Hindawi
Mobile Information Systems
Volume 2022, Article ID 7319529, 11 pages
https://doi.org/10.1155/2022/7319529

Research Article

UAV Image Small Object Detection Based on Composite Backbone Network

Wuji Liu, Jun Qiang, Xixi Li, Ping Guan, and Yunlong Du

School of Computer and Information, Anhui Polytechnic University, Wuhu, China

Correspondence should be addressed to Jun Qiang; chiang_j@ahpu.edu.cn

Received 8 December 2021; Accepted 28 March 2022; Published 13 April 2022

Academic Editor: Yvette Gonzales

Small objects in traffic scenes are difficult to detect. To improve the accuracy of small object detection using images taken by unmanned aerial vehicles (UAV), this study proposes a feature-enhancement detection algorithm based on a single shot multibox detector (SSD), named composite backbone single shot multibox detector (CBSSD), which uses a composite connection backbone to enhance feature representation. First, to enhance the detection effect of small objects, the lead backbone network, VGG16, is kept constant, and ResNet50 is added as an assistant backbone network, and the residual structure in ResNet50 is used to obtain lower feature information. The obtained lower feature information is then fused to the lead network through feature fusion, allowing the lead network to retain rich lower feature information. Finally, the lower feature information in the prediction layer increases. The experimental results show that CBSSD has a significantly higher recognition rate and a lower false detection rate than conventional algorithms, and it still maintains a good detection effect under low illumination. This is of great significance to small object detection using images taken by UAVs in traffic scenes. Furthermore, a method to improve the SSD algorithm is proposed.

1. Introduction

Recently, with the rapid development of artificial intelligence, unmanned aerial vehicle (UAV) detection technology has been widely applied to real traffic scenes [1, 2]. Vehicle and pedestrian detection, as an important part of UAV detection technology, is of research significance [3, 4]. Object detection methods can be classified as conventional machine learning and deep learning methods. Conventional machine learning methods first preprocess the image, and then the candidate area is determined using the sliding window technique. Subsequently, features of the candidate regions are extracted, and a classifier is used to determine the classification information of an object to realize object detection. Common machine learning methods include the scale-invariant feature transform [5], histogram of oriented gradient [6], Harr [7], and speeded up robust feature [8]. However, because conventional machine learning methods are based on the manual design of features, and the process of feature extraction is too complex, these methods often face problems, such as poor generalization ability, slow detection speed, low detection accuracy, and difficulty in adapting to detection tasks in different scenarios.

To address the abovementioned problems, the general object detection method based on convolutional neural networks (CNNs) has gradually gained research attention. At present, object detection methods based on deep convolutional networks are classified as two- and single-stage detection methods. Among the two-stage detection methods, the Faster R-CNN proposed by Ren et al. [9] has the best performance. This network introduces a regional proposal network (RPN) that can simultaneously predict the object boundary and object score of each position. After end-to-end training, high-quality regional suggestions are generated to improve the detection accuracy of the network. Given the efficiency issue, the single-stage method was proposed, with representative methods being you only look once (YOLO) [10] and the single shot multibox detector (SSD) [11–13]. YOLO uses the feature graph at the top of the CNN to predict category confidence and border bias, and it processes the detection problem as a regression problem, which provides the advantage of fast detection. However, YOLO uses a fully connected

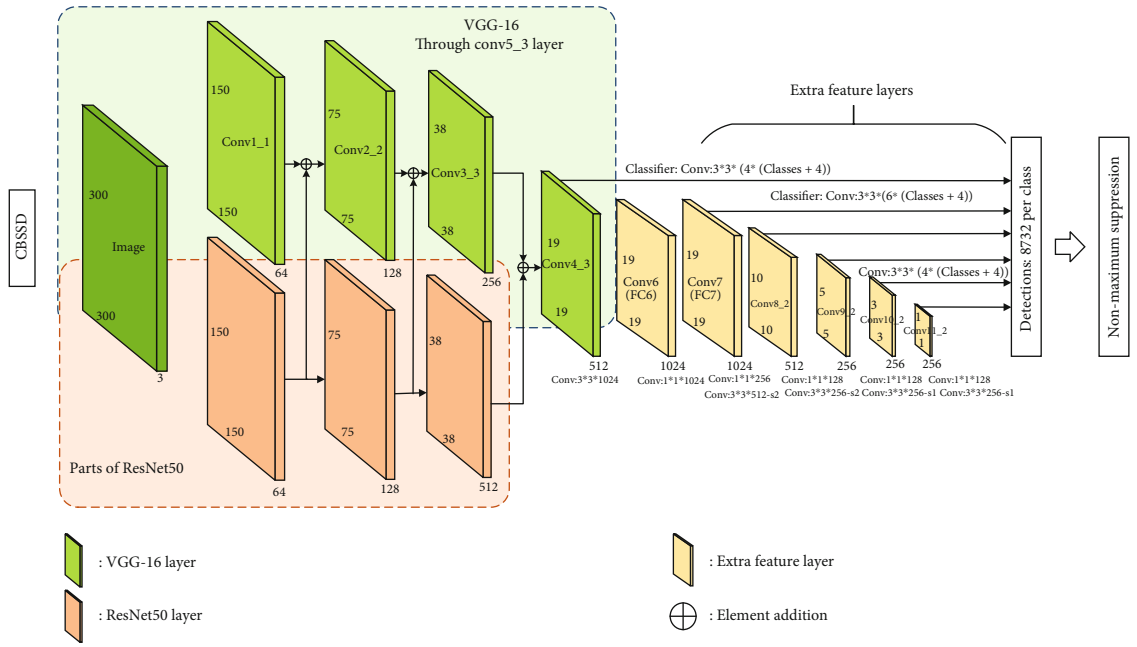


Figure 1: Block diagram of CBSSD algorithm.

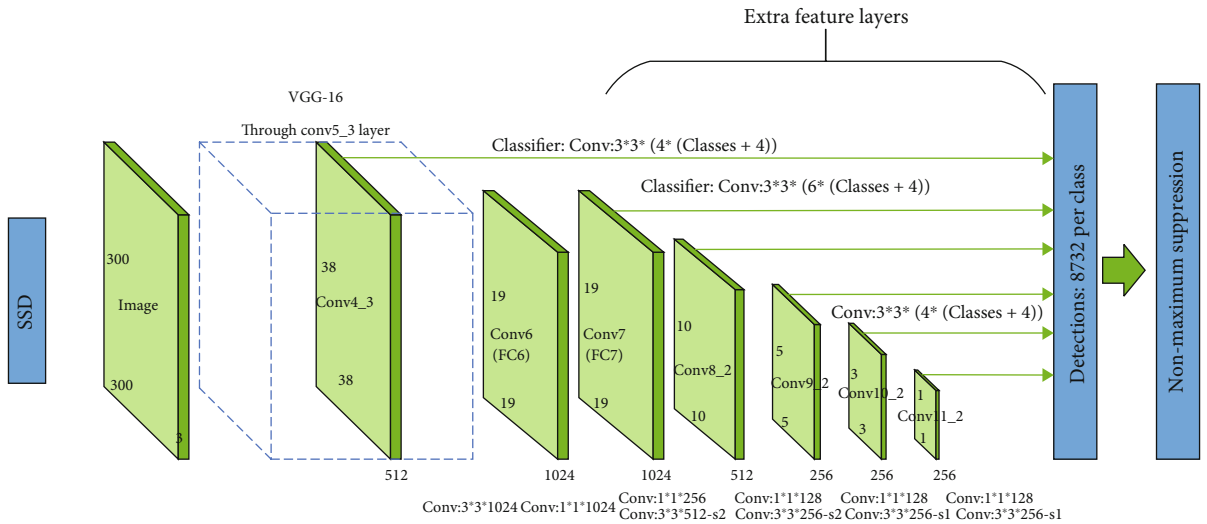


Figure 2: Block diagram of SSD algorithm.

network, which leads to the loss of spatial information, positioning errors, and missed object detection, especially a poor detection effect on small objects, affecting the final detection accuracy. SSD borrows the Anchor idea in Faster R-CNN and uses multiple feature maps of different scales for detection. SSD can detect objects of various sizes because the receptive fields of each feature map are different. However, the semantic information of the SSD shallow feature map is poor; therefore, it is not suitable for small object detection. To further improve the detection effect of SSD on small objects, Li et al. [14] proposed a feature fusion SSD (FSSD) model, which is an enhanced SSD model with a novel lightweight feature fusion module. This can significantly improve SSD performance. In the feature fusion module, the features of different layers are connected at different scales. Some subsampling blocks generate new feature pyramids, which are sent to multi-bounding box detectors to predict the final detection results. Recently, Liu et al. [15] proposed a new target detection method called the composite backbone network architecture (CBNet). This approach improves the performance of object detectors by combining multiple identical backbones, and CBNet can be easily integrated into most of the advanced detectors, thus significantly improving their performances.

In summary, although object detection technology has been well developed, problems still arise in the detection of small objects. To solve this problem, we propose a composite backbone SSD (CBSSD) object detection method. Based on the CBNet network, we introduce the ResNet50 [16–18] network as an auxiliary backbone network based on the SSD

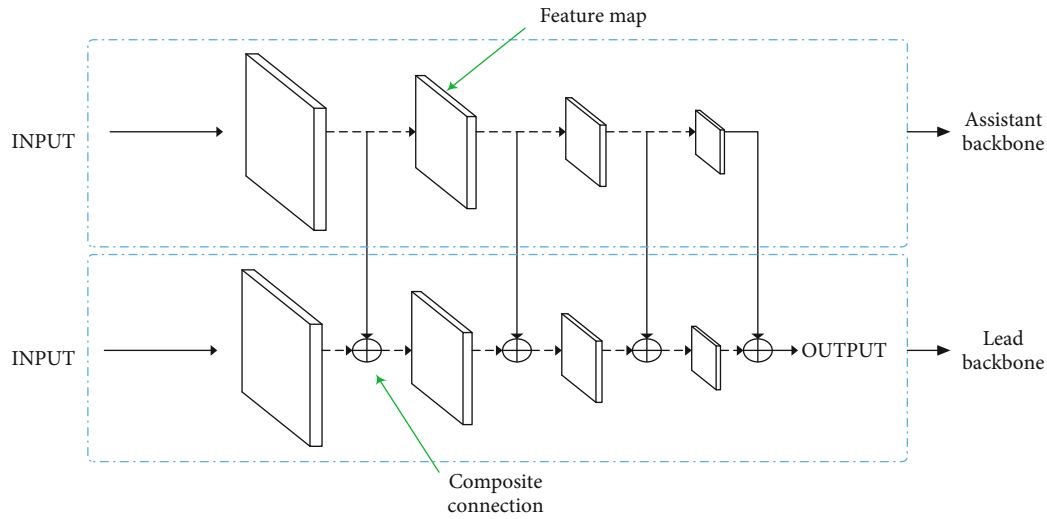


Figure 3: Composite connection backbone: ⊕ is the addition of the elements. Only three layers of composite connections are used in practice.

Table 1: mAPs of several algorithms on Pascal VOC 2012 datasets.

Method	SSD300	FSSD	YOLO3	CBSSD
mAP	77.21	78.82	69.33	*82.77*

network. The residual structure was used to improve the feature extraction ability of the network, retain richer underlying information, and merge deep and shallow features to improve the detection accuracy of the network.

2. Methods

Based on the CBNet network, the CBSSD method consists of a lead backbone network SSD and an assistant backbone network ResNet50, as shown in Figure 1.

2.1. Lead Network

2.1.1. Network Structure. In this study, the SSD model with VGG16 [17, 19, 20] as the main network was selected. VGG16 is a classical network with a network depth of 16. It uses 3×3 convolution kernels of a single size. The SSD method is based on the feedforward convolution network, which generates a set of priori-bounding boxes of fixed sizes and scores in the priori-bounding boxes of object class instances and then generates the final detection result through nonmaximal suppression (NMS) [21, 22]. The first few network layers are based on the standard architecture of high-quality image classification, which is called the basic network. Feature extraction layers, conv8_2, conv9_2, conv10_2, and conv11_2, were added to the basic network. SSD differs from YOLO in that SSD performs predictions on the previously selected five feature maps in addition to object detection on the final feature map.

Figure 2 shows the schematic of the SSD network prediction. Note that the detection process is not only conducted on the added feature graph but also on the basic network feature graphs conv4_3 and conv7 to ensure that the network has a good detection effect on small objects.

2.1.2. Priori-Bounding Box. SSD designs a priori-bounding box of different quantities, scales, and width-to-height ratios for each feature graph. These priori-bounding boxes are composed of a series of object-bounding boxes of fixed quantity and size generated by certain rules. The specific size of the priori-bounding box is determined by the scale and width-to-height ratio, and each layer of the feature map corresponds to a scale, which is generated as

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), k \in [1, m], \tag{1}$$

where s_k represents the scale of the priori-bounding box in the k^{th} feature graph, $s_{\min}$ is 0.2, $s_{\max}$ is 0.9, m represents the number of feature graphs used for detection, and the value of m is 6 in SSD. Each grid on each layer of the feature map must set different numbers and sizes of priori-bounding boxes. In particular, each grid of conv4_3, conv10_2, and conv11_2 generates four priori-bounding boxes with a width-to-height ratio a_{r1} of {1,2,1/2}. conv7, conv8_ 2, conv9_2 each grid on conv7, conv8_2, and conv9_2 feature maps produce six priori-bounding boxes with a width-to-height ratio a_{r2} of {1,2,1/2,3,1/3}. After determining the scale and width-to-height ratio, the size of the priori-bounding box can be obtained as follows:

$$\begin{aligned} w_k^a &= s_k\sqrt{a_r}, \\ h_k^a &= \frac{s_k}{\sqrt{a_r}}, \end{aligned} \tag{2}$$

where w_k^a and h_k^a are the width and height of the priori-bounding box, respectively, and a_r is a_{r1} or a_{r2}. For the priori-bounding box with a width-to-height ratio of 1,

Table 2: mAPs of different networks on the Visdrone2019 datasets.

Method	mAP	Car	Van	Bus	Truck	Motor	Tricycle	People	Pedestrian	Bicycle	Awning-tricycle	Ignored regions	Others
SSD300	11.47	0.43	0.21	0.26	0.21	0.08	0.05	0.04	0.07	0.00	0.02	*0.00*	0.01
FSSD	11.65	0.41	0.22	0.25	0.20	0.08	0.08	0.05	0.06	0.01	0.04	*0.00*	0.00
YOLO3	14.21	0.46	*0.26*	0.30	*0.26*	0.11	0.09	0.06	0.09	0.02	0.04	*0.00*	*0.02*
CBSSD	*19.00*	*0.61*	*0.26*	*0.35*	*0.26*	*0.21*	*0.13*	*0.15*	*0.21*	*0.04*	*0.05*	*0.00*	0.00

another scale s_k' is added, which can be calculated as

$$s_k' = \sqrt{s_k s_{k+1}}. \tag{3}$$

In SSD, the number of priori-bounding boxes in the first detection layer is $38 \times 38 \times 4 = 5776$, $19 \times 19 \times 6 = 2166$, $10 \times 10 \times 6 = 600$, $5 \times 5 \times 6 = 150$, $3 \times 3 \times 4 = 36$, and $1 \times 1 \times 4 = 4$. In total, the network outputs $5776 + 2166 + 600 + 150 + 36 + 4 = 8732$ priori-bounding boxes.

2.2. Composite Backbone Network. The objects in UAV images are mostly small and are subject to severe fuzzy and texture distortion problems and obscure features. Thus, it is difficult for some networks to extract key feature information and influence the recognition ability of classifiers. Therefore, based on CBNet [23], a composite backbone network was proposed, which combines two public backbone networks. Moreover, ResNet50, which can better maintain the details of the lower layer, was selected as the assistant backbone network. By maintaining the lead backbone network, the lower features extracted by ResNet50 are fused layer-by-layer into the VGG16 lead backbone network. The feature layer obtained after fusion is replaced by the original feature layer of the lead backbone network as a new feature layer for the next convolution step (Figure 3).

In the assistant backbone network, the result of each phase can be considered as a higher-level feature. The output of each feature level is part of the lead backbone input and flows to the parallel phase of the subsequent backbone. In this manner, multiple higher and lower features are fused to produce richer feature representations. This process can be expressed as follows:

$$F_{\text{out}} = F_l \bigoplus F_a, \tag{4}$$

$$F_{\text{OUT}} = \varepsilon(F_{\text{out}}), \tag{5}$$

where $\bigoplus$ represents element addition, F_l represents the output features of the lead backbone at the current stage, F_a represents the output features of the assistant backbone, F_{out} represents the display feature fusion results, and F_{OUT} is the input value of the next layer of the lead backbone. The process from $F_{\text{out}} - F_{\text{OUT}}$ is tuned via channels. As shown in Equation (5), ε acts as a 1×1 convolution operation. In theory, this composite connection method can be used at the trunk layer, and our experiment used the most basic and useful composite connection method. This shows that the proposed composite connection method is not limited by the feature size. To simplify the operations, 150×150, 75×75 and 38×38 feature layers were selected on the lead backbone corresponding to the output of the three-layer ResNet50.

2.3. Loss Function. The positive and negative sample generation rule of the SSD involves the calculation of the intersection over union (IOU) of each real-bounding box and all priori-bounding boxes and the matching of the priori-bounding box with the largest IOU value with the real-bounding box. For the other unmatched priori-bounding boxes, when their IOU exceeds the threshold of 0.5, the priori-bounding boxes are matched with the real-bounding boxes. If the IOU of a certain priori-bounding box and those of all real-bounding boxes exceed 0.5, the real-bounding box with the largest IOU is matched with the priori-bounding box. The priori-bounding boxes matched with the real-bounding boxes were set as positive samples, and the unmatched boxes were set as negative samples. As the network performs forward calculation, 8732 priori-bounding boxes are generated, most of which are negative samples. This results in an imbalance between the positive and negative samples. As a result, when calculating loss, negative samples occupy a large proportion, making it difficult for the model to converge. Therefore, after matching, a difficult sample-mining strategy is used to control the ratio of positive to negative samples at 1 : 3 and input the samples into the network for training.

The loss function selected in this study was the same as that used in the conventional SSD network, which is the weighted sum of positioning loss (smooth L1 [24–26]) and confidence loss (Softmax [27–29]), as expressed by

$$L(x, c, l, g) = \frac{1}{N}\left(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)\right), \tag{6}$$

where x is the matching result of the prediction-bounding box and the real-bounding box of different categories, c is the category confidence information of the prediction-bounding box, l is the location information of the prediction-bounding box, and g is the location of the real enclosure. N represents the number of matched priori-bounding boxes. When $N = 0$, the total loss is 0. α is the weight coefficient, L_{loc} (x, l, g) is the position loss, and L_{conf} (x, c) is the classified loss. The position loss is a smooth L1 loss between the prediction-bounding box and the real-bounding box, as expressed by

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} smooth_{L1}\left(l_i^m - \widehat{g}_j^m\right), \tag{7}$$

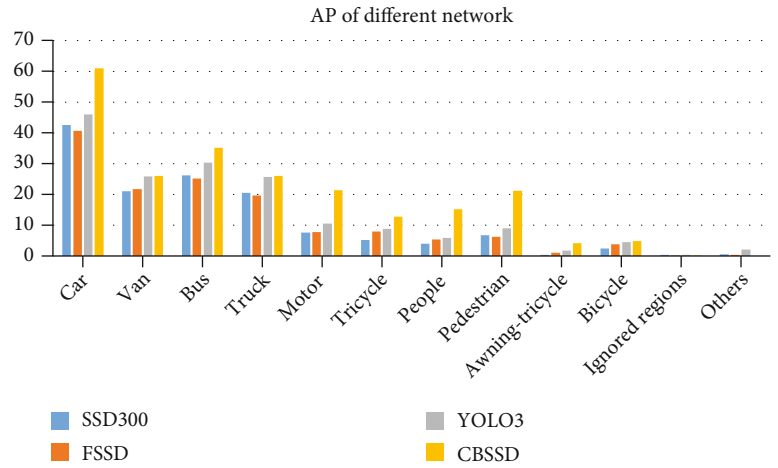


Figure 4: Average precision (AP) of different algorithms on Visdrone2019 datasets.

where x_{ij}^{k} represents whether the ith prediction-bounding box and the jth real-bounding box match in category k. If they match, the value is 1; if they do not match, the value is 0. Similar to Faster R-CNN, SSD performs regression on the central coordinate (cx, cy), width w, and offset of height h of the priori-bounding box. The calculation method is expressed by the following:

$$\begin{aligned}\hat{g}_j^{cx} &= \frac{\left(g_j^{cx} - d_i^{cx}\right)}{d_i^{w}},\\ \hat{g}_j^{cy} &= \frac{\left(g_j^{cy} - d_i^{cy}\right)}{d_i^{h}},\\ \hat{g}_j^{w} &= \log\left(\frac{g_j^{w}}{d_i^{w}}\right),\\ \hat{g}_j^{h} &= \log\left(\frac{g_j^{h}}{d_i^{h}}\right).\end{aligned} \tag{8}$$

The classic Softmax loss was used for loss classification, as expressed by

$$L_{\text{conf}}(x, c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} \log\left(\hat{c}_i^{p}\right) - \sum_{i\in Neg} \log\left(\hat{c}_i^{0}\right) \text{where} \hat{c}_i^{p} = \frac{\exp\left(c_i^{p}\right)}{\sum_p \exp\left(c_i^{p}\right)}, \tag{9}$$

where x_{ij}^{p} represents whether the ith prediction-bounding box and the jth real-bounding box match in category P. If they match, the value is 1; if they do not match, the value is 0. In $\hat{c}_i^{p} = \exp\left(c_i^{p}\right)/\sum_p \exp\left(c_i^{p}\right)$, c_i^{p} represents the prediction probability of the ith prediction-bounding box in category P, and $\hat{c}_i^{0}$ represents the probability that there is no object in the prediction-bounding box.

3. Experiment

3.1. Implementation Details. The proposed framework uses a composite connection of VGG16 and ResNet50 as the backbone. In the training phase, the learning rate of the first 50 epochs was set as $5 \times 10e^{-4}$, and the learning rate was automatically reduced by 50% when the loss function did not decrease by more than three times. The initial learning rate of the training for more than 50 epochs was set as $10e^{-4}$, and the learning rate was automatically reduced by 50% when the loss function did not decrease by more than three times. The training was completed when the loss function did not decrease after three attempts at lowering the learning rate. The experimental environment used in this study was as follows: CPU was Intel I5-9400F; the main frequency was 2.90 GHz (six cores); 16 GB memory; GPU was RTX2060Super; the operating system was 64-bit Windows; and the machine learning framework was Tensorflow2.3.

3.2. The Datasets. Two datasets were used in this study: Pascal VOC 2012 datasets [30] for testing the feasibility of the network and Visdrone2019 UAV aerial photography datasets [31] for training.

3.2.1. Pascal VOC2012 Datasets. As one of the benchmark datasets, Pascal VOC2012 has frequently been used in object detection, image segmentation experiments, and model effect evaluations. The datasets consist of four major categories and 20 subcategories, with 17125 images, including images and test images.

3.2.2. Visdrone2019 Aerial Datasets. The Visdrone2019 aerial datasets are low-altitude aerial datasets, mostly used for small object detection. There are 13 types of objects in the datasets and 7,634 images in the datasets. Most of the images in the datasets are traffic maps, which contain dense small objects.

3.3. Performance Inspection. In this study, the mean average precision (mAP) was used to evaluate the quality of the

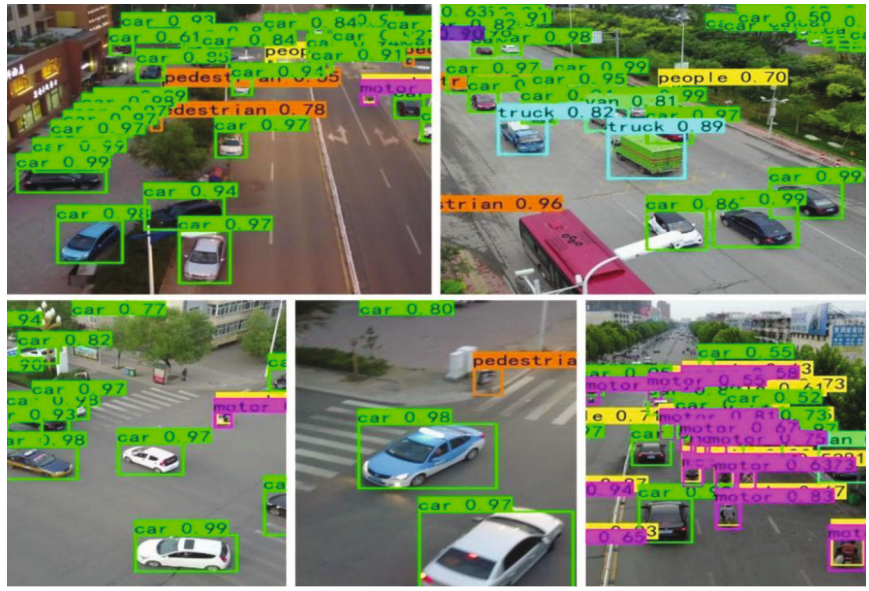


Figure 5: Results of CBSSD on Visdrone2019 datasets.

model. A larger mAP indicates a higher detection performance.

The calculation methods for mAP are expressed by the following:

$$\begin{aligned} Precision &= \frac{TP}{TP + FP}, \\ Recall &= \frac{TP}{TP + FN}, \\ AP &= \frac{\sum Precision}{AllImage}, \\ mAP &= \frac{AP}{Class\_num}, \end{aligned} \tag{10}$$

where TP represents the number of accurately predicted target boxes, FP represents the number of target boxes that failed to predict, FN represents the number of missed ground truths, and AP represents the average precision.

3.4. Performance Test. To test the detection effect of the proposed network, a performance test was conducted using the Pascal VOC 2012 datasets, and the performance was compared with conventional object detection algorithms. The experimental results revealed that the proposed network outperformed several conventional object detection algorithms in terms of mAP. As shown in Table 1, the proposed algorithm demonstrated the highest mAP, thus confirming its superior feasibility.

3.5. Training. To verify the detection effect of the proposed network on small objects, training was conducted using the Visdrone2019 UAV aerial photography datasets, and the performance of the proposed CBSSD algorithm was compared with the conventional object detection algorithm. Experimental data show that the proposed algorithm exhibited a significant improvement over the original network in terms of mAP. As shown in Table 2, the detection accuracy was improved by 7.5% compared with the original algorithm, and the improvement rate was as high as 65%. The improvement was therefore confirmed.

As shown in Figure 4, CBSSD has advantages in detection accuracy for each category of objects, especially for small objects.

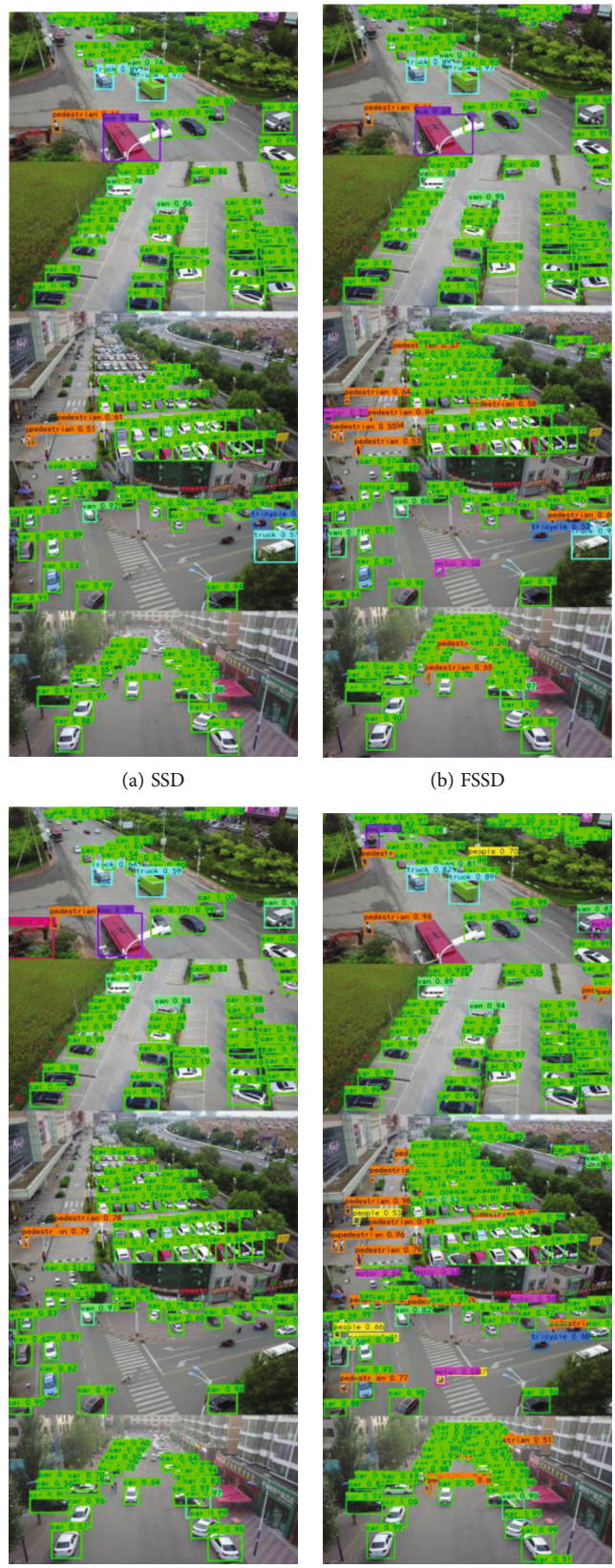
Figure 5 shows the detection results of the CBSSD on the Visdrone2019 datasets. As shown in the figure, CBSSD can maintain high performance despite dense and blurred images and uneven lighting.

Figure 6 shows a comparison of the detection results between the CBSSD algorithm and the classical detection algorithm. The figure shows that the CBSSD algorithm has a better detection effect than several classical object detection algorithms.

The CBSSD algorithm has a significant effect on dense small object detection, as shown in Figure 7. Unmanned aerial images are an important feature of many small objects, and the general algorithm for this type of object detection exhibits a lower performance, because the characteristics of the figure for this type of object in information loss are serious. CBSSD maintains the characteristic diagram with more low-level detail information; therefore, for this type of object, the detection effect is better.

The CBSSD algorithm still has good detection effects for images with weak light intensities and uneven lighting, as shown in Figure 8. CBSSD is also excellent in low-light environments, where the object texture is distorted, which makes detection more difficult.

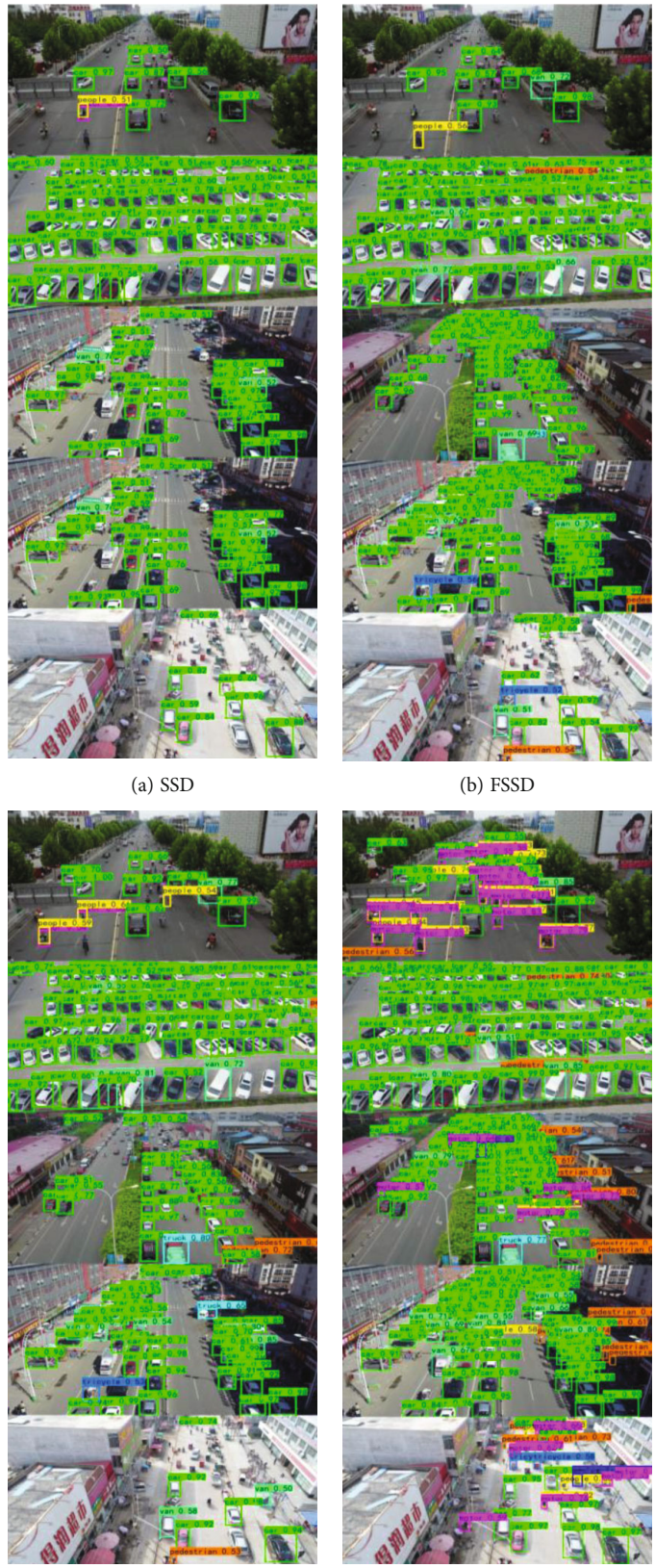
In summary, the experiments showed that the detection accuracy of the proposed CBSSD algorithm significantly improved. Object detection and recognition were significantly increased, recognition accuracy significantly improved, and error detection reduced. For dense small



(a) SSD (b) FSSD

(c) YOLO3 (d) CBSSD

Figure 6: Comparison of confidence between CBSSD and classical object detection algorithms.



(a) SSD (b) FSSD

(c) YOLO3 (d) CBSSD

Figure 7: Comparison of dense object detection effect between CBSSD and classical object detection algorithm.

(a) SSD (b) FSSD

Figure 8: Continued.

(c) YOLO3 (d) CBSSD

Figure 8: Comparison of low-illuminance detection effect between CBSSD and classical object detection algorithm.

objects, the detection effect was significantly enhanced. In particular, in the case of uneven lighting, fuzzy still maintained a good detection effect.

4. Conclusion

This study analyzed the problems associated with small object detection from UAV aerial images. By combining the existing feature extraction trunk in the form of a composite connection, a trunk with stronger feature expression ability is proposed, which solves the problem of poor monitoring when UAV aerial images were captured in dense, fuzzy, and uneven light. The experimental results showed that, compared with other algorithms, the proposed CBSSD algorithm significantly improved the detection effect of small objects in UAV aerial images. Hence, UAV aerial image detection technology can be better applied to traffic scenes. Moreover, an improvement method for the SSD algorithm was proposed.

In the future, a clustering algorithm will be used to cluster the size of feature-bounding boxes suitable for an SSD network, to solve the problems associated with manually setting the size of feature-bounding boxes in the SSD network, and to further increase the detection effect of small objects.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported in part by the Excellent Top-notch Talent Cultivation Funding Project of Colleges and Universities in Anhui Province (Grant No.

gxyqZD2021123), Key Projects of Natural Science Research in Colleges and Universities of Anhui Province (Nos. KJ2020A0361 and KJ2020A0362), University-level scientific research project of Anhui Polytechnic University (No. Xjky2020124), and Education Innovation Fund for Graduate Students of Anhui Polytechnic University.

References

[1] A. Lopez, J. M. Jurado, C. J. Ogayar, and F. R. Feito, "A framework for registering UAV-based imagery for crop-tracking in precision agriculture," *International Journal of Applied Earth Observation and Geoinformation*, vol. 97, 2021.

[2] N. T. Nguyen, M. L. L. Caceres, K. Moritake, S. Kentsch, H. Shu, and Y. Diez, "Individual sick fir tree (Abies mariesii) identification in insect infested forests by means of UAV images and deep learning," *Remote Sensing*, vol. 13, no. 2, 2021.

[3] G. Y. Tian, J. R. Liu, and W. Y. Yang, "A dual neural network for object detection in UAV images," *Neurocomputing*, vol. 443, pp. 292–301, 2021.

[4] M. Krusniak, A. James, A. Flores, and Y. Shang, "A multiple UAV path-planning approach to small object counting with aerial images," in *IEEE International Conference on Consumer Electronics*, pp. 1–6, Las Vegas, NV, USA., 2021.

[5] G. David, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conference on Computer Vision and Pat- tern Recognition*, vol. 2005, no. 1, pp. 886–893, 2005.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2001, no. 1, pp. 511–518, 2001.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[9] S. Q. Ren, K. He, R. Girshick, J. Sun, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition CVPR.*, vol. 2016, pp. 779–788, 2016.

[11] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *European Conference on Computer Vision ECCV*, vol. 1, pp. 21–37, 2016.

[12] F. Yang, H. J. Chen, J. P. Li, F. Li, L. Wang, and X. M. Yan, "Single shot multibox detector with Kalman filter for online pedestrian detection in video," *IEEE Access*, vol. 7, pp. 15478–15488, 2019.

[13] Y. Z. Wang, P. H. Niu, X. Y. Guo, G. W. Yang, and J. Chen, "Single shot multibox detector with deconvolutional region magnification procedure," *IEEE Access*, vol. 9, pp. 47767–47776, 2021.

[14] Z. X. Li, "Zhou FQFSSD: Feature Fusion Single Shot Multibox Detector," 2017, https://arxiv.org/abs/:1712.00960.

[15] Y. D. Liu, Y. T. Wang, S. Y. Wang et al., "CBNet: a novel composite backbone network architecture for object detection," *AAAI Technical Track: Vision*, vol. 34, no. 7, pp. 11653–11660, 2020.

[16] P. Wu and Y. M. Tan, "Estimation of economic indicators using residual neural network ResNet50," *International Conference on Data Mining Workshops*, vol. 2019, pp. 206–209, 2019.

[17] N. Takisawa, S. Yazaki, and H. Ishihata, "Distributed deep learning of ResNet50 and VGG16 with pipeline parallelism," *Eighth International Symposium on Computing and Networking Workshops*, vol. 2020, pp. 130–136, 2020.

[18] A. Çinar, M. Yildirim, and Y. Eroglu, "Classification of pneumonia cell images using improved ResNet50 model," *Traitement du Signal*, vol. 38, no. 1, pp. 165–173, 2021.

[19] P. Hridayami, I. K. G. D. Putra, and K. S. Wibawa, "Fish species recognition using VGG16 deep convolutional neural network," *Journal of Computing Science and Engineering.*, vol. 13, no. 3, pp. 124–130, 2019.

[20] P. C. Xu, J. X. Zhao, and J. Zhang, "Identification of intrinsically disordered protein regions based on deep neural network-VGG16," *Algorithms.*, vol. 14, no. 4, p. 107, 2021.

[21] Z. K. Luo, Z. Fang, S. X. Zheng, Y. B. Wang, and Y. W. Fu, "NMS-loss: learning with non-maximum suppression for crowded pedestrian detection," *International Conference on Multimedia Retrieval*, vol. 2021, pp. 481–485, 2021.

[22] Q. Zhou, C. H. Yu, C. H. Shen, Z. B. Wang, and H. Li, "Object detection made simpler by eliminating heuristic NMS. Computer Vision and Pattern Recognition," 2021, https://arxiv.org/abs/:2101.11782.

[23] B. J. Fan, W. Chen, Y. Cong, and J. D. Tian, "Dual refinement underwater object detection network," *European Conference on Computer Vision ECCV*, vol. 20, pp. 275–291, 2020.

[24] R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision ICCV*, vol. 2015, pp. 1440–1448, 2015.

[25] A. R. Sutanto and D. K. Kang, "A novel diminish smooth L1 loss model with generative adversarial network," in *International Conference on Intelligent Human Computer Interaction*, Springer, Cham, 2020.

[26] A. J. Levine and S. Feizi, "Improved, deterministic smoothing for L1 certified robustness," *38th International Conference on Machine Learning*, vol. 139, pp. 6254–6264, 2021.

[27] C. Herrmann, R. S. Bowen, and R. Zabih, "Channel selection using Gumbel Softmax," in *European Conference on Computer Vision*, pp. 241–257, Springer, Cham, 2020.

[28] L. Q. He, Z. D. Wang, Y. L. Li, and S. J. Wang, "Softmax dissection: towards understanding intra- and inter-class objective for embedding learning," *AAAI Conference on Artificial Intelligence.*, vol. 34, no. 7, pp. 10957–10964, 2020.

[29] D. G. Parthiban, Y. Y. Mao, and D. Inkpen, "On the Softmax bottleneck of recurrent language models," *AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, pp. 13640–13647, 2021.

[30] S. Shetty, "Application of convolutional neural network for image classification on Pascal VOC challenge 2012 dataset," *Computer Vision and Pattern Recognition*, vol. 1, 2016https://arxiv.org/abs/:1607.03785.

[31] D. Du, P. Zhu, L. Wen et al., "The vision meets drone object detection in image challenge results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop ICCV Workshops*, pp. 213–226, Seoul, Korea (South), October 2019.