A comprehensive swarming intelligent method for optimizing deep learning-based object detection by unmanned ground vehicles

Qian Xu, Gang Wang, Ying Li, Ling Shi, Yaxin Li

1 College of Computer Science and Technology, Jilin University, Changchun, People’s Republic of China, 2 Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, People’s Republic of China, 3 State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun, People’s Republic of China, 4 Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, 5 College of Automotive Engineering, Jilin University, Changchun, People’s Republic of China

* wanggang.jlu@gmail.com

Abstract

Unmanned ground vehicles (UGVs) are an important research application of artificial intelligence. In particular, the deep learning-based object detection method is widely used in UGV-based environmental perception. Good experimental results are achieved by the deep learning-based object detection method Faster region-based convolutional neural network (Faster R-CNN). However, the exploration space of the region proposal network (RPN) is restricted by its expression. In our paper, a boosted RPN (BRPN) with three improvements is developed to solve this problem. First, a novel enhanced pooling network is designed in this paper. Therefore, the BRPN can adapt to objects with different shapes. Second, the expression of BRPN loss function is improved to learn the negative samples. Furthermore, the grey wolf optimizer (GWO) is used to optimize the parameters of the improved BRPN loss function. Thereafter, the performance of the BRPN loss function is promoted. Third, a novel GA-SVM classifier is applied to strengthen the classification capacity. The PASCAL VOC 2007, VOC 2012 and KITTI datasets are used to test the BRPN. Consequently, excellent experimental results are obtained by our deep learning-based object detection method.

Introduction

Over the past decade, UGVs has gone from being laboratory curiosities to functional machines that are closely related to human life. UGVs can be applied to mining areas, parks and warehouses for different purposes. Specifically, object detection is a key technique of UGVs. Our research team has created a Jilin University (JLU) UGV with environmental perception, precise positioning and path planning functions. Our JLU UGV is illustrated in Fig 1.

Recently, deep learning technology has made great progress in object detection [1–4]. In particular, excellent results are obtained in the detection of pedestrians, bikes, cars and buses.
Hand-engineered feature-based methods [5–7] and deep learning network-based methods [8–11] are two kinds of mainstream object detection methods. The scale-invariant feature transform (SIFT) [12], histogram of oriented gradients (HOG) [13] and deformable part models (DPMs) [14] are famous hand-engineered feature-based methods. Invariant features can be sampled from the images using the SIFT method. Furthermore, different views of an object or scene are matched by the SIFT features. The appearance and shape of local objects in images can be extracted through the HOG method. In general, the HOG or SIFT features are applied to sample the features of objects in the DPM method.

Recently, deep learning network-based methods have obtained excellent detection results. Specifically, OverFeat [15] and multicolumn deep neural networks [16] have obtained better classification results than traditional object detection methods. Experience is applied to develop the training model of traditional methods such as the DPM and HOG method; thus, the adaptability of traditional methods is poor. Nevertheless, deep learning network-based methods can be suitable for various objects based on the same model.

Excellent classification accuracy and localization have been achieved by deep learning methods with high-quality region proposals [17, 18]. Region proposal-based object detection methods are illustrated as follows. Good object detection results are obtained by the region-based convolutional neural network (R-CNN) [19]. Additionally, the detection performance of the R-CNN is improved in the Fast R-CNN [20]. Furthermore, the quality of region proposals is enhanced in Faster R-CNN [21]. The selective search (SS) [22] is replaced by a region proposal network (RPN) to accelerate the generation speed of region proposals. In Faster R-CNN, an RPN is applied to generate region proposals. The RPN is a type of fully convolutional network (FCN) [23].

From the description of the aforementioned object detection methods, we can see that three drawbacks of Faster R-CNN need to be addressed. First, the lower and higher information is not integrated with the output features. Additionally, the scale for the RPN pooling method is fixed. In other words, the pooling method is not suitable for objects of different
sizes. Therefore, the feature sampling ability of the RPN is affected. Second, the classification loss function is not sufficiently trained on the negative samples. Moreover, the coefficients for classification and the regression loss function are irrational. Thereafter, the training performance of the RPN is not satisfactory. Third, the softmax method is not adapted to classify the positive and negative proposals. As a result, the classification ability is not good. Therefore, the performance of the RPN needs to be improved.

Three improvements are developed in this paper. First, the synthesized feature sampling strategy is introduced to integrate the different layer features with the output feature. In addition, multiple pooling sizes are added to the RPN. Therefore, the RPN can adapt to objects with different shapes, and the feature sampling ability of the RPN is enhanced. Second, the expression of classification is improved to train the negative samples. Specifically, additional parameters are added to the classification and regression loss function. Furthermore, the grey wolf optimizer (GWO) is used to optimize the parameters of the improved loss function in the RPN. As a result, the performance of the loss function is improved. Third, the softmax function is replaced by a support vector machine (SVM) to improve the classifier. Additionally, the parameters of SVM are optimized by the genetic algorithm (GA). Therefore, the classification performance of the RPN is improved. As a result, the object detection ability of Faster R-CNN is strengthened.

Because some parameters of the RPN are designed based on prior knowledge, the generalizability of the RPN is inadequate. In general, we can solve this problem by replacing the fixed value with variables. Therefore, the exploration space of the RPN is expanded. The boosted RPN (BRPN) is introduced based on these reasons. However, obtaining the optimal parameter values of the BRPN is a nondeterministic polynomial-time hard (NP-hard) problem. Therefore, the GA and GWO are applied to the BRPN. From the experimental results, we can see that our methods are effective.

The following part will introduce related work. Then the original GWO will be introduced in the literature review part. After that, our approach and contributions are described in detail. Then the experimental setup and result analysis are shown seriously. At last, we conclude the advantage and the limitation of our proposed method, and prospect the future research direction.

Related work

In this section, several well-known deep learning-based methods are presented. An R-CNN is designed to model the relationship between object detection and image classification. First, approximately 2000 region proposals are generated using the SS method. Second, a pretrained CNN [24–27] is applied to extract the convolutional features of each region proposal. Third, the output features are classified by the SVM method. The object detection mean average precision (mAP) is improved by the RPN. Nevertheless, the processing speed is the bottleneck for the RPN.

The object detection ability of the RPN is improved by Fast R-CNN. In the training stage, the locations are optimized, and the region proposals are classified. Specifically, the region of interest (ROI) pooling method is applied to sample the features. As a result, the improved pipeline of sequentially trained tasks is used to improve the Fast R-CNN training effect. Nevertheless, the region proposal generation speed is affected by using the SS method. This problem is solved by Faster R-CNN. The RPN method is developed to reduce the time of region proposal generation.

In a multiregion CNN (MR-CNN) [28], the bounding box regression scheme is used to improve the object detection ability. The semantic segmentation-aware features are extracted through a deep MR-CNN, and the bounding boxes are evaluated twice. The Inside-Outside Net (ION) [29] extracts the inside and outside the ROI information. Skip pooling [30] is applied to achieve multiple abstraction levels and scales. HyperNet [31] performs synthesized object detection and region proposal generation tasks. Then it obtains good object detection
results on the PASCAL VOC 2007 and 2012 datasets. The locations of objects in the image and their orientations on the ground plane are generated simultaneously by the FRCNN+Or [32] method. In the field of autonomous driving, Mono3D [33] is applied to perform 3D object detection from a single monocular image. The proposal generation step is the main improvement to Mono3D. A base feature extraction region is defined in the regionlet [34]. Mono3D organizes the regionlets into small groups with stable relative positions. Consistently, better joint object localization and viewpoint estimation are provided in DPM-VOC+VP [35] to improve the 3D object detection ability. Multilabel classification with partial labels [36] introduces a new classification loss that exploits the proportion of known labels per example. This approach allows the use of the same training settings as when learning with all the annotations. The stereo R-CNN [37] extends Faster R-CNN for stereo inputs to simultaneously detect and associate objects in left and right images. The sparse and dense semantic and geometric information in stereo imagery is fully exploited by this method. The generalized intersection over union (GIOU) [38] is designed to address the weaknesses of the IoU by introducing a generalized version as both a new loss and a new metric.

Recently, meta-heuristics algorithms, such as monarch butterfly optimization (MBO) [39], Slime mould algorithm (SMA) [40], Moth search algorithm (MSA) [41] and Harris Hawks Optimizer (HHO) [42], are popular to solve optimization problems. MBO is inspired by simplifying and idealizing the migration of monarch butterflies. The positions of the monarch butterflies are updated by offsprings and butterfly adjusting operator. SMA applies adaptive weights to simulate the process of producing positive and negative feedback of the propagation wave of slime mould based on its oscillation mode in nature. MSA uses the phototaxis and Levy flights of the moths to build up the general-purpose optimization method. HHO is inspired by the surprise pounce of Harris hawks. The algorithm mathematically mimics the dynamic patterns and behaviors witch based on the dynamic nature of scenarios and escaping patterns of the prey.

In the PBLS_SRPN [43], a particle swarm optimization (PSO) and bacterial foraging optimization (BFO)-based learning strategy (PBLS) is applied to improve Faster R-CNN. However, multiple pooling sizes are not considered. Additionally, the expression of the classification and regression loss function is not optimized jointly. A detailed comparison is shown in our improvements of this paper. Significant experimental results are achieved by our proposed method.

**Literature review**

**Grey wolf optimizer**

The GWO [44] is a metaheuristic optimization method [45]. The leadership hierarchy and hunting mechanism of grey wolves is mimicked by the GWO algorithm. The hunting process for grey wolves is shown in Fig 2.

Specifically, excellent results are obtained by this method in solving optimization problems. Each grey wolf is an optimization solution in the problem field. Four types of grey wolves, alpha (α), beta (β), delta (δ), and omega (ω) wolves, are employed to simulate the leadership hierarchy. α represents the most suitable solution. β and δ represent the second and third best solutions, respectively. The rest of the candidate solutions are presented as ω. Encircling prey, attacking prey and searching for prey are the three main hunting steps of the GWO. The mathematical model for the encircling behavior is proposed in the following expressions:

\[
\overrightarrow{D} = |\overrightarrow{C} \cdot \overrightarrow{X}_{p(t)} - \overrightarrow{X}_{o(t)}|
\]

\[
\overrightarrow{X}(t + 1) = \overrightarrow{X}_{p(t)} - \overrightarrow{A} \cdot \overrightarrow{D}
\]

where t represents the current iteration; the variables \( \overrightarrow{A} \) and \( \overrightarrow{C} \) are coefficient vectors; variable
$\mathbf{X}$ is the position vector of the prey; and $\mathbf{X}$ is the position vector of a grey wolf. The equations for $\mathbf{A}$ and $\mathbf{C}$ are expressed as follows:

$$\mathbf{A} = 2\mathbf{\bar{X}} \cdot \mathbf{r}_1 - \mathbf{\bar{X}}$$

$$\mathbf{C} = 2 \cdot \mathbf{r}_2$$

(2)

where the value of $\mathbf{\bar{X}}$ is linearly decreased from 2 to 0 as the algorithm is iterated and the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are in $[0, 1]$.

The expression for encircling prey is shown as follows:

$$\mathbf{D}_a = |\mathbf{C}_1 \cdot \mathbf{X}_a - \mathbf{x}|, \mathbf{D}_b = |\mathbf{C}_2 \cdot \mathbf{X}_b - \mathbf{x}|$$

$$\mathbf{D}_d = |\mathbf{C}_3 \cdot \mathbf{X}_d - \mathbf{x}|$$

$$\mathbf{X}_1 = \mathbf{X}_a - \mathbf{A}_1 \cdot (\mathbf{D}_a), \mathbf{X}_2 = \mathbf{X}_b - \mathbf{A}_2 \cdot (\mathbf{D}_b)$$

$$\mathbf{X}_3 = \mathbf{X}_d - \mathbf{A}_3 \cdot (\mathbf{D}_d)$$

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3}$$

(3)

The GWO has some variants, such as GWOCMALOL [46], IGWO [47], EGWO wrapped ELM [48], GWO-KELM [49]. The covariance matrix adaptation evolution strategy (CMAES), levy flight mechanism, and orthogonal learning (OL) are applied into GWOCMALOL to solve the structural defects and uncertain performance problem. Random local search around the optimal grey wolf in beta grey wolves, and random global search in omega grey wolves are introduced in IGWO to improve the stochastic behavior, and exploration capability of grey wolves. EGWO wrapped ELM combines the EGWO with the extreme learning machine (ELM) to identify the PQ poisoned patients. GWO-KELM adopts GWO to construct an effective kernel extreme learning machine (KELM) model for bankruptcy prediction. In this paper, the classic GWO is applied to our proposed methods. In the future, we will consider to apply other variants to improve the optimization ability.
Our approach

Overview
The pretrained 16-layer Visual Geometry Group (VGG-16) model is applied to sample the features of the input image. In addition, Conv1, Conv2, Conv3, Conv4 and Conv5 represent the convolutional layers of VGG-16 [50]. The general structure of BRPN is illustrated in Fig 3.

From Fig 3, we can see there are three improvements in our proposed method. First, a novel enhanced pooling network is designed to sample the features of the input image in this paper. Second, the softmax classifier is replaced by an SVM classifier. In addition, the GA is used to optimize the parameters of the SVM. Third, the expression of the classification and regression loss function is improved. In addition, the parameters of the loss function are optimized by the GWO.

A novel enhanced pooling network

Because only the last convolutional features are included in the output features for RPN, the lower feature information is not combined with the output information. Therefore, the small object detection ability of Faster R-CNN is not satisfactory. The synthesized feature sampling strategy is introduced in the enhanced pooling network (EPN).

From Fig 4, we can see that the output feature is combined with the different layers in our synthesized feature sampling strategy. Thereafter, the output feature includes the information of the lower and higher layers. Compared with PBLS_SRPN, the sampling ability of the low convolutional layer is enhanced by this improvement. In other words, the small object detection performance of Faster R-CNN is improved.

At the beginning of Faster R-CNN, the short side of the input image is changed to 600 pixels. However, the aspect ratio of the input image is not modified. As a result, the shape of the feature map is different. Therefore, the shape of the output feature map of VGG-16 is not fixed. Specifically, the width and length of feature maps vary greatly. Nevertheless, the scale for the RPN pooling method is fixed at 7×7. Therefore, the pooling method of the RPN is not suitable for different sizes of input objects.
As shown in Fig 5, we can add more ROI pooling sizes to the RPN; thus, the different shapes of the input objects can be processed very well. According to this concept, our EPN is developed for this work.

From Fig 6, we can see that two pooling methods of sizes $10 \times 4$ and $4 \times 10$ are applied in our EPN. The objects with heights far smaller than their widths are detected by $10 \times 4$ pooling. In addition, objects with widths far smaller than their heights are detected by $4 \times 10$ pooling. In this way, an irregular image can be detected effectively. Theoretically, we can use more pooling sizes to adapt to more shapes of input objects. However, it will cost more computation time. For a real object detection system, this is not acceptable. Moreover, different dimensions of pooling methods can be fused by using a flattened layer to make the pooled feature map into one-dimensional vectors. Considering the processing speed, the number of parameters is reduced by dimension reduction for each pooled feature map, and the vectors are sent to the fully connected layer by the concatenation layer.
Through the experimental results, we can see that the detection performance of our proposed method is improved. Thus, our improvements are effective.

A novel GA-SVM classifier

In this paper, the SVM classifier with a radial basis function (RBF) kernel is applied to distinguish the foreground region boxes and the background region boxes. The RBF kernel of the SVM contains parameters $C$ and $\gamma$. Nevertheless, there are no parameters for the softmax function. Therefore, the SVM algorithm has more optimization potential than the softmax method. In this paper, the GA-SVM with strong fitting ability is designed to replace the softmax method. From the experimental results, we can see that the performance of the SVM with fixed parameters is worse than that of the softmax method. However, the classification results of the GA-SVM are better than those of the softmax method. In other words, our improvements are effective.

To improve the performance of the SVM, we need to train $C$ and $\gamma$ of the RBF kernel. In this work, these two parameters are optimized by the GA. The relationship between the GA and the parameters are shown in Fig 7.

The parameters of the SVM are optimized by the GA. Therefore, parameters $C$ and $\gamma$ are represented as a gene. In other words, our target is to find the best gene. The parameters $C$ and $\gamma$ are generated by the two dimensions of the gene. The fitness function is used to evaluate the performance of each gene. The values of $C$ and $\gamma$ are updated through the calculated results of the fitness function. The parameters of the SVM optimization process by the GA are shown in Fig 8.

The GA and SVM parameter are randomly initialized at the beginning of the optimization process. The SVM is trained by the anchor boxes that are generated by the BRPN. In addition, the number of positive samples must equals the number of negative samples. Because, in general, there are fewer positive samples than negative samples, some negative samples are used to complement the positive samples. The gene is updated according to the fitness calculation results. If the maximum number of iterations of the GA is reached or the fitness value satisfies
the threshold, then the optimal values of parameters $C$ and $\gamma$ have been obtained; $[0.01, 35,000]$ and $[0.0001, 32]$ are the ranges of parameters $C$ and $\gamma$, respectively. In the PBLA_SRPN, the parameters of the SVM are optimized by BFO. Because the computational complexity of the BFO algorithm is high, the training time is affected. In this paper, the GA is applied to optimize the SVM.

Because the values of $C$ and $\gamma$ are randomly initialized at the start of the optimization process, the number of iterations of the GA is too high. In other words, the values of parameters $C$ and $\gamma$ converge after many optimization iterations. The operation time of the BRPN should be slowly decreased. The iteration expression of the GA is described as follows:

$$n_{\text{iteration\_GA}} = N_{\text{iteration\_GA\_total}} \times \frac{M_{\text{iteration\_BRPN\_total}} - m_{\text{iteration\_BRPN}}}{M_{\text{iteration\_BRPN\_total}}}$$  (4)

where $n_{\text{iteration\_GA}}$ is the GA operation time for one iteration of the BRPN; $N_{\text{iteration\_GA\_total}}$ is the total operation time of the GA; $m_{\text{iteration\_BRPN}}$ is the current operation time of the BRPN; and $M_{\text{iteration\_BRPN\_total}}$ is the total operation time of the BRPN.

From Eq (4), we can see that $m_{\text{iteration\_BRPN}} = 1$ at the start of the BRPN. Then, the value of $\frac{M_{\text{iteration\_BRPN\_total}} - m_{\text{iteration\_BRPN}}}{M_{\text{iteration\_BRPN\_total}}}$ is close to 1. The variable $n_{\text{iteration\_GA}}$ approximately equals $N_{\text{iteration\_GA\_total}}$. The value of $m_{\text{iteration\_BRPN}}$ is increased while the value of $n_{\text{iteration\_GA}}$ decreases. Thereafter, the GA iteration times are small at the end of the training phase. In other words, the training time of the GA-SVM is effectively reduced. Therefore, Eq (4) satisfies our requirement.
Improved classification and regression loss function

The classification loss function $L_{cls}(p_i; p^*_i)$ of Faster R-CNN is not designed rationally. Detailed information is presented in the following paragraphs. The equation for the classification loss $L_{cls}$ is as follows:

$$L_{cls}(p_i; p^*_i) = p^*_i \log(p_i) - (1 - p^*_i) \log(1 - p_i)$$  \hspace{1cm} (5)

The ground-truth label $p^*_i$ equals 1 if the anchor box is positive and 0 if the anchor box is negative. If the ground-truth label $p^*_i$ equals 1, then the $i$th anchor box is classified correctly. Therefore, the value of $-\log(p_i)$ is small. Moreover, if the ground-truth label $p^*_i$ equals 0, the value of $-\log(1 - p_i)$ is large. Sometimes the value of $-\log(1 - p_i)$ is not much larger than that of $-\log(p_i)$. Nevertheless, the number of positive samples is far less than that of negative samples in general. Therefore, the training effect of the negative samples should be more than that of the positive samples. In this paper, the classification loss function is improved to reduce the effect of the positive samples. The improved classification loss function is developed as follows:

$$L_{cls}(p_i; p^*_i) = e^{-\beta p_i} \log(p_i) - e^{-\beta p^*_i} (1 - p_i) \log(1 - p_i)$$  \hspace{1cm} (6)

where the coefficient $\eta$ is applied to adjust the function.
If $p_i^* = 0$, then the value of $e^{-\eta(p_i^*)}$ is greater than 1. Therefore, the training effect of the negative samples is increased. Moreover, if $p_i^* = 1$, then the value of $e^{-\eta(p_i^*)}$ is lower than 1. Thus, the training effect of positive samples is seriously reduced. As a result, the negative samples are learned more than the positive samples. In other words, the training performance of the RPN is enhanced. From Fig 9, we can see that the curve of $e^{-\eta p_i}$ is affected by different values of $\eta$. Therefore, the value of the coefficient $\eta$ is important to the training performance. However, the coefficient $\eta$ is hard to design. In this paper, this coefficient is optimized by the swarming intelligence function. The definition of the regression loss function is as follows:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$  \hspace{1cm} (7)$$

The smooth $L_1$ loss function is constructed based on the $L_1$ and $L_2$ loss functions. The $L_1$ and $L_2$ loss functions are defined as follows:

$$L_1(x) = \lambda |x|$$
$$L_2(x) = \theta x^2$$  \hspace{1cm} (8)$$

where $\lambda$ and $\theta$ are the influencing parameters. From Eq (8), we can see that the descending speed of the $L_2$ loss function is slower than that of the $L_1$ loss function. As a result, the convergence effect of the $L_1$ loss is better than that of the $L_2$ loss. The smooth $L_1$ loss function is represented by the $L_2$ loss function between -1 and 1, while the smooth $L_1$ loss function is the $L_1$ loss function in other areas. Thereafter, the features of the $L_1$ and $L_2$ loss functions are included in the smooth $L_1$ loss function. The smooth $L_1$ loss function is a special case of the Huber loss function. Specifically, the $L_2$ loss function plays an important role if the value of $x$ is in the range of $[-\lambda, \lambda]$. However, the $L_1$ loss function occupies a leading position in other ranges. Therefore, the calculation effectiveness of the Huber loss function depends on the value of $\lambda$. Additionally, 0.5 may not be the optimal value; 0.5 is represented by the variable $\theta$. The improved loss function is described as follows:

$$H(x) = \begin{cases} \theta x^2 & \text{if } |x| < \lambda \\ \lambda(2\theta|x| - \theta \lambda) & \text{otherwise,} \end{cases}$$  \hspace{1cm} (9)$$

where $\lambda$ and $\theta$ are the influencing factors. The parameters $\lambda$ and $\theta$ can be assigned by the swarming intelligence function.

Consequently, the performance of the classification loss function is enhanced by strengthening the negative sample training effect. Additionally, the coefficients of the
classification and regression loss function are suitable to be trained by the swarming intelligence function.

**Loss function parameters optimization with the GWO**

The factors $\lambda$ and $\theta$ are set to 1 and 0.5 in the regression loss function based on experience. Nevertheless, these values are not globally optimal. In addition, the improved classification loss function contains the factor $\eta$. Finding the best values for influence factors $\lambda$, $\theta$ and $\eta$ is not easy. The grid search method is the conventional way to find the best values of $\lambda$, $\theta$ and $\eta$. However, the value of the search step is difficult to define in the grid search method. If the value of the search step is not suitable, then the operation results are not optimal. In other words, the exploitation ability is affected by the search step.

In this paper, the GWO is introduced to optimize the influencing factors. The loss function coefficients are determined by two types of parameters. Each grey wolf is defined by three parameters, $\lambda$, $\theta$ and $\eta$. Therefore, the changes in parameters $\lambda$, $\theta$ and $\eta$ are represented by grey wolves. In addition, the ability of each grey wolf is calculated by the fitness function. In this work, our improved loss function is the fitness function. The leaders are called $\alpha$ and include males and females. The $\alpha$ wolves are mostly responsible for making decisions about hunting, finding a place for sleeping, determining the time to wake, and so on. The second level in the hierarchy of grey wolves is $\beta$. The $\beta$ wolves are subordinate wolves that help the $\alpha$ wolf with decision-making and other pack activities. The lowest ranking grey wolves are $\omega$ wolves. The $\omega$ wolves play the role of scapegoats. The last type of wolves is $\delta$ wolves, which are subordinate to $\alpha$ and $\beta$ wolves but dominate $\omega$ wolves. Because the best solution is $\alpha$, to find the optimal $\lambda$, $\theta$ and $\eta$ is to find $\alpha$. The fitness function is a way to measure the wolves. The relationship between the GWO and the loss function coefficients is shown in Fig 10.

![GWO method](https://doi.org/10.1371/journal.pone.0251339.g010)
First, the parameters $\lambda$, $\theta$ and $\eta$ are initialized. The default values of $\lambda$, $\theta$ and $\eta$ in the loss function are 1, 0.5 and 1 at the beginning. The optimal grey wolves $\alpha$ are calculated and updated by the GWO. The best values are processed based on the fitness function. The optimization flow is finished when the operation stop condition of the GWO is satisfied. The flowchart in Fig 11 represents the BRPN parameter optimization process through the GWO.

In the PBLS_SRPN, the GA optimization process is contained in the PSO process. Thereafter, the training procedure is complicated. Nevertheless, the SVM and loss function optimization procedures are serially executed by the GA and GWO. In other words, the training process is simplified. As a result, the training time is decreased.
Experiments

Datasets

The PASCAL VOC 2007, 2012 [51] and KITTI [52] datasets are applied to train and test our proposed method. The comparison between our proposed method and the state-of-the-art object detection methods is shown in this part. The information on our experimental datasets is presented in Table 1. Moreover, the performance of the BRPN on object detection is verified.

The Caffe [53] framework is used to realize our proposed method. The convolutional parameters are initialized through the pretrained VGG-16 model. The VGG-16 model contains 13 convolutional layers and 3 fully connected layers. The performance of our proposed method is evaluated based on the mAP and recall. To ensure the stability of the experimental results, we repeat each method 3 times and average the results.

Parameter setting

The GA, GWO and BRPN parameters used during our experiments are shown in the Tables 2–4.

Experiments on the PASCAL VOC 2007 dataset

In the experiments, the proposed method named BRPN-GE only contains the EPN strategy and GA-SVM classifier improvements. The final proposed method named BRPN is based on BRPN-GE. Additionally, it includes a strengthened loss function with the parameters is optimized by GWO.

Table 1. Dataset information.

| No. | Dataset          | No. of categories | No. of annotated objects |
|-----|------------------|-------------------|--------------------------|
| 1   | PASCAL VOC 2007  | 20                | 24,640                   |
| 2   | PASCAL VOC 2012  | 20                | 27,450                   |
| 3   | KITTI            | 3                 | 80,256                   |

https://doi.org/10.1371/journal.pone.0251339.t001

Table 2. GA parameters.

| Parameter | Description         | Value |
|-----------|---------------------|-------|
| N         | Number of genes     | 30    |
| N_{IN}    | Number of iterations| 100   |
| N_{C}     | Crossover rate      | 0.8   |
| N_{M}     | Mutation rate       | 0.05  |

https://doi.org/10.1371/journal.pone.0251339.t002

Table 3. GWO parameters.

| Parameter | Description                  | Value  |
|-----------|------------------------------|--------|
| α         | Initial value of the GWO parameter | 2      |
| r{sub}1   | Random factor for the GWO     | [0, 1] |
| r{sub}2   | Random factor for the GWO     | [0, 1] |

https://doi.org/10.1371/journal.pone.0251339.t003
In Table 5, the mAP of the BRPN-GE model is 78.2%, which is higher than that of the Fast R-CNN, Faster R-CNN, ION and HyperNet models. The reason includes as follows. Firstly, the output feature maps of BRPN-GE contain lower feature maps and higher feature maps. And multiple pooling sizes are added to the BRPN-GE. Therefore, the BRPN-GE model can adapt to objects with different shapes. The feature sampling ability for our proposed method is boosted through EPN improvement. Secondly, the classification capability is promoted based on the GA-SVM method. From the experimental results, improvements in the BRPN-GE model are effective.

Additionally, the mAP of the BRPN model (79.1%) is the best. Compared with the BRPN-GE model, this is benefit from the strengthened loss function which the parameters are optimized by GWO. Outstanding results are achieved by the BRPN on bikes, birds, boats, bottles, cats, chairs, dogs, horses, motorbikes, plants, sofas and trains. In other words, the BRPN can classify objects more accurately than the other comparison methods. The classification ability of the BRPN is improved based on our modifications.

In Table 5, the mAP of the BRPN-GE model is 78.2%, which is higher than that of the Fast R-CNN, Faster R-CNN, ION and HyperNet models. The reason includes as follows. Firstly, the output feature maps of BRPN-GE contain lower feature maps and higher feature maps. And multiple pooling sizes are added to the BRPN-GE. Therefore, the BRPN-GE model can adapt to objects with different shapes. The feature sampling ability for our proposed method is boosted through EPN improvement. Secondly, the classification capability is promoted based on the GA-SVM method. From the experimental results, improvements in the BRPN-GE model are effective.

Additionally, the mAP of the BRPN model (79.1%) is the best. Compared with the BRPN-GE model, this is benefit from the strengthened loss function which the parameters are optimized by GWO. Outstanding results are achieved by the BRPN on bikes, birds, boats, bottles, cats, chairs, dogs, horses, motorbikes, plants, sofas and trains. In other words, the BRPN can classify objects more accurately than the other comparison methods. The classification ability of the BRPN is improved based on our modifications.

Table 4. BRPN parameters.

| 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 the dropped learning rate                              | 0.1       |
| size         | Drop the learning rate every stepsize iterations                 | 50,000    |
| momentum     | Weight of the previous update                                    | 0.9       |
| weight_decay | Regularization factor                                            | 0.0005    |
| max_iter_rpn | Number of BRPN execution steps                                   | 70,000    |
| IoU_foreground | IoU for the BRPN foreground proposals                            | [0.6, 1]  |
| IoU_background | IoU for the BRPN background proposals                            | [0, 0.3]  |
| α            | Factor for the improved loss function                            | 1         |
| β            | Factor for the improved loss function                            | 0.5       |
| η            | Factor for the improved loss function                            | 1         |
| NMS_IoU      | Threshold value of the IoU method                                | 0.65      |

https://doi.org/10.1371/journal.pone.0251339.t004

Table 5. Detection results on the PASCAL VOC 2007 test set.

| Approach  | mAP | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | motorbike | 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|
| 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|
| BRPN-GE   | 78.2 | 78.4 | 83.4 | 78.4 | 68.9 | 68.4   | 86.1| 86.1| 87.8| 61.7  | 83.4| 74.4  | 86.3| 85.9  | 84.9     | 78.2   | 51.3  | 77.1  | 76.2 | 85.7  | 82.0|
| BRPN      | 79.1 | 79.6 | 84.7 | 79.3 | 69.9 | 69.5   | 87.5| 86.5| 88.1| 62.3  | 84.5| 74.8  | 87.7| 86.8  | 85.8     | 79     | 52.4  | 78.8  | 77.1 | 85.9  | 82.8|

The best AP of each object category and mAP are highlighted in bold.

https://doi.org/10.1371/journal.pone.0251339.t005
Experiments on the PASCAL VOC 2012 dataset

Our proposed BRPN-GE and BRPN are trained and tested on the PASCAL VOC 2012 dataset. Additionally, the VOC 2007 and VOC 2012 datasets are integrated as training data. In Table 6, a mAP of 74.3% is achieved by the BRPN-GE model. The mAPs of the Fast R-CNN, Faster R-CNN, ION, MR-CNN and HyperNet models are lower than that of the BRPN-GE model. However, the mAP of the BRPN model is higher than that of the BRPN-GE and PBLS_SRPN models. The PBLS_SRPN model achieves the highest AP on buses, cars and people. Furthermore, the Fast R-CNN model achieves the highest AP of 89.3% on cats. Obviously, the highest APs are obtained by the BRPN model on the remaining objects. In other words, our proposed BRPN model can achieve the best object detection results on the PASCAL VOC 2012 dataset. Therefore, the BRPN model is adaptive to sophisticated datasets.

Small-object detection

As we know, small objects are composed of few pixels; therefore, small objects are difficult to detect. In the VOC 2007 and VOC 2012 dataset, birds, bottles and potted plants are small objects. Thus, our proposed methods are applied to detect these challenging objects.

From Fig 12, we can see that the APs of the Fast R-CNN and Faster R-CNN models on detecting bottles, birds and potted plants are lower than that of the BRPN-GE model. The reason is that the lower feature maps and higher feature maps are synthesized with the output feature maps in the BRPN-GE model. Additionally, multiple pooling sizes are supplemented with the BRPN-GE method; therefore, the capability of small-object feature sampling of the BRPN-GE model is boosted. Furthermore, the BRPN achieves the highest AP in detecting small-object detection results on the PASCAL VOC 2007 and VOC 2012 test sets.

The best AP of each object category and mAP are highlighted in bold.

https://doi.org/10.1371/journal.pone.0251339.t006

Fig 12. Small-object detection results on the PASCAL VOC 2007 and VOC 2012 test sets.

https://doi.org/10.1371/journal.pone.0251339.g012
birds, bottles and potted plants. The softmax classifier is replaced by the GA-SVM classifier with the RBF kernel, which strengthens the classification ability of the BRPN model. In addition, the GWO is applied to optimize the parameters of the classification loss functions. As a result, the classification ability of the BRPN model is further improved. In other words, the small-object detection performance of our proposed methods is enhanced.

Analysis of the recall vs. IoU and ROC curves

The current well-known object detection method is compared with our proposed method. In total, 150, 450 and 850 region proposals are applied to the BRPN. The different IoU values are applied to calculate the recall of the region proposals on the PASCAL VOC 2007 dataset. The chosen region proposals are ranked according to the scores from high to low. From Fig 13, we can see that the recall value of the MR-CNN, ION, HyperNet, Fast R-CNN and Faster R-CNN models drop quickly as the number of region proposals decreases. The quality of the region proposals in these object detection methods is poor because of their feature sampling strategy. Features from different levels are not considered in feature sampling. Therefore, the recall values are affected at the same IoU value. Multiple-layer features are synthesized based on our EPN method. Additionally, the number of pooling sizes of the BRPN is increased. Therefore, the feature sampling ability of our proposed method is enhanced. Furthermore, the foreground and background region proposals are distinguished through the GA-SVM classifier. As a result, only 150 region proposals are sufficient for our proposed method to generate valid region proposals. As the number of region proposals is reduced from 850 to 150, the recall of our proposed method is basically unchanged. The recall of our BRPN model is high, and the number of region proposals is equal to 150 when the IoU value is assigned to 0.65. Therefore, the value of the IoU threshold for the BRPN is set to 0.65. Because the object detection speed is affected by the number of region proposals, the computing speed of the BRPN can be accelerated with fewer region proposals.

The receiver operating characteristic (ROC) curves for the 3 best schemes and the Faster R-CNN model are presented in Fig 14. It can be seen that the BRPN, BRPN-GE and MR-CNN models consistently outperform the Faster R-CNN model. Furthermore, the BRPN-GE model achieves the best experimental results.

Analysis of the novel EPN and training process

In our proposed EPN improvement, more ROI pooling sizes are added to the BRPN; thus, the different shapes of input objects can be processed effectively. Experiments over different

![Fig 13. Recall versus IoU threshold on the VOC 2007 test set. Left: 150 region proposals. Middle: 450 region proposals. Right: 850 region proposals.](https://doi.org/10.1371/journal.pone.0251339.g013)
pooling sizes are executed on the VOC 2007 dataset. From Table 7, we can see that a mAP of 78.6% is achieved by the BRPN model with a 7×7 pooling size. In this situation, the scale for the BRPN pooling method is fixed at 7×7. Therefore, the pooling method of the BRPN is not suitable for different sizes of input objects. Moreover, a mAP of 78.9% is achieved by the BRPN model with 7×7 and 10×4 pooling sizes. Therefore, objects with heights far smaller than their widths are detected by a pool size of 10×4. In other words, the detection capability of the BRPN is strengthened. Finally, a mAP of 79.1% is achieved by the BRPN with 7×7, 10×4 and 4×10 pooling sizes. The reason is that objects with widths far less than their heights are detected by the pool size of 4×10. As a result, the BRPN with three pooling sizes achieves the best results. In other words, the feature sampling capability of the BRPN is promoted by the novel EPN.

The curve of the classification loss function is affected by different values of $\eta$. In addition, $\lambda$ and $\theta$ are the influencing factors for the improved regression function. Therefore, the values of coefficients $\eta$, $\lambda$ and $\theta$ are optimized by the GWO. To reduce the training time, the ranges of $\eta$, $\lambda$ and $\theta$ are restricted to (0, 2). In this section, the fluctuations of $\eta$, $\lambda$ and $\theta$ in the VOC 2007 dataset are illustrated as follows.

The initial values of $\lambda$, $\theta$ and $\eta$ in the loss function are 1, 0.5 and 1, respectively. These coefficients are optimized by the GWO. Each grey wolf is defined by three parameters $\lambda$, $\theta$ and $\eta$. Therefore, the optimal solution for these parameters is to find the wolf leaders. The fitness function is a way to measure the wolves. The optimal grey wolves $\alpha$ can be calculated and

![Fig 14. ROC curves of the compared methods.](https://doi.org/10.1371/journal.pone.0251339.g014)

**Table 7. Experimental results over the novel enhanced pooling network.**

| Pooling size | mAP | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | motorbike | person | plant | sheep | sofa | train | tv |
|--------------|-----|------|------|------|------|--------|-----|-----|-----|-------|-----|-------|-----|-------|-----------|--------|-------|-------|------|-------|----|
| 7×7          | 78.6| 79.1 | 84.2 | 78.2 | 69.6 | 69.2   | 87.1| 85.9| 87.2| 62.1   | 83.8| 74.1   | 86.8| 86.6  | 85.5      | 78.4   | 52.2  | 78.5  | 76.1 | 85.5  | 82.4|
| 7×7,10×4     | 78.9| 79.3 | 84.5 | 78.7 | 69.8 | 69.4   | 87.2| 86.3| 87.8| 62.2   | 84.4| 74.6   | 87.4| 86.7  | 85.5      | 78.9   | 52.4  | 78.6  | 76.7 | 85.6  | 82.6|
| 7×7,10×4, 4×10| 79.1| 79.6 | 84.6 | 79.1 | 69.9 | 69.5   | 87.5| 86.5| 88.1| 62.3   | 84.5| 74.8   | 87.7| 86.8  | 85.7      | 79     | 52.4  | 78.8  | 77.1 | 85.9  | 82.8|

[https://doi.org/10.1371/journal.pone.0251339.t007](https://doi.org/10.1371/journal.pone.0251339.t007)
updated in the GWO. From Fig 15 and Table 8, we can see that the values of $\lambda$ and $\eta$ are reduced when the number of iterations = 100. When the maximum number of iterations is between 100 and 200, these values are larger than before. The reason is that the GWO gradually finds better wolves from the lower-level wolves. The value of $\theta$ is increased before the number of iterations = 200 and then decreased when number of iterations is between 200 and 600. This fluctuation represents the process of finding the wolf leaders. Specifically, the values of $\lambda$, 

Table 8. Runtime data for the different coefficients.

| Experimental Results          | $\eta$ | $\lambda$ | $\theta$ | AP  |
|-------------------------------|--------|-----------|----------|-----|
| Number of iterations = 0      | 0.5    | 0.5       | 1        | 0   |
| Number of iterations = 200    | 0.73   | 0.77      | 1.22     | 42.50% |
| Number of iterations = 400    | 1.47   | 1.26      | 1.1      | 67.40% |
| Number of iterations = 600    | 1.68   | 1.39      | 0.82     | 78.90% |
| Number of iterations = 800    | 1.69   | 1.41      | 0.79     | 79.10% |

https://doi.org/10.1371/journal.pone.0251339.t008
\(\theta\) and \(\eta\) are stable in the last iteration phases, and the global optimal solution is found. Additionally, we can see that the AP continues to increase. Based on the experimental results, we can conclude that the GWO is effective in improving the classification ability of the BRPN.

From Table 9, we can see that the training error of the Faster R-CNN model is 0.173 when it is trained 160,000 times. However, the training error for the BRPN is 0.08 when it is trained 800 times. Obviously, the training error of the BRPN model is better than that of the Faster R-CNN model. The reason is that the feature sampling ability of the BRPN is enhanced by introducing the synthesized feature sampling strategy. Therefore, the quality of the input data for the BRPN is strengthened. Additionally, the classification and regression loss function of the BRPN is improved. Therefore, the effect of backpropagation for the BRPN and the performance of the BRPN are improved.

**Analysis of the novel loss function**

In this part, the VOC 2007 and VOC 2012 datasets are used to show the effect of our novel loss function. The experimental results of the three situations are compared in Table 10.

Only the classification loss \((\lambda = 0)\) in the loss function is presented in the first column of each group. Thus, the worst results are achieved without the bounding box regression loss function for the Faster R-CNN model. The classification loss and bounding box regression loss of the RPN are shown in the second column of each group. Because the quality of the region proposals is enhanced based on the bounding box regression loss function, the processing results of the second column for each group are better than those of the first column. In the third column of each group, the classification loss function of the BRPN is improved to reduce the effect of positive samples. As a result, the negative samples are learned more than the positive samples, and the training performance of the BRPN is enhanced. In addition, more coefficients are added to the regression loss function. Moreover, the parameters of the loss function are optimized by our GWO. Therefore, the best results are achieved by the BRPN in the third column.

**The effect of improved classification method**

In this section, the optimization ability of the GA-SVM method is presented. The experimental results of the GA-SVM, SVM and softmax classifiers based on the BRPN with only EPN improvement (BRPN-EPN) are shown. These experiments are performed on the PASCAL VOC 2007 test set. The BRPN-EPN model achieves a mAP of 77.2\% with the softmax classifier (Table 11). Moreover, the BRPN-EPN with the SVM classifier achieves a mAP of 76.9\%. We
can see that this value is 0.3 percentage points lower than that of the BRPN with the softmax classifier. The parameters $C$ and $\gamma$ are the key factors of the RBF kernel of the SVM. In general, the grid search method is used to find the suitable values of $C$ and $\gamma$. However, this optimization method usually cannot find the global optimization values. Therefore, the classification ability of the SVM cannot meet our requirements. The GA method is applied to optimize $C$ and $\gamma$ to improve the classification performance of the SVM. We find that the mAP of the BRPN-EPN with GA-SVM classifier is 78.2%, which is the best in Table 11. Therefore, our proposed GA-SVM is effective.

The computational complexity of the SVM classifier is $O(n^3)$. Additionally, the calculation complexity of the softmax classifier is $O(n)$. Therefore, the computation time of the SVM classifier is greater than that of the softmax classifier. The average computational time of the Faster R-CNN model with the softmax classifier is approximately 0.5 seconds. However, the computational time of the Faster R-CNN model with the SVM classifier is approximately 200 seconds. Nevertheless, the choice of the classifier mainly affects the training time. The test time for our improved method is not significantly affected. Therefore, our method can be applied to real-time classification applications. Based on the experimental results, we can see that the classification capability of the BRPN model with the GA-SVM classifier is strengthened.

### Experiments on the KITTI dataset

The BRPN-GE, BRPN and other compared algorithms are tested on the KITTI autonomous driving dataset in this section. The KITTI dataset includes approximately 7481 training images and 7518 test images. Additionally, the KITTI dataset includes three object categories: cars, pedestrians and cyclists. The images are sampled from autonomous driving vehicles. Furthermore, easy, moderate and hard evaluation levels are provided in this dataset. The moderate level is used in general. Specifically, the pedestrian and cyclist data need 50% overlap. In addition, the car data requires 70% overlap. The AP value is applied to measure the object detection method capability. Table 12 shows the experiment results.

| Method       | Cars Easy | Cars Moderate | Cars Hard | Pedestrians Easy | Pedestrians Moderate | Pedestrians Hard | Cyclists Easy | Cyclists Moderate | Cyclists Hard |
|--------------|-----------|---------------|-----------|------------------|----------------------|------------------|---------------|-------------------|---------------|
| FRCNN+Or     | 89.87     | 78.95         | 68.97     | 71.18            | 56.78                | 52.86            | 70.05         | 57.37             | 51.00         |
| Mono3D       | 90.27     | 87.86         | 78.09     | 77.30            | 66.66                | 63.44            | 75.22         | 63.85             | 58.96         |
| Faster R-CNN | 87.9      | 79.11         | 70.19     | 78.35            | 65.91                | 61.19            | 71.41         | 62.81             | 55.44         |
| Regionlets   | 86.50     | 76.56         | 59.82     | 72.96            | 61.16                | 55.22            | 70.09         | 58.69             | 51.81         |
| DPM-VOC+VP   | 80.45     | 66.25         | 49.86     | 59.60            | 44.86                | 40.37            | 43.65         | 31.16             | 28.29         |
| Occlusion Detector | 79.2 | 56.8          | 38.5      | 56.4             | 42.5                 | 35.9             | 61.3          | 47.3              | 36.4          |
| PBL_SRPN     | 89.7      | 88.6          | 78.6      | 82.4             | 70.5                 | 65.1             | 78.72         | 70.3              | 61.6          |
| BRPN-GE      | 89.1      | 88.5          | 80.7      | 81.8             | 70.2                 | 64.2             | 77.9          | 69.8              | 61.2          |
| BRPN         | **90.4**  | **89.9**      | **81.2**  | **82.7**         | **71.3**             | **65.5**         | **78.8**      | **70.4**          | **61.9**      |

The best result is highlighted in bold.
The moderate category values are the most representative test results. From Table 12, we can see that the moderate category values for our proposed BRPN-GE model are better than those of the FRCNN+Or, Mono3D, Faster R-CNN, regionlets, Occlusion Detector and DPM-VOC+VP models on the car, pedestrian and cyclist data. The output features of the BRPN-GE model contain lower-level and higher-level features. In addition, different pooling sizes are added to the BRPN-GE. Therefore, the quality of the region proposals is strengthened. Specifically, the classifier is enhanced by introducing the GA-SVM classifier. Moreover, the classification and regression loss function is improved. In the BRPN, the GWO is applied to optimize the coefficients of the novel loss function. From the results, we can see that the moderate category values for the BRPN are the best. In other words, the improvements in the BRPN are effective.

Runtime analysis

From Table 13, we can see that the detection speed of the BRPN is faster than that of the other methods. The reason is that fewer region proposals are generated by the BRPN. Based on a single NVIDIA Titan X GPU, our BRPN-based Faster R-CNN system has a frame rate of 5.2 fps for VGG-16. Consequently, our BRPN model achieves an outstanding detection speed.

Conclusion

The experimental results of the BRPN-GE model are better than those of the Faster R-CNN model. The reason is that the feature sampling ability is improved by the EPN and the classification capability is enhanced by the GA-SVM classifier. Moreover, the best test results are achieved by the BRPN model. The classification loss function of the BRPN is improved when training on the negative samples. Additionally, more parameters are added to the regression loss function of the BRPN. Furthermore, the GWO is applied to optimize the parameters of the improved loss function. In other words, our proposed BRPN is effective for object detection in UGVs. However, processing steps are increased for introducing the optimization method in the training phase. Therefore the training time is affected compared to the original method. To solve this drawback, we will focus on improving the optimization technique to reduce the training time.

Supporting information

S1 File. Datasets description. The details of experimental datasets are described in the separated S1 File.

Author Contributions

Conceptualization: Gang Wang.
Writing – original draft: Qian Xu, Gang Wang.
Writing – review & editing: Qian Xu, Ying Li, Ling Shi, Yaxin Li.
References

1. B. Alexe, T. Deselaers, and V. Ferrari, “What is an object?” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June. 2010, vol. 23, no. 3, 73–80.

2. X. Ren, and D. Ramanan, Histograms of sparse codes for object detection. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2013, 9 (4), 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), June. 2014, pp. 2155–2162.

4. Z. Yuting, S. Kihyuk, V. Ruben, P. Gang, 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. Schmid C. and Mohr R., “Local gray value invariants for image retrieval,” IEEE Trans. Pattern Anal. Mach. Intell., May. 1997, vol. 19, no. 5, pp. 530–534.

6. M. Weber, M. Welling, and P. Perona, “Unsupervised learning of models for recognition,” Eur.Conf. Comput. Vis., Jun. 2000, pp. 391–405.

7. P. F. Felzenszwalb, and D. Huttenlocher, “Efficient matching of pictorial structures,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June. 2000, pp. 66–75.

8. MM. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, “Bing: Binarized normed gradients for objectness estimation at 300fps,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June. 2014, pp. 3286–3293.

9. C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” Eur.Conf. Comput. Vis., Sep. 2014, pp. 91–110.

10. A. Ghodrati, M. Pedersoli, T. Tuytelaars, A. Diba, and L. V. Gool, “Deepproposal: Hunting objects by cascading deep convolutional layers,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Feb. 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., Dec. 2015, pp. 2479–2487.

12. Lowe D. G., “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., Jan. 2004, vol. 60, no. 2, pp. 91–110.

13. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June. 2005, pp. 886–893.

14. P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2008, pp. 1–8.

15. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Yann. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks. International Conference on Learning Representations, 2014.

16. Ciresan D. C., Meier J., and Schmidhuber J., Multi-column deep neural networks for image classification. Computer Vision & Pattern Recognition, 2012, pp. 3642–3649.

17. Y. Hua, K. Alahari, and C. Schmid, “Online object tracking with proposal selection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Dec. 2015, pp. 3092–3100.

18. Hosang J., Benenson R., Dollár P., and Schiele B., “What makes for effective detection proposals?” IEEE Trans. Pattern Anal. Mach. Intell., Apr. 2016, vol. 38, no. 4, pp. 814–830. https://doi.org/10.1109/TPAMI.2015.2465908 PMID: 26959679

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 Anal., 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. R. Girshick, “Fast R-CNN: Towards realtime object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., Jun. 2017, vol. 39, no. 6, pp. 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031 PMID: 27295650

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

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

24. LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., et al. “Backpropagation applied to handwritten zip code recognition,” Neural Comput., Dec. 1989, vol. 1, iss. 4, pp. 541–551.

25. C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.
26. Ren S., He K., Girshick R., Zhang X., and Sun J., “Object detection networks on convolutional feature maps,” IEEE Trans. Pattern Anal. Mach. Intell., Jul. 2017, vol. 39, no. 7, pp. 1476–1481. https://doi.org/10.1109/TPAMI.2016.2601099 PMID: 27541490

27. Torralba A., “Contextual priming for object detection,” Int. J. Comput. Vis., July. 2003, vol. 53, no. 2, pp. 169–191.

28. 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.

29. 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.

30. He K., Zhang X., Ren S., and Sun J., Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37 (9), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824 PMID: 26353135

31. 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.

32. Carlos G., Martin D., and Maria J., “Fast Joint Object Detection and Viewpoint Estimation for Traffic Scene Understanding”, IEEE Intel. Transp. Sy, 2018.

33. X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, “Monocular 3D object detection for autonomous driving,” IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 2147–2156.

34. Wang X.Y., Yang M., Zhu S.H., and Lin Y.Q., “Regionlets for Generic Object Detection”, IEEE Trans. Pattern Anal. Mach. Intell, 2015, pp. 2071–2084. https://doi.org/10.1109/TPAMI.2015.2389830 PMID: 26353185

35. Pepik B., Stark M., Gehler P., and Schiele B., “Multi-view and 3D deformable part models,” IEEE Trans, Pattern Anal. Mach. Intell, 2015, pp. 2232–2245. https://doi.org/10.1109/TPAMI.2015.2408347 PMID: 26440264

36. H. Rezatofighif, N. Tsoi, J.Y. Gwak, A. Sadeghian, I. Reid and S. Savarese, “Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June. 2019.

37. T. Durand, N. Mehrasa and G. Mori, “Learning a Deep ConvNet for Multi-label Classification with Partial Labels,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June. 2019.

38. PL. Li, XZ. Chen and SJ. Shen, “Stereo R-CNN based 3D Object Detection for Autonomous Driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June. 2019.

39. Wang G., Deb S., Cui Z., Monarch butterfly optimization, Neural Computing and Applications, Jul. 2019, vol. 31, no. 7, pp. 1995–2014.

40. Li S., Chen H., Wang M., Heidari A. A., Mirjalili S., Slime mould algorithm: A new method for stochastic optimization, Future Generation Computer Systems, Oct. 2020, vol. 111, pp. 300–323.

41. Wang G., Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Memetic Computing, Jun. 2018, vol. 10, no. 2, pp. 151–164.

42. Heidari A. A., Mirjalili S., Faris H., Aljarah I., Mafarja M., Chen H., Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems, Aug. 2019, vol. 97, pp. 849–872.

43. Wang Gang, Guo Jingming, Chen Yupeng, Li Ying and Xu Qian. “A PSO and BFO-based Learning Strategy applied to Faster R-CNN for Object Detection in Autonomous Driving,” IEEE ACCESS, 2019, vol 7, pp. 14908–1492.

44. Seyedali M., Seyed M., and Andrew L., “Grey Wolf Optimizer”, Advances in Engineering Software, 2014, pp. 46–61.

45. S. Nowozin, “Optimal Decisions from Probabilistic Models: The Intersection-over-Union Case”, IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Columbus, OH, USA, 2014.

46. Hu J., Chen H., Heidari A. A., Wang M., Zhang X., Chen Y., et al, “Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection”, Knowledge Based Systems, Feb. 2021, vol. 213, pp. 106684.

47. Cai Z., Gu J., Luo J., Zhang Q., Chen H., Pan Z., et al, “Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy”, Expert Systems with Applications, 30 Dec. 2019, vol. 138, pp. 112814.

48. Zhao X., Zhang X., Cai Z., Tian X., Wang X., Huang Y., et al. “Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients”, Computational Biology and Chemistry, Feb. 2019, vol. 78, pp. 481–490. https://doi.org/10.1016/j.compbiolchem.2018.11.017 PMID: 30501982
49. Wang M., Chen H., Li H., Cai Z., Zhao X., Tong C., et al, “Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction”, Engineering Applications of Artificial Intelligence, Aug. 2017, vol. 63, pp. 54–68.

50. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” Int.Conf.Learn.Represent., Sep. 2014, pp. 1150–1210.

51. Everingham M., Eslami S. M. A., Van Gool L., Williams C. K. I., Winn J., and Zisserman A., “The pascal visual object classes challenge: A retrospective,” Int. J. Comput. Vis., Jan. 2015, vol. 111, no. 1, pp. 98–136.

52. A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” IEEE Conference on Computer Vision & Pattern Recognition, 2012, 1, 6, 7.

53. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al, “Caffe: Convolutional architecture for fast feature embedding,” ACM Int.Conf.Multimed, Jun. 2014, pp. 675–678.