBF kernel function is introduced to RPN for classifying the foreground region boxes and the background region boxes in this paper. Because RBF kernel function can map samples to a higher dimensional space, therefore the fitting ability of RBF kernel function is better than logistic regression classifier. The parameters of RBF kernel function are C and γ. Thereafter the classifying time and training time of SVM with RBF kernel function is not large. Specially, the classifying time is very important for autonomous driving system. Consequently, RBF kernel function is a good choice for our proposed method.

E. A NOVEL BFO-SVM CLASSIFIER

In order to achieve the outstanding classifying ability, the parameters C and γ of RBF kernel function need to be trained. The penalty parameters C and γ can influence the classification effect of SVM. The classification accuracy is strengthened with the increase of parameters C and γ. However if the parameters C and γ are too large, the overfitting problem may appear in SVM. Vice versa, the underfitting problem can occur if the values of C and γ are too small. In a word, the appropriate values of parameters C and γ are very important.

Conventionally, the values of parameters C and γ are found through the grid search method. However, the length of searching step is difficult to define. On the one hand if the length of searching step is large, then the classifying results are not stable. In other words, the exploitation ability is not good. On the other hand if the length of searching step is small, thus the classifying results may fall into local optima easily. In this situation, the exploring ability of SVM is insufficient.

In this paper, the BFO method is applied to optimize the parameters of C and γ. BFO algorithm is a kind of swarming intelligent method which has a satisfactory performance in solving the optimization problem based on the design of chemotaxis, swarming, reproduction and elimination-dispersal operations. The structure of each bacterium is designed for applying BFO to optimize SVM. First, each bacterium means a set of solution for C and γ. In other words, the searching of best bacterium is to find the best C and γ. Second, the bacterium has two dimensions which represent two parameters C and γ. Third, the capability of each bacterium is evaluated by the fitness function. The values of C and γ are updated based on the fitness results.

Additionally, the swarming characteristic of *E.coli* bacteria is described in the BFO. However, the original swarming equation cannot describe cell to cell attraction-repulsion relationship accurately when bacteria gathered at the same point. The improved swarming equation (30) is proposed in ISEDBFO [29] to overcome the above problems. This formula is not only more suitable for describing the behavior of bacteria, but also to improve the efficiency of searching optimal value.

$$\begin{aligned} &Jcc\left(\theta, \theta^i\,(j,k,l)\right) \\ &\quad = \sum_{i=1}^{S} \left[d_{attract} \tanh\left(w_{attract} ||\theta - \theta^i||_2 \right) \right] \\ &\qquad - \sum_{i=1}^{S} \left[h_{repellant} \exp\left(-w_{repellant} ||\theta - \theta^i||_2 \right) \right] \end{aligned} \tag{30}$$

where $Jcc(\theta, \theta^i(j,k,l))$ is the cell to cell communication value which is added to the result of fitness function in the chemotaxis phase j; S is the number of bacteria; $d_{attract}$, $w_{attract}$, $h_{repellant}$, $w_{repellant}$ are different factors which represent the strength of attraction and repulsion. In order to limit the swarming effect in a reasonable range, the original equation $\sum_{f=1}^{p} (\theta_f - \theta_f^i)^2$ is replaced by formula $||\theta - \theta^{\mathrm{i}}||_2$.

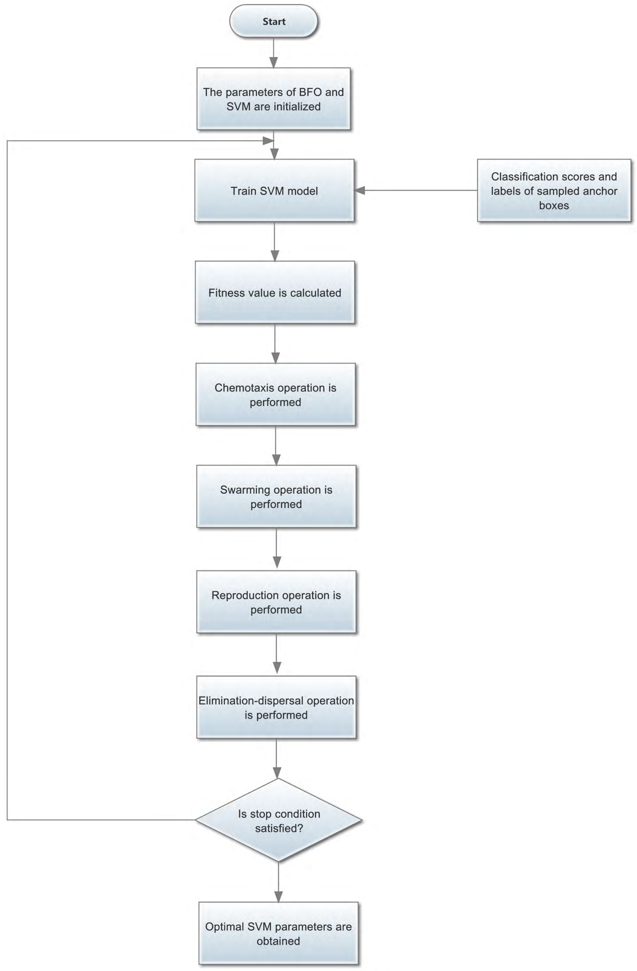

Start
The parameters of BFO and SVM are initialized
Train SVM model
Classification scores and labels of sampled anchor boxes
Fitness value is calculated
Chemotaxis operation is performed
Swarming operation is performed
Reproduction operation is performed
Elimination-dispersal operation is performed
Is stop condition satisfied?
Optimal SVM parameters are obtained

FIGURE 9. The process of SVM parameters optimization through BFO.

At the same time, hyperbolic function *tanh*(x) is introduced to express attraction part more accurately. In this paper, this improved swarming equation is applied to our BFO-SVM classifier. Therefore, the optimization effect is further enhanced. The process of SVM parameters optimization through BFO is showed in Figure 9.

From Figure 9 we can see that the parameters of BFO and SVM are initialized at the start of SRPN. The anchor boxes are randomly selected from the input image to train the model of SVM. Simultaneously, the proportion of positive samples and negative samples is close to 1:1. Because the number of negative samples is always more than the number of positive samples, thereafter part of positive samples is padded with some negative samples. Moreover, the chemotaxis, swarming, reproduction and elimination-dispersal operations are performed separately. As a result, the bacterium with poor fitness would be changed. If the fitness value satisfies our requirement or the iteration number of BFO is reached, then the optimal parameters C and γ are achieved. The range of parameters C and γ is [0.01, 35000] and [0.0001, 32] respectively.

At the beginning of SVM optimization, the values of C and γ are randomly initialized. Thereafter the iteration times of BFO should be relatively large. After many optimization steps, the value of parameters C and γ tend to be stable. Therefore, the number of iterations should be reduced at this time. In other words, iteration times of SRPN should be gradually reduced. The following expression shows the iteration times of BFO.

$$n_{\text{iteration\_BFO}} = N_{\text{iteration\_BFO\_total}} \times \sin\left(\frac{M_{\text{iteration\_SRPN\_total}} - m_{\text{iteration\_SRPN}}}{M_{\text{iteration\_SRPN\_total}}} \times \frac{\pi}{2}\right) \quad (31)$$

where $n_{\text{iteration\_BFO}}$ is the BFO iteration times for one cycle of computation of SRPN; variable $N_{\text{iteration\_BFO\_total}}$ is the total iteration times of BFO; variable $m_{\text{iteration\_SRPN}}$ is the current iteration times of SRPN; variable $N_{\text{iteration\_SRPN\_total}}$ is the total iteration times of SRPN. The curve of (31) is presented in Figure 10.

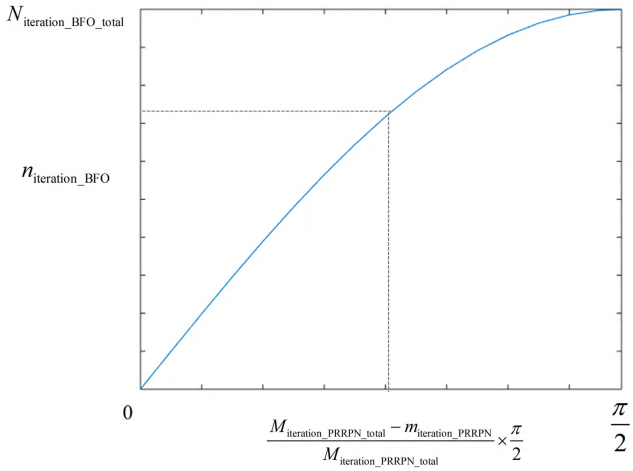


FIGURE 10. The BFO iteration number.

From (31) and Figure 10, we can find that the $m_{\text{iteration\_SRPN}} = 1$ at the start of SRPN. Then the value of $\frac{M_{\text{iteration\_SRPN\_total}} - m_{\text{iteration\_SRPN}}}{M_{\text{iteration\_SRPN\_total}}} \times \frac{\pi}{2}$ is closed to $\frac{\pi}{2}$. In other words, the variable $n_{\text{iteration\_BFO}}$ approximately equals to $N_{\text{iteration\_BFO\_total}}$. The value of $n_{\text{iteration\_BFO}}$ is gradually decreased while the value of $m_{\text{iteration\_SRPN}}$ increases. As a result, the calculation training time of BFO-SVM is effectively diminished. Therefore the (31) satisfies our requirements.

F. A NOVEL PARAMETERS OPTIMIZATION METHOD WITH PSO

A lot of parameters are contained in RPN network. These parameters can be divided into two categories. One part is convolutional layers related parameters. The other part is the parameters in the improved loss function. The parameters of Faster R-CNN are listed in the Table 1. The variables *base_lr*, *lr_policy*, *gamma* and *stepsize* are the parameters of learning rate. The definition of learning rate can be described

TABLE 1. Parameters of faster R-CNN.

Parameter	Description	Value
base_lr	Initial value of the learning rate	0.001
lr_policy	Learning rate policy: drop the learning rate in steps by a factor of gamma every stepsize iterations	"step"
gamma	Factor of dropped learning rate	0.1
stepsize	Drop the learning rate every stepsize iterations	50000
momentum	Weight of the previous update	0.9
weight_decay	Factor of the regularization	0.0005
max_iter_rpn	Number of RPN execution steps	80000
IoU_ FG_RPN	IoU value for RPN foreground proposals	[0.7,1]
IoU_ BG_RPN	IoU value for RPN background proposals	[0,0.3)
NMS_IoU	Threshold value of IoU for NMS method	0.7
max_iter_FRC	Number of Fast R-CNN execution steps	40000
IoU_ FG_FRC	IoU value for Fast R-CNN foreground proposals	[0.5,1]
IoU_ BG_FRC	IoU value for Fast R-CNN foreground proposals	[0.1,0.5)

as follows:

$$w_i = w_i - \eta \frac{\partial E(w)}{\partial w_i} \tag{32}$$

where w_i represents the convolutional weights, the item $E(w)$ means the loss function, variable η is the learning rate.

From (32) we can see that the update speed of convolutional weights are decided by the learning rate η. If the value of η is small, then the update speed of convolutional weights is slow. In other words, the training time is relative long. However if the value of η is large, thus the update speed of convolutional weights is fast. Nevertheless, the optimal weights may not be found. We can see that the setting of η is very important. In Faster R-CNN, the expression for η is showed as follows:

$$\eta = \eta \times \gamma \tag{33}$$

where η is the learning rate and the initial value of η is *base_lr*, variable γ is the factor of dropped learning rate. The (33) is executed after each *stepsize* iteration. The value of η is relative large at the beginning of training, and the value of η is gradually decreased as the iteration increases. However, the value of γ is selected by the prior knowledge. Therefore these values may not be the best.

In the other side, the α and β factors in smooth L_1 loss function are set as 1 and 0.5 based on the experiments. However these values are not the optimal. Specially, the additional factor θ is added to the improved loss function. Therefore, the complexity of experiment for finding the best values of influence factors α, β and θ are increased.

In general, the principle for finding the best values of γ, α, β and θ is the same as grid search method. However, the definition of searching step is hard to find by grid search method. If the value of searching step is large, then the processing results are not stable. In other words, the exploitation ability is not good. In contrary, if the value of searching step is small, thus the processing results may be the local optimal value.

In order to optimize these influence factors, the optimization method PSO is introduced in our proposed PBLS_SRPN. The optimization of influence factors is determined by two important aspects. First, each particle is constructed by four parameters γ, α, β and θ. Thereupon, the flying of particles represents the changes in parameters γ, α, β and θ. Secondly, the performance of each particle is evaluated by the fitness function. In this paper, the fitness function is our improved loss function. Therefore the local and global optima are updated based on the fitness value.

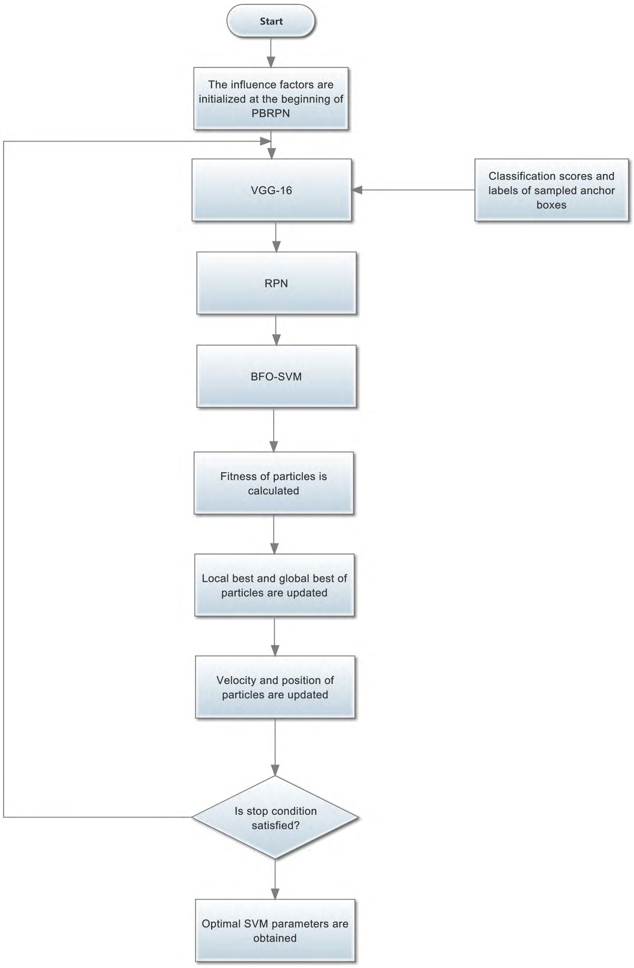

Start
The influence factors are initialized at the beginning of PBRPN
VGG-16
Classification scores and labels of sampled anchor boxes
RPN
BFO-SVM
Fitness of particles is calculated
Local best and global best of particles are updated
Velocity and position of particles are updated
Is stop condition satisfied?
Optimal SVM parameters are obtained

FIGURE 11. The flowchart of parameters optimization process with PSO.

From Figure 11 we can see the flowchart of the PBLS_SRPN parameters optimization with PSO. At first, the PSO related variables and the parameters γ, α, β and θ are initialized. The change of learning rate is mainly decided by the parameters γ. The initial values of γ is 0.001. If the range of random change value is large, then the convergence speed is slow. Therefore, the range of random change value is limited in the PBLS_SRPN. In addition, the initial values of α, β and θ in loss function are 1,0.5 and 0.5. The range of random

TABLE 2. Date set information.

No.	Data set	No. Of categories	No. Of images	No. Of annotated objects
1	PASCAL VOC 2007	20	9963	24640
2	PASCAL VOC 2012	20	11530	27450
3	MS COCO 2015	80	328000	2500000
4	KITTI	3	14999	80256

change for these parameters is also limited. In PSO, the local best and global best of particles are generated and updated. The local best and global best values are calculated through fitness function. Then, the velocity and position of particles are updated. Furthermore, if the iteration stop condition of PSO is reached, then the optimization process is finished.

G. PSEUDO CODE OF PBLS_SRPN

The main target of our PBLS_SRPN is to calculate region proposals. In order to introduce the process clearly, pseudo code of PBLS_SRPN is presented. The pseudo code for the training process of PBLS_SRPN is showed as follows:

```
[Step One] The parameters of PSO are initialized and the process of PSO optimization is started.
For $i = 1$ to $PSO\_iter$ Do
    [Step Two] The VGG-16 network is initialized by the pre-trained model.
    For $i = 1$ to $maxiter$ Do
        [Step Three] A resized image is sent to VGG-16 network for forward propagation calculation.
        [Step Four] The lower feature maps and the higher feature maps are integrated with the output features in MLEN.
        [Step Five] The anchor boxes are generated based on the output features.
        [Step Six] The classification and improved regression loss function is calculated based on the top-level features.
        [Step Seven]The back propagation for PBLS_SRPN is executed.
        [Step Eight]The local best and global best values are updated.
    End
End
```

The pseudo code for the testing process of PBLS_SRPN is described as follows:

```
[Step One] The VGG-16 network is initialized by the pre-trained model.
For $i = 1$ to $maxiter$ Do
    [Step Two] A resized image is sent to VGG-16 network for forward propagation calculation.
    [Step Three] The lower feature maps and the higher feature maps are integrated with the output features in MLEN.
    [Step Four] The anchor boxes are generated based on the output features.
    [Step Five] The region proposals are generated by the classification and improved regression operations.
End
```

H. COMPARING PBLS_SRPN TO FASTER R-CNN

The object detection performance is affected by the feature sampling quality. Therefore the feature sampling ability is strengthened by introducing the MLEN method in PBLS_SRPN. In order to increase the optimizability of RPN, the improved smooth L_1 loss function and SVM classifier is applied to RPN. However, acquiring the optimal parameter values of SRPN is a NP-hard problem. Thereupon, the learning strategy with PSO and BFO is used to optimize the parameters of SRPN. In other words, the parameters of SVM and the improved smooth L_1 loss function are optimized by the BFO and PSO methods respectively. Thereafter, the exploring space of RPN is expanded by our proposed method. Specially, the global optimal solution can be achieved by using our learning strategy. Consequently, the object detection ability of PBLS_SRPN is much better than Faster R-CNN.

I. IMPLEMENTATION DETAILS

Firstly, the short side of the input image is changed to 600. In addition, the aspect ratio of the input image is not modified. Secondly, the adjusted image is processed by the pre-trained VGG-16 model of PBLS_SRPN. Thirdly, the multi-level features are sampled based on our proposed MLEN. Fourthly, the anchor boxes that do not satisfy our condition are ignored. In addition, the NMS method is applied to reduce the overlapped region proposals. In PBLS_SRPN, the value of *NMS_IoU*is set to 0.7. In the experiment chapter, the results are showed the performance of our improved PBLS_SRPN based on the optimized parameters.

V. EXPERIMENTS

A. DATA SETS INTRODUCTION

The training and testing data sets of our proposed PBLS_SRPN are PASCAL VOC 2007, 2012 [45], MS COCO [46] and KITTI [47]. The information of our experimental data sets is listed in Table 2. The comparison between PBLS_SRPN and the state-of-the art object detection methods is presented in this chapter. Furthermore, the advantages of PBLS_SRPN on object detection are analyzed deeply.

Our improved PBLS_SRPN is programmed through the Caffe [48] framework. VGG-16 model is pre-trained to set the convolutional parameters. In addition, 13 convolutional layers and 3 fully-connected layers are included in the VGG-16 model. In this paper, 3 fully-connected layers are not used for RPN. The mAP and recall are the mainly evaluation standard

TABLE 3. Parameters of fast R-CNN.

Parameter	Description	Value
base_lr	Initial value of the learning rate	0.001
lr_policy	Learning rate policy: drop the learning rate in steps by a factor of gamma every stepsize iterations	"step"
gamma	Factor of dropped learning rate	0.1
stepsize	Drop the learning rate every stepsize iterations	30000
momentum	Weight of the previous update	0.9
weight_decay	Factor of the regularization	0.0005
max_iter_FRC	Number of Fast R-CNN execution steps	40000
IoU_ FG_FRC	IoU value for Fast R-CNN foreground proposals	[0.5,1]
IoU_ BG_FRC	IoU value for Fast R-CNN foreground proposals	[0.1,0.5)

TABLE 4. Parameters of MR-CNN.

Parameter	Description	Value
base_lr	Initial value of the learning rate	0.001
lr_policy	Learning rate policy: drop the learning rate in steps by a factor of gamma every stepsize iterations	"step"
gamma	Factor of dropped learning rate	0.1
stepsize	Drop the learning rate every stepsize iterations	50000
momentum	Weight of the previous update	0.9
weight_decay	Factor of the regularization	0.0005
max_iter	Number of execution steps	40000
IoU_ FG	IoU value for foreground proposals	[0.5,1]
IoU_ BG	IoU value for background proposals	[0.1,0.5)

TABLE 5. Parameters of ION.

Parameter	Description	Value
base_lr	Initial value of the learning rate	0.001
lr_policy	Learning rate policy: drop the learning rate in steps by a factor of gamma every stepsize iterations	"step"
gamma	Factor of dropped learning rate	0.1
stepsize	Drop the learning rate every stepsize iterations	50000
momentum	Weight of the previous update	0.9
weight_decay	Factor of the regularization	0.0005
max_iter_rpn	Number of RPN execution steps	60000
max_iter_FRC	Number of Fast R-CNN execution steps	40000
NMS_IoU	Threshold value of IoU for NMS method	0.443

for our proposed PBLS_SRPN based on the experimental data sets.

B. PARAMETER SETTING

In this experimental part, the values for parameters of Fast R-CNN, MR-CNN, ION, HyperNet, PBLS_SRPN, BFO and PSO are presented in Tables 3-9. The multi-level extracting network, improved smooth L_1 loss function and SVM classifier are contained in SRPN. In addition, the all improvements are included in PBLS_SRPN.

TABLE 6. Parameters of HyperNet.

Parameter	Description	Value
base_lr	Initial value of the learning rate	0.001
lr_policy	Learning rate policy: drop the learning rate in steps by a factor of gamma every stepsize iterations	"step"
gamma	Factor of dropped learning rate	0.1
stepsize	Drop the learning rate every stepsize iterations	50000
momentum	Weight of the previous update	0.9
weight_decay	Factor of the regularization	0.0005
max_iter_rpg	Number of region proposal generation execution steps	60000
max_iter_detection	Number of detection execution steps	40000
IoU_ FG	IoU value for foreground proposals	[0.45,1]
IoU_ BG	IoU value for background proposals	[0,0.3)
NMS_IoU	Threshold value of IoU for NMS method	0.7

TABLE 7. Parameters of PBLS_SRPN.

Parameter	Description	Value
base_lr	Initial value of the learning rate	0.001
lr_policy	Learning rate policy: drop the learning rate in steps by a factor of gamma every stepsize iterations	"step"
γ	Factor of dropped learning rate	0.1
stepsize	Drop the learning rate every stepsize iterations	50000
momentum	Weight of the previous update	0.9
weight_decay	Factor of the regularization	0.0005
max_iter_rpn	Number of ERPN execution steps	60000
IoU_ FG_RPN	IoU value for ERPN foreground proposals	[0.6,1]
IoU_ BG_RPN	IoU value for ERPN background proposals	[0,0.3)
α	Factors for improved loss function	1
β	Factors for improved loss function	0.5
θ	Factors for improved loss function	0.5
NMS_IoU	Threshold value of IoU for NMS method	0.7

TABLE 8. Parameters of BFO.

Parameter	Description	Value
N	Bacterial number	30
N_{re}	Reproductive steps	5
N_{ed}	Number of elimination–dispersal steps	2
N_c	Number of chemotactic steps	10
S	Number of swims	4
Ped	Elimination probability	0.25
$d_{attract}$	Height of attractant	0.1
$w_{attract}$	Width of attractant	5
$h_{repellant}$	Height of repellant	0.1
$w_{repellent}$	Width of repellant	10

C. EXPERIMENTS ON PASCAL VOC 2007

The advantages of PBLS_SRPN are presented based on the experiments. PASCAL VOC 2007 data set is applied

TABLE 9. Parameters of PSO.

Parameter	Description	Value
N	Number of particles	10
W	Value of weight	0.7
C_1 *and* C_2	Value of Coefficients	2
i	Maximum number of iterations	30

to evaluate the ability of PBLS_SRPN. Around 5k trainval images and 5k test images of 20 categories are included in the PASCAL VOC 2007 data set. In order to enhance the quality of training, the trainval datasets of VOC 2007 and VOC 2012 are integrated into one experimental training data set. The average precision (AP) value is calculated based on the object detection method for a kind of object in data set. In addition, if the highest AP value is achieved by an object detection method for a kind of object, then this AP value is bold-faced in the experiment results. Additionally, all algorithms are trained on PASCAL VOC 2007 and 2012 data sets in section B and C.

From Table 10 we can find that the mAP of SRPN is 74.6% which is higher than Fast R-CNN and Faster R-CNN. Because the MLEN is included in SRPN, thus the lower feature maps and the higher feature maps are integrated with the output feature maps. In other words, the capability of feature sampling method of our SRPN is enhanced. Moreover, the square of L_2 loss is added to the loss function to promote the ability of fitting. In this way, the performance of L_2 loss is strengthened in range of $(-\alpha, \alpha)$. As a result, the performance of smooth L_1 loss function is promoted. Thereafter, good results are obtained by SRPN based on our improvements. Particularly, the mAP 78.9% of PBLS_SRPN is the best. The outstanding results are got by PBLS_SRPN on the bike, bird, boat, bottle, bus, car, chair, table, horse, mbike, person, plant, train and tv. The reason is that the PBLS is applied to SRPN. The BFO method is applied to optimize the parameters of C and γ. Consequently, the performance of classification is promoted by BFO-SVM classifier. In addition, the optimization method PSO is introduced in our proposed PBLS_SRPN. Therefore, the exploring space of PBLS_SRPN is strengthened.

D. EXPERIMENTS ON PASCAL VOC 2012

The PASCAL VOC 2012 data set is applied to train and test by our proposed SRPN[ ]and PBLS_SRPN. Specially, the training data is comprised of the VOC 2007 and VOC 2012 dataset. From Table 11 we can see that the highest AP 89.3% is achieved by Fast R-CNN on cat. Moreover, the highest APs 85.5% and 65.8% are got by MR-CNN. Particularly, the highest APs are acquired by PBLS_SRPN on the rest objects. Furthermore, the 72.8% mAP is achieved by SRPN. This result is higher than Fast R-CNN, Faster R-CNN, ION, MR-CNN and HyperNet. Additionally, PBLS_SRPN obtains 74.8% mAP which is 2.0 point higher than SRPN method. Consequently, the excellent object detection results are acquired by our proposed PBLS_SRPN method on more challenging PASCAL VOC 2012 data set. In other words, the object detection ability of PBLS_SRPN is stable in different data sets. Therefore, the generalization ability of PBLS_SRPN is boosted.

E. SMALL OBJECTS DETECTION

The small object detection task is very challenging. Because few pixels are kept in the small objects, so the small object detection is really difficult. Detection results for bird, bottle and potted plant on VOC 2007 and 2012 are presented in this section. From Table 12 we can see that that the AP of SRPN on detecting bottle, bird and potted plant is higher than Fast R-CNN, Faster R-CNN. Specially, the AP of PBLS_SRPN on detecting bird, bottle and potted plant is the best.

Because the size of feature maps in lower level is large, then the output features combining with lower level features are strengthened in our MLEN. Moreover, the lower feature maps and the higher feature maps are compressed into a uniform space. In this way, the effect of small object feature sampling is strengthened. Simultaneously, the convolutional layers related parameters are optimized through PSO method. Therefore, the training effect of our PBLS_SRPN is enhanced. As a result, our PBLS_SRPN can handle the small object detection problem effectively.

F. ANALYSIS OF RECALL-TO-IoU

Our proposed PBLS_SRPN is compared with the current state-of-the-art object detection method. The number of region proposals for PBLS_SRPN is assigned to 200, 400 and 800 respectively. The selected region proposals are the top ranked ones according to the scores from high to low. On PASCAL VOC 2007 data set, the recall of region proposals is calculated based on the different IoU values. From Figure 12 we can see that as the number of region proposals drops, the recall of MR-CNN, ION, HyperNet, Fast R-CNN and Faster R-CNN is decreased significantly. In other words, these object detection methods need to increase the number of candidate boxes to promote recall at the same IoU value. The reason is that the quality of region proposals in these object detection methods is poor. In addition, the computation speed is affected by using too many region proposals. In our proposed PBLS_SRPN, the lower feature maps and the higher feature maps are integrated with the output features in MLEN. Thereupon, the quality of region proposals in PBLS_SRPN is enhanced. Moreover, the BFO-SVM classifier is introduced to distinguish the foreground and the background region proposals. Thus the valid region proposals are kept. Thereupon the number of region proposals for PBLS_SRPN is assigned to 200. From the object detection results we can see that the recall of our proposed PBLS_SRPN is stable when the number of region proposals is reduced from 800 to 200. Furthermore, the recall of PBLS_SRPN is higher than other methods when the IoU threshold is over 0.6. When the value of IoU equals to 0.7, the recall of our PBLS_SRPN is high and the number of region proposals is not

TABLE 10. Detection results on PASCAL VOC 2007 test set. The best AP of each object category and mAP are bold-faced.

Approach	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster R-CNN	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
MR-CNN	78.2	**80.3**	84.1	78.5	70.8	68.5	88.0	85.9	**87.8**	60.3	**85.2**	73.7	**87.2**	86.5	85.0	76.4	48.5	76.3	75.5	85.0	81.0
ION	75.6	79.2	83.1	77.6	65.6	54.9	85.4	85.1	87.0	54.4	80.6	73.8	85.3	82.2	82.2	74.4	47.1	75.8	72.7	84.2	80.4
HyperNet	76.3	77.4	83.3	75.0	69.1	62.4	83.1	87.4	87.4	57.1	79.8	71.4	85.1	85.1	80.0	79.1	51.2	**79.1**	**75.7**	80.9	76.5
SRPN	74.6	74.6	79.7	74.4	64.9	64.6	82.8	82.1	83.3	56.5	79.7	70.2	82.8	81.6	80.7	74.3	47.4	73.1	71.3	85.6	81.9
PBLS_SRPN	**78.9**	79.7	**84.6**	**79.2**	**69.7**	**68.9**	**88.3**	**87.8**	87.6	**61.8**	83.7	**74.9**	86.2	**86.6**	**85.7**	**79.3**	**52.2**	77.5	75.5	**86.1**	**82.3**

TABLE 11. Detection results on PASCAL VOC 2012 test set. The best AP of each object category and mAP are bold-faced.

Approach	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	**89.3**	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster R-CNN	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
MR-CNN	73.9	**85.5**	82.9	76.6	57.8	62.7	79.4	77.2	86.6	55.0	79.1	62.2	87.0	83.4	84.7	78.9	45.3	73.4	**65.8**	80.3	74.0
ION	71.1	83.3	80.5	71.5	56.5	53.0	77.4	73.8	85.8	52.6	76.8	59.1	83.9	81.3	79.3	77.2	45.7	72.6	64.2	80.1	68.1
HyperNet	71.4	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
SRPN	72.8	84	81.8	75.2	55.6	61.8	77.1	76.3	85.2	54.1	77.9	61.2	85.5	83.1	83.4	77.8	47.1	72	64.6	79.8	72.5
PBLS_SRPN	**74.8**	85.2	**83.5**	**76.8**	**58.1**	**63.3**	**79.5**	**80.2**	86.5	**55.7**	**79.5**	**63.1**	**87.8**	**84.6**	**85.1**	**82.1**	**49.2**	**74.1**	65.4	**81.9**	**74.5**

TABLE 12. Small objects detection results on PASCAL VOC 2007 and VOC 2012 test set. The best AP of each object category is bold-faced.

Approach	VOC 2007			VOC 2012		
	bird	bottle	plant	bird	bottle	plant
Fast R-CNN	69.3	38.3	31.8	70.8	38.7	35.1
Faster R-CNN	70.9	52.1	38.8	74.3	49.8	40.1
MR-CNN	78.5	68.5	48.5	76.6	62.7	45.3
ION	77.6	54.9	47.1	71.5	53.0	45.7
HyperNet	75.0	62.4	51.2	73.6	53.7	48.6
SRPN	74.4	64.6	47.4	75.2	61.8	47.1
PBLS_SRPN	**79.2**	**68.9**	**52.2**	**76.8**	**63.3**	**49.2**

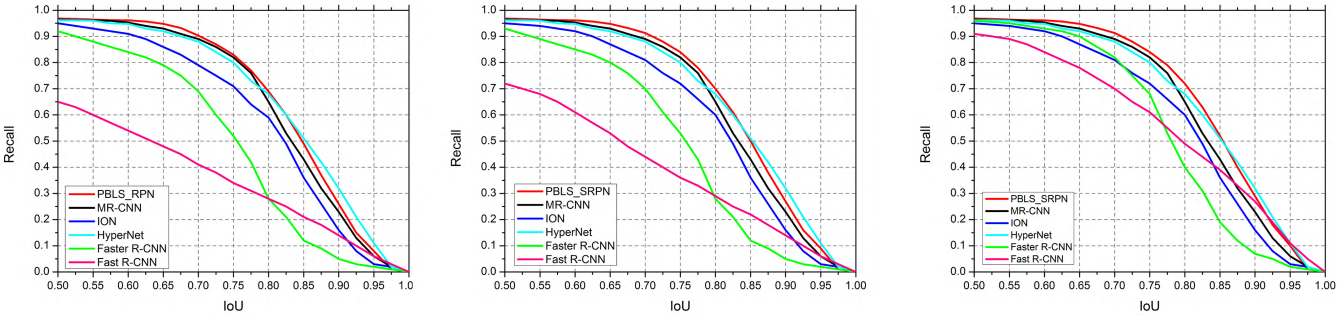


FIGURE 12. Recall versus IoU threshold on the PASCAL VOC 2007 data set. Left: 200 region proposals. Middle: 400 region proposals. Right: 800 region proposals.

seriously filtered. Thereafter, the value of IoU threshold is assigned to 0.7 for our PBLS_SRPN. Consequently, around 0.9k region proposals are generated. This number of region proposal is seriously less than other comparison methods. Thereupon, the advantages of our PBLS_SRPN are presented through the experiment results in this section.

TABLE 13. Experiment results over different loss functions.

Data set	VOC 2007				VOC 2012				MS COCO			
L_{cls}	√	√	√	√	√	√	√	√	√	√	√	√
L_{reg}(RPN)		√				√				√		
L_{reg}(SRPN)			√				√				√	
L_{reg}(PBSL_SRPN)				√				√				√
mAP (%)	69.4	73.2	74.6	78.9	66.2	70.4	72.8	75.8	16.8	21.9	24.8	31.5

TABLE 14. Comparison between softmax, SVM and BFO-SVM classifiers.

Classifier	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	talbe	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Softmax	74.9	75.4	80.5	75.2	65.7	65.4	83.6	82.9	84.1	57.3	80.5	71	83.6	82.4	81.5	75.1	48.2	73.9	72.1	81.4	78.1
SVM	74.6	74.5	80.1	75	65.1	65	83.1	82.2	83.6	57.1	80.3	70.6	83.2	82	81.3	74.7	47.9	73.6	72.2	81.6	78.3
BFO-SVM	**76.8**	**77.4**	**82.2**	**77.2**	**67.3**	**67**	**85.7**	**84.6**	**85.7**	**59.4**	**81.9**	**72.7**	**85.2**	**84.4**	**83.9**	**76.5**	**50.2**	**76.2**	**74.4**	**83.5**	**80.3**

G. ANALYSIS OF IMPROVED LOSS FUNCTION

In this section, the advantages of our proposed loss function are presented on VOC 2007, VOC 2012 and MS COCO data sets. Four different situations are showed in Table 13. The first column of each group means that only the classification loss ($\lambda = 0$) is contained in loss function. In other words, the bounding box regression function is not available. Therefore the worst results are got by the RPN with only classification loss. The second column of each group represents the RPN loss function. Clearly, the results of second column for each group are better than that of first column. Thereafter, the bounding-box regression function is very effective for loss function. Moreover, the third column of each group shows the RPN with improved loss function. However, the parameters are the initialized values. From the results we can see that the performance of the third column is higher than that of RPN loss function. Thereupon, the novel loss function is available. Finally, the results of fourth column are the best. Consequently, the performance of PBLS_SRPN through PSO based learning strategy is excellent. Thus our novel loss function and learning strategy are effective. Because the factors of the L_{cls} and L_{reg} are the same as that of RPN, thus the value λ is assigned to 10.

H. THE EFFECT OF BFO OPTIMIZATION METHOD

In order to present the optimization ability of BFO, the BFO-SVM classifier is applied to compare with the SVM and softmax classifiers in this section. The comparing results are illustrated in the following table on PASCAL VOC 2007 test set.

From Table 14 we can see that the mAP for SRPN with softmax is 74.9%. In addition the mAP for SRPN with SVM is 74.6% which is 0.3 point lower than that of SRPN with softmax. The grid search method is applied to select the suitable parameters for the RBF kernel function of SVM. However the global optimization ability for grid search is poor. Therefore the solutions are easily to fall into local optima. In other words the exploring ability for grid search cannot satisfy our requirements.

The capability of SVM is significantly affected by the parameters C and γ. In order to strengthen the performance of SVM, the BFO method is used to optimize the parameters C and γ of SVM. From Table 14 we can find that the mAP of SRPN with BFO-SVM is 76.8% which is higher than other methods. Thereupon the classification ability of SRPN is enhanced by introducing BFO-SVM. In other words, our proposed improvement BFO-SVM is effective.

I. THE EFFECT OF PSO OPTIMIZATION METHOD

The optimization parameters are divided into two categories. The first type of parameter is the convolutional network related parameter. Learning rate γ represents this type of parameter. This parameter is reduced by increasing the iteration. The initial value of γ is 0.1. Because the training of PBLS_SRPN takes some time, thus the range of variation for γ is limited in [0.05, 0.15] to save time. The second type of parameter is the improved loss function related parameter. Parameters α, β and θ need to be optimized by our proposed PSO method. Simultaneously, the initial values for α, β and θ are 1, 0.5 and 0.5 respectively. Moreover, the range of variation for these parameters is defined in [0.02, 1.8], [0.1, 0.9] and [0.1, 0.9]. In this part, the experiments are executed by our proposed PBSL_SRPN on PASCAL VOC 2007 data set.

Ten particles are constructed to search the optimal value for γ, α, β and θ. In order to present the variation process for the values of γ, α, β and θ clearly, one particle is sampled to illustrate the variation process for the values of γ, α, β and θ in Figure 13. Four parameters γ, α, β and θ are initialized by 0.1, 1, 0.5 and 0.5 respectively. Moreover these parameters are affected by the random value from our

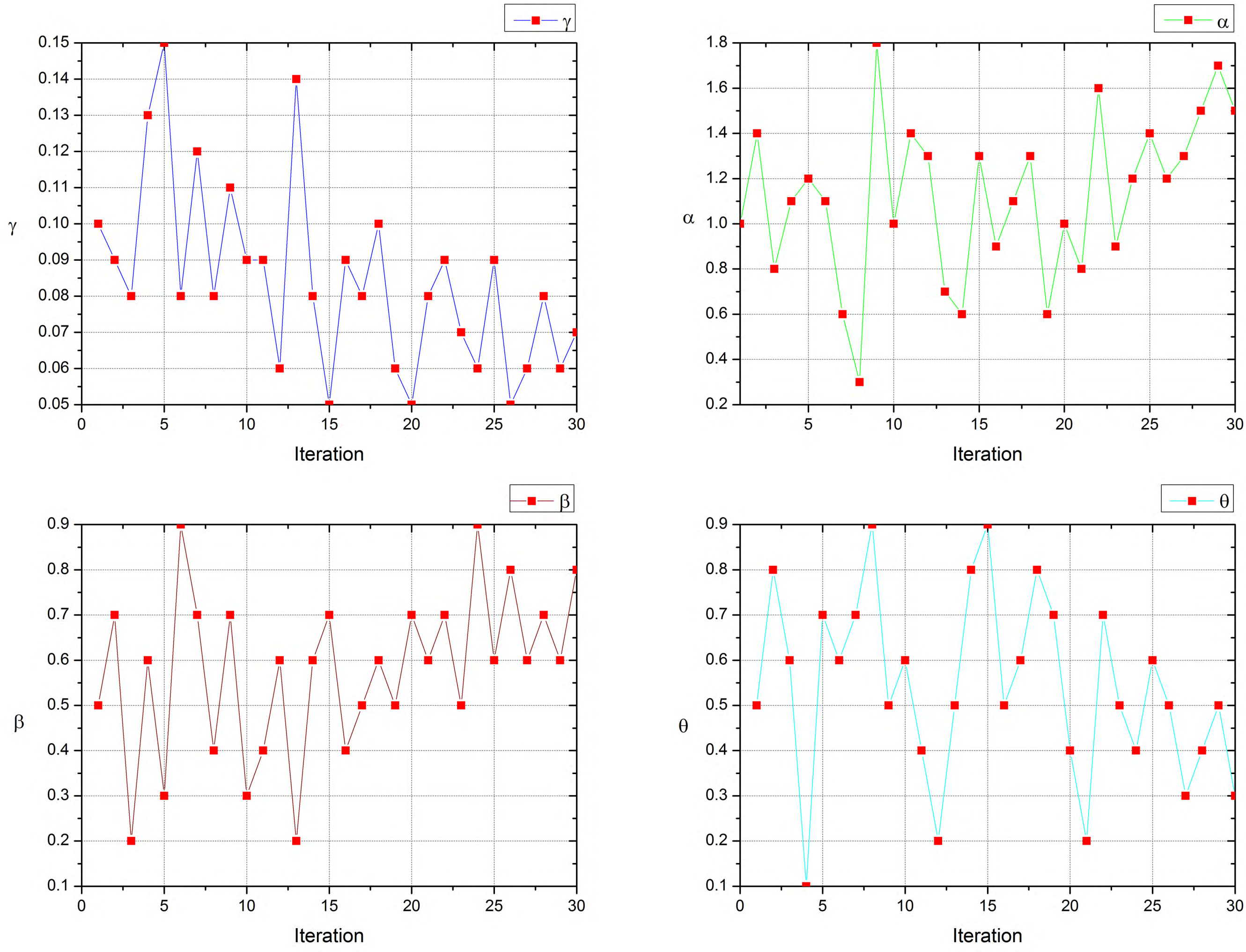


FIGURE 13. The variation process for optimization parameters.

defined range. Thereupon we can find that the values for γ, α, β and θ change drastically before 10 iteration steps. Specially, the exploring ability of PSO is presented by this phenomenon. Variation value can be selected by the particles. Furthermore through the update of speed and location, local optimal values and global optimal value are obtained. The optimization of γ, α, β and θ is affected by local optimal values and global optimal value which promote the exploitation ability of the PSO method. From the experiment results we can see that the values for γ, α, β and θ tend to be stable at the last 10 iteration steps. The values of γ and θ are gradually decreased. In addition, the values of α and β are progressively increased. Finally, the best value for parameters of γ, α, β and θ can be achieved by our proposed method.

J. EXPERIMENTS ON MS COCO

In this part, our experiment is executed over the MS COCO data set. Around 80k training images and 40k validation images are included in the MS COCO data set. These images are divided into 80 categories. The experiment results are presented on the standard test set (test-std). The experiment results are calculated from different aspects based on the MS COCO data set. The IoU threshold is fixed on the PASCAL VOC dataset. Therefore, the coverage for the experiment results is not enough. Nevertheless, different IoU thresholds are applied to execute the experiments. Thereafter, MS COCO dataset can reflect the ability of the object detection methods better. In SRPN, the lower feature maps and the higher feature maps are integrated with the output features in MLEN. From Table 15, we can find that the average experiment result of SRPN is higher than that of Fast R-CNN and Faster R-CNN. Therefore, our MLEN improvement is available. Additionally the fitting ability of loss function for SRPN is promoted. Furthermore, the capability for distinguishing the foreground and the background region boxes is strengthened through BFO-SVM classifier.

Consequently, the parameters of covolutional layers and the improved loss function are optimized by PSO method. As a result, the best experiments are obtained by the PBLS_SRPN with all improvements. Thereupon, excellent

TABLE 15. Detection results on MS COCO test-std. The best result is bold-faced.

Method	Avg.Precision,IoU:			Avg.Precision,Area:			Avg.Recall,#Dets:			Avg.Recall,Area:		
	0.5:0.95	0.5	0.75	Small	Med.	Large	1	10	100	Small	Med.	Large
Fast R-CNN	19.3	39.3	19.9	3.5	18.8	34.6	21.4	29.5	29.8	7.7	32.2	50.2
Faster R-CNN	21.9	42.7	22.3	4.2	19.3	36.1	22.5	31.2	31.4	8.9	34.2	52.3
ION	30.7	52.9	31.7	11.8	32.8	44.8	27.7	42.8	45.4	23	50.1	63
SRPN	24.8	46.2	25.1	8.8	22.6	39.4	25.8	35.6	34.5	13.5	37.5	55.7
PBRPN	**31.5**	**53.8**	**31.6**	**13.1**	**33.8**	**45.7**	**28.5**	**43.4**	**46.2**	**24.7**	**50.8**	**64.6**

TABLE 16. Detection results on KITTI data set. The best result is bold-faced.

Method	Car			Pedestrian			Cyclist		
	Moderate	Easy	Hard	Moderate	Easy	Hard	Moderate	Easy	Hard
FRCNN+Or	78.95	89.87	68.97	56.78	71.18	52.86	57.37	70.05	51.00
Mono3D	87.86	**90.27**	78.09	66.66	77.30	63.44	63.85	75.22	58.96
Faster R-CNN	79.11	87.9	70.19	65.91	78.35	61.19	62.81	71.41	55.44
Regionlets	76.56	86.50	59.82	61.16	72.96	55.22	58.69	70.09	51.81
DPM-VOC+VP	66.25	80.45	49.86	44.86	59.60	40.37	31.16	43.65	28.29
SRPN	87.2	88.2	77.4	69.3	81.2	63.9	69.3	77.8	60.8
PBLS_SRPN	**88.6**	**89.7**	**78.6**	**70.5**	**82.4**	**65.1**	**70.3**	**78.72**	**61.6**

experiments are achieved by our proposed PBLS_SRPN object detection method.

K. EXPERIMENTS ON KITTI

In order to test the effect of our proposed methods on autonomous driving dataset, SRPN, PBLS_SRPN and other five comparing algorithms are tested on the KITTI data set in this section. Around 7481 training images and 7518 test images are contained on the KITTI data set. A total of 80256 objects are labeled in this data set. The images are captured based on the autonomous driving vehicles. Three object categories: car, pedestrian and cyclist are included in KITTI. Additionally, three levels of evaluation: easy, moderate and hard are provided. The most frequently used level is moderate. Specially, 70% overlap for cars is required. Simultaneously, 50% overlap is needed for pedestrians and cyclists. The ability of object detection method is evaluated by the AP value. The comparing experiment results are presented in the Table 16. We can see that the moderate values of SRPN are better than that of FRCNN+Or, Mono3D, Faster R-CNN, Regionlets and DPM-VOC+VP on pedestrian and cyclist dataset. The moderate value of SRPN is better than that of FRCNN+Or, Faster R-CNN, Regionlets and DPM-VOC+VP on car dataset. The reason is that MLEN and novel loss function are designed in the SRPN. In other words, the lower and higher feature maps are merged with the top feature maps. Simultaneously, the novel loss function based on Huber loss is developed. Therefore, the fitting ability of SRPN is enhanced. Moreover, the moderate values of PBLS_SRPN are higher than other methods. Thereafter, the outstanding results are achieved by our learning strategy. The parameters of SVM and convolutional network are optimized by BFO and PSO respectively. As a result, our proposed solution obtains excellent results on KITTI data set.

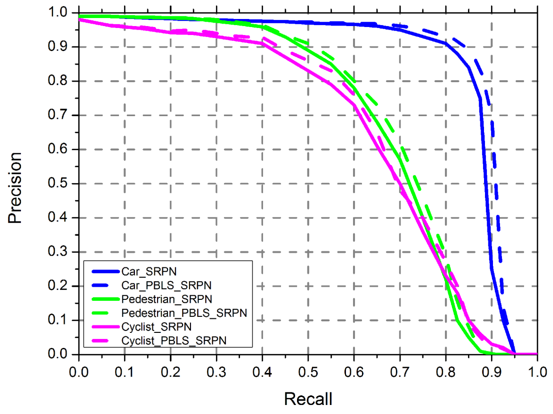


FIGURE 14. Recall versus precision on the KITTI data set.

As we all know that recall and precision are often used to evaluate the performance of object detection algorithms. Additionally, recall and precision are often conflicting goals in the sense that if one wants to see more relevant items (i.e., to increase recall value), usually more nonrelevant ones are also retrieved (i.e., precision decreases). On KITTI data set, the precision values of SRPN and PBLS_SRPN are calculated based on the different recall values. From Figure 14 we can see that as the recall value decreases, the precision values of SRPN and PBLS_SRPN also decrease. Additionally, the precision values of PBLS_SRPN are better than that of SRPN in all car, pedestrian and cyclist categories. In other words, the performance of classification is strengthened by BFO-SVM classifier. In addition, the parameters of γ, α, β and θ are optimized by PSO. Therefore, PBLS_SRPN method can achieve the excellent result. Moreover, the precision values of PBLS_SRPN and SRPN are close for pedestrian and cyclist. Thereafter, the ability of PBLS_SRPN for detecting pedestrian and cyclist needs to be further improved. However, the precision value of PBLS_SRPN is significantly better than that of SRPN for car. Thereupon, our PBLS_SRPN is suitable to detect car category. As a result, our learning strategy with PSO and BFO is effective.

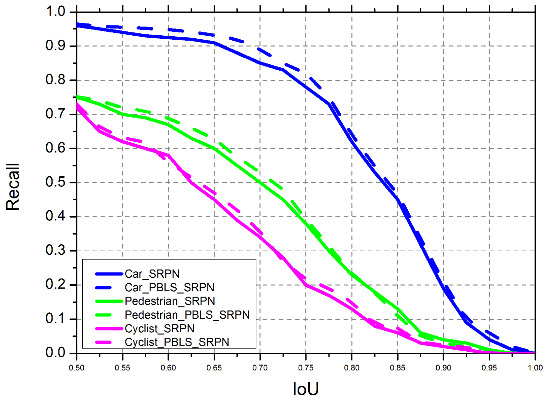


FIGURE 15. **Recall versus IoU on the KITTI data set.**

In this part, the recall values are calculated based on the different IoU values on KITTI data set. Additionally, the number of region proposals for SRPN and PBLS_SRPN is assigned to 200 as we have discussed in section E. The selected region proposals are the top ranked ones according to the scores from high to low. From Figure 15 we can see that as the IoU value falls, the recall values of SRPN and PBLS_SRPN decrease. However the recall values of PBLS_SRPN are significantly better than SRPN when the IoU threshold is greater than around 0.7. On the one hand, from the experiment results we can see the quality of region proposals is promoted by our learning strategy with PSO and BFO. The reason is that the lower feature maps and the higher feature maps are integrated with the output features in MLEN. On the other hand, the IoU threshold can be assigned to 0.7 as we described in section E. Therefore the computation speed is not affected. Consequently, our learning strategy with PSO and BFO contributes to the quality of region proposals.

L. RUNNING TIME

The object detection speed of PBLS_SRPN is 5 fps which is the same as Faster R-CNN. The frame rate of our PBLS_SRPN method is tested through a single NVIDIA TitanX GPU for VGG-16. The speed of our proposed method is not accelerated, but the mAP of PBLS_SRPN is enhanced by introducing the PBLS learning strategy. Additionally the generalization ability of our proposed method is improved. In autonomous driving filed, safety is a very important factor. Thus, the accuracy and the speed are both should be concerned.

VI. CONCLUSION

In this paper, SRPN is developed to enhance the feature sampling ability and increase the parameters of RPN. On PASCAL VOC 2007, 2012 and MS COCO data sets, SRPN obtained mAPs of 74.6%, 72.8% and 31.5% which are better than that of Faster R-CNN. The reason is that the exploring space of SRPN is expanded. In other words, the generalization ability of SRPN is boosted. Because the optimization of SRPN is a NP-hard problem, therefore our novel PSO and BFO based learning strategy PBLS is designed to solve this problem. From the experiments we can see that the best results are achieved by PBLS_SRPN. This is because the learning ability of classifier is promoted by using BFO method. Additionally, the learning ability of loss function is raised by introducing PSO method. Thereupon the performance of SRPN is promoted by applying our novel learning strategy. Specially, excellent results are achieved by PBLS_SRPN on KITTI data sets. As a result, our methods can be applied to autonomous driving for object detection effectively. In the future, we will apply our improved smooth L_1 loss function and the learning strategy to other methods for improving the ability of object detection.

REFERENCES

[1] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 23, no. 3, pp. 73–80, Jun. 2010.

[2] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, vol. 9, no. 4, pp. 3246–3253.

[3] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2155–2162.

[4] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, "Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 249–258.

[5] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 530–534, May 1997.

[6] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *Proc. Eur. Conf. Comput. Vis.*, Jun. 2000, pp. 18–32.

[7] P. F. Felzenszwalb and D. Huttenlocher, "Efficient matching of pictorial structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2000, pp. 66–75.

[8] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3286–3293.

[9] C. L. Zitnick and P. Dollär, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 391–405.

[10] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. van Gool, "Deepproposal: Hunting objects by cascading deep convolutional layers," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 2578–2586.

[11] W. Kuo, B. Hariharan, and J. Malik, "DeepBox: Learning objectness with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2479–2487.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[14] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. CVPR*, Jun. 2008, pp. 1–8.

[15] D. C. Ciregan, J. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 3642–3649.

[16] D. Biswas, H. B. Su, C. Y. Wang, J. Blankenship, and A. Stevanovic, "An automatic car counting system using OverFeat framework," *Sensors*, vol. 17, no. 7, p. 1535, Jun. 2017.

[17] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" in *Proc. IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 814–830, Sep. 2016.

[18] Y. Hua, K. Alahari, and C. Schmid, "Online object tracking with proposal selection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3092–3100.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[20] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[22] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[23] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2015.

[24] Y.-P. Chen *et al.*, "A novel bacterial foraging optimization algorithm for feature selection," *Expert Syst. Appl.*, vol. 83, pp. 1–17, Apr. 2017.

[25] P. Zhou, G. Cheng, Z. Liu, S. Bu, and X. Hu, "Weakly supervised target detection in remote sensing images based on transferred deep features and negative bootstrapping," *Multidimensional Syst. Signal Process.*, vol. 27, no. 4, pp. 925–944, Oct. 2016.

[26] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[27] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 265–278, Jan. 2019.

[28] A. Torralba, "Contextual priming for object detection," *Int. J. Comput. Vis.*, vol. 53, no. 2, pp. 169–191, 2003.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2012, vol. 25, no. 2, pp. 1097–1105.

[30] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware CNN model," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1134–1142.

[31] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2874–2883.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[33] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 845–853.

[34] C. Guindel, D. Martin, and J. M. Armingol, "Fast joint object detection and viewpoint estimation for traffic scene understanding," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 4, pp. 74–86, Apr. 2018.

[35] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2147–2156.

[36] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 37, no. 10, pp. 2071–2084, Oct. 2015.

[37] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Multi-view and 3D deformable part models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2232–2245, Nov. 2015.

[38] B. Pepikj, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3286–3293.

[39] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[40] M. Jiang, Y. Liang, X. Feng, and X. Fan, "Text classification based on deep belief network and softmax regression," *Neural Comput. Appl.*, vol. 29, no. 7, pp. 61–70, Jun. 2016.

[41] S. Nowozin, "Optimal decisions from probabilistic models: The intersection-over-union case," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 548–555.

[42] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[43] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Jun. 2002.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, Sep. 2014, pp. 1150–1210.

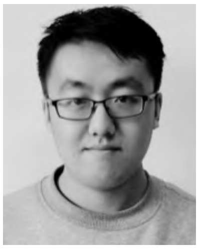
[45] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[46] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Apr. 2014, pp. 740–755.

[47] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3354–3361.

[48] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, Jun. 2014, pp. 675–678.

GANG WANG received the Ph.D. degree in theoretical computer science from Jilin University. He held a Postdoctoral position with the College of Geo-Exploration Science and Technology, Moving Platform Exploration Technology R&D Center, Jilin University, where he is currently an Associate Professor with the College of Computer Science and Technology. His research interests include computer vision and machine learning.



JINGMING GUO received the bachelor's degree in software engineering from Jilin University, where he is currently pursuing the master's degree with the College of Computer Science and Technology. His research interests include automatic control theory and computer vision.

YUPENG CHEN received the B.S. and M.S. degrees in computer science and technology from Jilin University, where he is currently pursuing the Ph.D. degree in computer application technology. He has published six papers in journals and international conference. His research interests include feature selection, machine learning, and object detection. He is a valued member of the Society of Automotive Engineers International.

YING LI received the B.S., M.S., and Ph.D. degrees from Jilin University, where she has been a Professor in computer application technology, since 2006. She has published over 60 papers in journals and international conference. Her research interests include big data, 3-D visual modeling, 3-D image processing, machine vision, and machine learning. She is currently a Fellow of the China Computer Federation.

QIAN XU received the B.S. and M.S. degrees in software engineering from Jilin University, where he is currently pursuing the Ph.D. degree in computer application technology. His research interests include object detection and image segmentation.