Vehicle Detection Method for Remote Sensing Images Based on Feature Anti-Interference and Adaptive Residual Attention

Qingyu Song and Xueyun Chen

Abstract — Vehicle detection in remote sensing images is of great significance to urban traffic intelligence. Though existing vehicle detection methods for remote sensing images such as fully convolutional regression network, spatial density building net, and pretraining and random-initialized fusion network have made many efforts and progress on network structural optimization, their models remain being weak on the feature anti-interference, contextual information utilization, and neglect the loss of feature information during the down-sampling process. In this article, we propose feature anti-interference and adaptive residual attention-Net, a remote sensing image object detection algorithm based on feature anti-interference and adaptive residual attention. First, a feature interference module is constructed, fed with the shallow feature and random noise, and generates interference in the detection process, making the detector improve its anti-interference ability against the disturbance in the adversarial training process. Second, a novel adaptive residual attention module is introduced into the network to extract the adaptively contextual features and enhance the weak features. Finally, a cross level fusion module is designed to enhance the collaboration between multiscale feature layers to reduce the loss of small target feature information. The effectiveness of the method is verified by comparing the method proposed in this article with other mainstream methods on the UCAS-AOD, CARPK, and OVDS datasets.

Index Terms — Adaptive residual attention, cross level fusion, feature anti-interference, remote sensing image.

I. INTRODUCTION

In recent years, vehicle detection has been getting more and more attention for its application potential in traffic intelligence and military defense [1], [2], [3], [4], [5]. In this field, deep learning has become the research hotspot and the mainstream. Most researchers focus on optimizing feature and network structure, including feature richness enhancement, spatial context information extraction, and cross-layer fusion. However, due to the complex background and small targets, it remains a long way to meet the practical needs. More efficient feature extraction and structural optimization methods are expected.

Up to the present, vehicle detection algorithms based on deep learning can be roughly divided into two categories: the first is the anchor-based category, and the second is the shape-regressed category. In the first category, Ren et al. [6] proposed faster-RCNN based on fast-RCNN [7], which used region proposal networks to generate the anchor boxes and adjust the coordinates replacing the other R-CNN [7], [8] algorithms that generated anchor boxes by rules. Nan and Li [9] proposed an oriented vehicle detection framework for aerial images based on improved faster-RCNN, which used the oversampling and stitching data augmentation method to decrease the negative effect of category imbalance. Liu et al. [10] proposed a one-stage detection network single shot multibox detector (SSD). Tang et al. [11] proposed an oriented SSD to generate arbitrarily-oriented detection results, which used default boxes with various scales on each feature map location to produce detection bounding boxes. Anchor-based methods can not realize pixel-wise detection, thus, unable to distinguish dense-align objects under its nonmaximal suppression algorithm.

In the second category, the shape-regression methods detect objects pixel-wisely, without any nonmaximal suppression, and are suitable for dense-align and small object detection. Duan et al. [12] proposed the coarse-grained density map network, where the coarse-grained density map was predicted by the estimation network, and cluster regions were generated based on density maps. The network achieved accuracy improvements. Li et al. [13] proposed a density-map guided object detection network, which generated a density map and learned scale information based on density intensities to form cropping regions, and it improved the detection accuracy in high-resolution aerial images. Tayara et al. [14] proposed the fully convolutional regression network (FCRN) that mapped detected vehicles from the aerial images into the spatial density elliptic kernels and significantly improved the detection accuracy. Chen et al. [15] presented a two-stage spatial density building net (SDBN) to obtain accurate vehicle geometrical parameters based on the spatial density map. Liu et al. [16] proposed a pretraining and random-initialized fusion network (PRFN), which fused the random-initialized model and the pretrained model to improve the richness and robustness of the features of the model. In addition, Makantasis et al. [17], [18] proposed a novel Rank-R FNN, a tensor-based nonlinear classifier. The model can fully exploit the structural information along every data dimension.
The model has fewer trainable parameters and achieves high classification accuracy on small sample data sets. Protopapadakis et al. [19] proposed a stack autoencoder-driven and semisupervised-based deep neural network (DNN). Less than 0.08% of labeled data were used to train the depth model. And semisupervised techniques were used to estimate the soft labels of large amounts of existing unlabeled data. This method significantly reduced the time of manual data annotation.

However, the existing shape-regressed methods have some common shortcomings as follows.

1) Lack of anti-interference ability, vulnerable to the complex background in the detection process.
2) The detection feature of the remote sensing images relies severely on the local context information, but the existing vehicle methods pay less attention to obtaining the spatial context information.
3) The loss of small target feature information caused in the down-sampling process is not fully compensated.

In order to improve the anti-interference ability of the network and reduce the influence of complex backgrounds on the detection process, Ding et al. [20] proposed an anti-interference road detection network, which generated pseudofeatures by the generator and superposed them onto the multilayer features of the encoder. To enhance spatial context feature extraction, Woo et al. [21] proposed a lightweight general module that could sequentially infer attention maps along the channel dimension and spatial dimension. Wang et al. [22] proposed the nonlocal module to get the global context information through the self-attention mechanism, which captures long-distance feature dependencies.

Motivated by the abovementioned research works, in this article, a vehicle detection network based on feature anti-interference and adaptive residual attention (FICLAR-Net) is proposed. Instead of simply stacking the pseudofeatures with the encoder, in the feature interference module (FIM), shallow features and random noise are used as the input of the generator, and the interference features obtained are connected to the trunk network through special modules. We design four connection modules (CM) and add interference at four different locations. In addition, we design a new attention module (ARAM), which enhances weak features through deformable convolution residual blocks, and obtains context information through the pixel attention mechanism. Finally, we use atrous convolution to construct the cross level fusion module (CLFM) and fuse multilayer feature information to increase feature richness. The contributions of this article are summarized as follows.

1) A FIM is constructed, fed with the shallow feature and random noise, and generates interference in the detection process, making the detector improve its anti-interference ability against the disturbance in the adversarial training process.
2) A novel ARAM is introduced into the network for extracting the adaptively contextual features and enhancing the weak features.
3) A CLFM is designed to reduce the loss of small target feature information via the collaboration between multiscale feature layers.

II. RELATED WORKS

A. Generative Adversarial Network

Goodfellow et al. [23] proposed generative adversarial network based on game theory. It used generator and discriminator to carry out generative adversarial training. This model has attracted the attention of researchers since it was put forward, and many improvements have been carried out [24], [25], [26], [27]

\[
\min_G \max_D V(D, G) = E_{x \sim P_{data}(x)} [\log D(x)] + E_{z \sim P_{z}(z)} [\log(1 - D(G(z))].
\]

In formula (1), the input of the generator (denoted as \( G \)) is the random variable (denoted as \( z \)) from the hidden space (denoted as \( p_z \)). The input of the discriminator (denoted as \( D \)) is the real sample (denoted as \( x \)) or the generated sample (denoted as \( x' \)). The training objective is to distinguish the real sample from the generated sample.

B. Deformable Convolution

Unlike traditional convolution, deformable convolution [28] can adaptively learn two-dimensional (2-D) position offset to obtain a kernel with adaptive shape, adapting to geometric deformation of target size and scale in complex scenes. Furthermore, DCNv2 [29] introduced the modulation mechanism into the standard deformable module to enhance the ability to manipulate the support area of the space. The deformable convolution of modulation is expressed as

\[
y(p) = \sum_{k=1}^{K} w_k * x(p + p_k + \Delta p_k) * \Delta m_k
\]

where \( K \) is the number of sampling positions, \( \Delta m_k \) is the learnable modulation scalar, and, \( w_k, p_k, \) and \( \Delta p_k \) represent the weights of the sample positions, the default offsets, and the learnable offsets, respectively.

III. METHODOLOGY

The model proposed in this article is shown in Fig. 1. The model structure includes backbone, FIM, CM, ARAM, CLFM, and decoder with the skip-convolution. The backbone is based on the DenseNet [30]. The clustering algorithm and the loss functions are also presented in this section.

A. Feature Interference Module and Connection Module

As shown in Fig. 1, we construct a FIM to improve the performance of the network. The shallow features of the input image are extracted through the first layer of the backbone, then fused with random noise of the same size and sent to the generator to obtain the interfering features, and then sent to the network to obtain the combined features. A discriminator is used to distinguish between the mixed feature distribution and the actual feature distribution. Both the generator and the discriminator consist of multiple convolutional layers stacked on top of each other. Sending the mixed features to the detector causes some interference to the detection process and facilitates
The model includes backbone (DenseNet), FIM, CM, ARAM, CLFM, and Decoder. In the training process, for the same input, the network is divided into two training steps: no noise added ($a=0$) and noise added ($a=1$). The input image is sent into the backbone and then through the CLFM and the ARAM. The shallow features and random noise ($a=1$ or $a=0$) are stitched together and fed into the Generator, and then the resulting interfering features are fed into the trunk network. The Discriminator is used to discriminate whether the main feature adds interference. Finally, the Decoder outputs the spatial density map and calculates the loss with the label. In addition, we design four CM and add interference at four different locations.

$$F_{\text{noise}} = \text{Gen} (\text{concat}(a \cdot \text{Noise}, F_1))$$

(3)

$$F_{\text{mix}} = \text{CM} (F_{\text{origin}}, aF_{\text{noise}})$$

(4)

$$\begin{align*}
\text{Dis}(F_{\text{mix}}) & \xrightarrow{a=0} \text{real} \\
\text{Dis}(F_{\text{mix}}) & \xrightarrow{a=1} \text{fake}
\end{align*}$$

(5)

$$\begin{align*}
\text{De}(F_{\text{end}}) & \xrightarrow{a=0} \text{GT} \\
\text{De}(F_{\text{end}}) & \xrightarrow{a=1} \text{GT}
\end{align*}$$

(6)

$$\min_{G} \max_{D} V(D, G)$$

(7)

where $F_{\text{noise}}$ is the interference feature generated by the generator, Gen is the generator, Noise is the random noise, $F_1$ is the shallow features, $F_{\text{origin}}$ is the feature without interference, $F_{\text{mix}}$ is the mixed feature after interference, $\text{Dis}$ is the discriminator, $\text{De}$ is the decoder, and $F_{\text{end}}$ is the input of the decoder.

In order to analyze the impact of different interference location methods on the network, we designed four different interference locations, as shown in Fig. 1. Location 1 adds the interferential features after the backbone, location 2 adds the interferential features after the cross level fusion module, location 3 adds the interferential features after the deformable convolution module inside the adaptive residual attention module, and location 4 adds the interferential features after the adaptive residual attention module. In addition, we designed four different connection modules, as shown in Fig. 2. Module (a) is designed in a similar way to Ding [17], module (b) and module (c) automatically adjust the interference ratio through learnable parameters, and module (d) adds interference twice. We analyzed the impact of different interference locations and connection modules on network performance.

### B. Adaptive Residual Attention Module

There are partial occlusions in remote sensing images, and the shape of the occluded vehicles is incomplete. The convolution kernel of deformable convolution can adaptively extract features through offset learning. Therefore, we use deformable convolution residual blocks to improve the detection network’s...
attention to occluded objects. In addition, the feature information of the occluded targets and some inconspicuous targets are weak, and the context information has a better enhancement effect on the insignificant features, so we introduce a new attention mechanism for feature enhancement. As shown in Fig. 3(a), the front end of the module is composed of deformable convolution residual blocks, and the back end is an attention mechanism.

C. Cross Level Fusion Module

Due to the small sizes of the vehicles in the remote sensing images, fewer pixels can be used, and after multiple down-sampling, some feature information is lost. In the network, the low-level layers contain spatial features, and the high-level layers contain semantic features, so we combine spatial information with semantic information to obtain richer high-level feature information. The structure of the cross level fusion module is shown in Fig. 4. First, we adjust the receptive field of the feature layers by using atrous convolution to improve the feature fit between feature layers and then aggregate with high-level layers. Since the output of the first layer contains too much background information, which is not conducive to the detection process, only the outputs of layers 2, 3, and 4 are aggregated with high-level features.

D. Decoder With the Skip Convolution

Skip connections are used in many networks to combine high-resolution shallow features, and up-sampled output deep feature layers to guide local pixel classification with global contextual information. However, due to the large span of the feature layer, the information is discontinuous, resulting in the misclassification of pixels. To this end, we increase the coherence of features by using skip convolutions on skip connections. By combining local information with high-level semantic information, we jointly guide the generation of density maps. The skip convolution used in the decoder is shown in Fig. 5.
target location in the density map, it is a convex distribution. We set the empirical threshold to obtain a binary graph and judge the connectivity of the region through the connectivity algorithm. Then, filter the binary image to get the accurate target area.

**F. Loss Function**

The trunk network uses the BCE loss function for the first 150 epochs (a total of 600 epochs) and then uses the combined loss function of BCE loss and Dice loss. L1 loss is used for the discriminator. The generator loss is included in the trunk network. The loss function formulas are as follows:

\[
\text{Loss}_{\text{BCE}} = - [y \log y' + (1 - y) \log(1 - y')] \quad (8)
\]

\[
\text{Loss}_{\text{Dice}} = 1 - \frac{2 |y \cap y'|}{|y| + |y'|} \quad (9)
\]

\[
\text{Loss}_{\text{trunk}} = \text{Loss}_{\text{BCE}} + \alpha \text{Loss}_{\text{Dice}} \quad (10)
\]

\[
\text{Loss}_{1} = \sum_{i=1}^{n} |y_i - y'_i| \quad (11)
\]

\[
\text{Loss}_{\text{Dis}} = \text{Loss}_{1}(\text{Dis}(F_{\text{mix-noise}}, 0.9)
\]

\[
+ \text{Loss}_{1}(\text{Dis}(F_{\text{mix-noise}}, 0.1) \quad (12)
\]

\[
\text{Loss}_{\text{all}} = \text{Loss}_{\text{trunk}} + \text{Loss}_{\text{Dis}} \quad (13)
\]

where \( y \) is the label value, \( y' \) is the output value, and \( \alpha \) is a constant.

**IV. EXPERIMENTS AND DETAILS**

In this section, we introduce the dataset used for the experiments, density map generation, training details, value evaluation, and experimental results. All experiments are performed on a piece of NVIDIA GeForce RTX 3090 24GB using the Pytorch framework.

**A. Datasets**

The model proposed in this article is validated on three open datasets. The first is the UCAS-AOD [31] dataset. The UCAS-AOD dataset contains 510 remote sensing images of size \( 1280 \times 659 \) pixel, with a total of 7114 vehicles. The second is the CARPK [32] dataset, which contains 1448 aerial images with a resolution of \( 1280 \times 972 \) Image, satellite imagery of the San Francisco area from Google-Earth. The third is the OVDS [33] dataset, which contains 111 images with a resolution of 1368 \( \times \) 972 Image, satellite imagery of the San Francisco area from Google-Earth. There are many disturbances in these datasets, such as occlusion and street scenes similar to vehicles, which increase the difficulty of vehicle detection. Some dataset samples are shown in Fig. 6.

**B. Density Map**

We use the 2-D Gaussian ellipse kernel function to generate density maps. The pixel value of each point represents the probability of the vehicle targets. The main formulas are as follows:

\[
f(x, y) = Ae^{-(a(x-x_0)^2+2b(x-x_0)(y-y_0)+c(y-y_0)^2)} \quad (14)
\]

\[
a = \frac{\cos^2\theta}{2\sigma_x^2} + \frac{\sin^2\theta}{2\sigma_y^2} \quad (15)
\]

\[
b = \frac{-\sin^2\theta}{4\sigma_x^2} + \frac{\sin^2\theta}{4\sigma_y^2} \quad (16)
\]

\[
c = \frac{\sin^2\theta}{2\sigma_x^2} + \frac{\cos^2\theta}{2\sigma_y^2} \quad (17)
\]

where \( A \) is set to 255, \( \sigma_x \) and \( \sigma_y \) are the length and width of the vehicle, and \( \theta \) is the direction of the target vehicle. Some dataset samples are shown in Fig. 7.

**C. Training Details**

The network uses the Kaiming method to initialize weights, and the generative adversarial network is initialized using Pytorch’s default settings. In the training phase, we use the Adam optimizer, and the initial learning rate is 1e-4. During the training process, the first 150 epochs do not add noise at all (\( a = 0 \)), and the learning rate is unchanged. After 150 epochs add noise to the network (\( a = 1 \) or \( a = 0 \)), the learning rate decays by 10 per 150 epochs (a total of 600 epochs). The input to the network is a three-channel RGB image with a size of \( 256 \times 256 \). Preprocessing operations such as scaling and random rotation are performed before being fed into the network.
Fig. 7. First column in the figure is the original inputs, and the second column is the density maps.

**D. Evaluation Metrics**

In this article, we use precision, recall, and f1-score to evaluate the experimental results, and the formulas are as follows:

\[
\text{recall} = \frac{TP}{TP + FN} \tag{18}
\]

\[
\text{precision} = \frac{TP}{TP + FP} \tag{19}
\]

\[
F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{20}
\]

where true positive (TP) represents the number of positive samples that are correctly judged as positive samples, false positive (FP) represents the number of negative samples that are incorrectly judged as positive samples, and false negative (FN) represents the number of positive samples that are incorrectly judged as negative samples number.

**E. Experimental Results**

In order to verify the effectiveness of our proposed modules, we conducted ablation experiments. The experimental results are shown in Table I. The F1 score of the network increased by 0.43% after adding the CLFM, the F1 score of the network increased by 0.51% after adding the ARAM, and the F1 score of the network increased by 0.82% after adding the FIM. In addition, we list the number of parameters and computations for each network structure in Table I.

We analyzed the impact of different interference locations and connection modules on the network performance. We have found that the optimal results are obtained by adding interference at mode 2 using the connection module (b), and the experimental results are shown in Tables II and III.

In addition, we compared the adaptive residual attention module with other common attention modules, and the experimental results are shown in Table IV. The adaptive residual attention module achieved the best results.

To demonstrate the performance of the network, we compare our model with some mainstream methods on three open datasets. The experimental results are shown in Tables V, VI, and VII. On the CARPK Dataset, the Precision and F1 score improved by 1.56% and 0.81%, respectively. On the OVDS dataset, the Precision and F1 score increased by 2.49%.
Fig. 8. Detection results of different parameters of the network. (B: DenseNet; S: Skip Convolution; D: Decoder; F: Cross Level Fusion Module; A: Adaptive Residual Attention Module; G: Feature Interference Module). The green bounding boxes represent the detection results, the yellow dots indicate missed inspections, and the red dots indicate false detections.

| Table V | Results on the CARPK Dataset |
|---------|-----------------------------|
| Method  |   Recall  |  Precision  |  F1 score  |
| Faster-RCNN[9] | 81.27% | 84.85% | 83.02% |
| SSD[11] | 87.52% | 89.09% | 88.30% |
| YOLO[34] | 74.50% | 83.33% | 78.67% |
| FCN8s[35] | 86.03% | 89.02% | 87.50% |
| SegNet[36] | 86.89% | 87.77% | 87.33% |
| U-Net[37] | 89.61% | 91.43% | 90.51% |
| FCRN | 88.45% | 92.50% | 90.43% |
| SDBN | 89.08% | 93.17% | 91.08% |
| PRFN | **92.03%** | 92.63% | 91.31% |
| Proposed | 90.14% | **94.19%** | **92.12%** |

| Table VI | Results on the OVDS Dataset |
|---------|-----------------------------|
| Method  |   Recall  |  Precision  |  F1 score  |
| SSD | 89.50% | 89.17% | 89.34% |
| FCN8s | 85.33% | 89.36% | 87.30% |
| SegNet | 84.01% | 85.14% | 84.57% |
| U-Net | 89.66% | 91.58% | 90.61% |
| FCRN | 90.41% | 90.67% | 90.53% |
| SDBN | 89.33% | 94.53% | 91.86% |
| PRFN | **91.71%** | 92.67% | 92.19% |
| Proposed | 89.98% | **95.16%** | **92.50%** |

and 0.31%, respectively, and the Precision and F1 scores on the UCAS-AOD Vehicle Dataset improved by 2.30% and 0.47%, respectively. Analysis shows that our network achieves significant improvements on the vehicle dataset, indicating that it is robust to small objects such as vehicles.

| Table VII | Results on the UCAS-AOD Vehicle Dataset |
|---------|-----------------------------|
| Method  |   Recall  |  Precision  |  F1 score  |
| FCN8s | 85.79% | 87.51% | 86.64% |
| SegNet | 80.17% | 88.43% | 84.10% |
| U-Net | 88.71% | 93.33% | 90.96% |
| FCRN | 89.09% | 90.50% | 89.79% |
| SDBN | 88.38% | 92.45% | 90.37% |
| PRFN | **90.57%** | 92.63% | 91.59% |
| Proposed | 89.36% | **94.93%** | **92.06%** |

| Table VIII | Results on the UCAS-AOD Vehicle Dataset |
|---------|-----------------------------|
| Train Set Size | SDBN(F1) | PRFN(F1) | Proposed(F1) |
| 25% | 83.61% | 82.58% | 83.42% |
| 50% | 85.92% | 85.86% | **86.32%** |
| 75% | 88.76% | 89.34% | **90.97%** |
| 100% | 90.37% | 91.59% | **92.06%** |

To study the robustness of the method in this article, we refer to the experimental method of Makantasis et al. [17], [18] to observe the variation of the accuracy of the network by changing the sample number of the training set and comparing it with other methods. We set the proportions of the training set to 25%, 50%, 75%, and 100%. And the test set remains the same. The experimental results are shown in Table VIII. When the proportion of the training set is more than 50%, our network can still achieve better accuracy than other methods. However, when
the proportion of the training set is less than 25%, our network is less accurate than other networks.

In addition, we compare with some semisupervised networks. Set the training set label ratio to 50%, and the related results are shown in Table IX. Due to less labeling information, the accuracy of our network is slightly lower than that of DNN-WeiAve, but higher than that of BAS4Net.

### V. Conclusion

In this article, we propose FICLAR-Net, a vehicle detection network in remote sensing images based on feature anti-interference and adaptive residual attention. We use the FIM to obtain interferential features and send them into the main network to enhance the anti-interference ability. We construct an ARAM based on deformable convolution, which utilizes contextual information to enhance weak features and implement adaptive feature extraction. In view of the lack of available features for vehicle targets, we propose a CLFM to obtain richer features and improve the detection performance of the network. We conduct experiments on three open datasets to verify the effectiveness of our method. In addition, this model also has shortcomings. This is a supervised network requiring manual annotation and a certain amount of memory. In the future, we will try to combine the method with the self-supervised methods in future research work to improve the learning ability of the network and reduce the use of annotated data.
REFERENCES

[1] Q. Tan, J. Ling, J. Hu, X. Qin, and J. Hu, “Vehicle detection in high resolution satellite remote sensing images based on deep learning,” IEEE Access, vol. 8, pp. 153394–153402, 2020, doi: 10.1109/ACCESS.2020.3017894.

[2] A. K. Dutta, M. Elhoseny, V. Daihya, and K. Shankar, “An efficient hierarchical clustering protocol for multithop internet of vehicles communication,” Trans. Emerg. Telecommun. Technol., vol. 31, no. 5, 2020, Art. no. e3690.

[3] R. H. Ahmad Fuad and M. Drieberg, “Remote vehicle tracking system using GSM modem and google map,” in Proc. IEEE Conf. Sustain. Utilization Develop. Eng. Technol., 2013, pp. 15–19, doi:10.1109/CUSD.2013.6670977.

[4] L. De Paula Veronica et al., “Re-emission and satellite aerial maps applied to vehicle localization on urban environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2015, pp. 4256–4259, doi:10.1109/IROS.2015.7353984.

[5] T. Senlet and A. Elgammal, “A framework for global vehicle localization using stereo images and satellite and road maps,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2011, pp. 2034–2041, doi:10.1109/ICCVW.2011.6130498.

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards realtime object detection,” in Proc. IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[7] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1440–1448, doi:10.1109/ICCV.2015.169.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2014, pp. 580–587, doi:10.1109/CVPR.2014.81.

[9] M. Nan and Y. Li, “Oriented vehicle detection in high-resolution remote sensing images based on feature amplification and category balance by oversampling data augmentation,” in Proc. Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci., 2020, pp. 14–20.

[10] W. Liu et al., “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 21–37.

[11] T. Tang, S. Zhou, Z. Deng, L. Lei, and H. Zou, “Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks,” Remote Sens., vol. 9, no. 11, 2017, Art. no. 1170, doi:10.3390/rs9111170.

[12] C. Duan, Z. Wei, C. Zhang, S. Qu, and H. Wang, “Coarse-grained density map guided object detection in aerial images,” in Proc. IEEE/ACM Int. Conf. Comput. Vis. Workshops, 2021, pp. 2789–2798, doi:10.1109/ICCVW54120.2021.00013.

[13] C. Li, T. Yang, S. Zhu, C. Chen, and S. Guan, “Density map guided object detection in aerial images,” in Proc. IEEE/ACM Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 737–746, doi:10.1109/CVPRW50498.2020.00103.

[14] H. Tayara, K. G. Soo, and K. T. Chong, “Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network,” IEEE Access, vol. 6, pp. 2220–2230, 2018, doi:10.1109/ACCESS.2017.2782260.

[15] X. Chen, J. Lin, S. Xiang, and C.-H. Pan, “Detecting maneuvering target accurately based on a two-phase approach from remote sensing imagery,” IEEE Geosci. Remote Sens. Lett., vol. 17, no. 5, pp. 849–853, May 2020, doi:10.1109/LGRS.2019.2952530.

[16] H. Liu, Q. Ding, Z. Hu, and X. Chen, “Remote sensing image vehicle detection based on pre-training and random-initialized fusion network,” IEEE Geosci. Remote Sens. Lett., vol. 19, 2022, Art. no. 508605, doi:10.1109/LGRS.2021.3109657.

[17] K. Makantasis, A. Georgogiannis, A. Voulodimos, I. Georgoulas, A. Doulamis, and N. Doulamis, “Rank-R FNN: A tensor-based learning model for high-order data classification,” IEEE Access, vol. 9, pp. 58609–58620, 2021, doi:10.1109/ACCESS.2021.3072973.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi:10.1109/TPAMI.2016.26.

[19] X. Sun, A. Shi, H. Huang, and H. Mayer, “BAS4+[4]N: Boundary-aware semi-supervised semantic segmentation network for very high resolution remote sensing images,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 13, pp. 5398–5413, 2020, doi:10.1109/JSTARS.2020.3021098.

Xueyuan Chen received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014. He is currently an Associate Professor and a Ph.D. Supervisor with the School of Electrical Engineering, Guangxi University, Nanning, China. His research interests include object detection and recognition in remote sensing image, face detection, road detection, and automatic driving.