A Lightweight Remote Sensing Aircraft Object Detection Network

Subjects: Remote Sensing

Contributor: Jiale Wang, Zhe Bai, Ximing Zhang, Yuehong Qiu

Remote sensing aircraft object detection is crucial in various applications. In civil aviation and the aerospace industry, it helps identify other aircraft, drones, or obstacles around an aircraft to prevent collisions and enhance aviation safety. It also aids in the real-time monitoring and tracking of civil aviation flights, cargo planes, and private aircrafts to ensure their flight path and status.

Keywords: deep learning ; lightweight network ; YOLOv5n ; Shufflenet v2

1. Introduction

Remote sensing aircraft object detection is crucial in various applications. In civil aviation and the aerospace industry, it helps identify other aircraft, drones, or obstacles around an aircraft to prevent collisions and enhance aviation safety. It also aids in the real-time monitoring and tracking of civil aviation flights, cargo planes, and private aircrafts to ensure their flight path and status. In the military domain, it identifies and tracks enemy aircraft, performs aerial reconnaissance, gathers intelligence, and supports aerial strikes and combat. In emergencies like aircraft disappearance or deviation from course, it assists in search and rescue operations to locate the aircraft and passengers.

The breadth of application domains in remote sensing object detection corresponds to the complexity of target data sources. Particularly for remote sensing targets, completing object detection tasks depends not only on the algorithm but also on the data source. Factors affecting detection effectiveness include not only the reliability of artificial intelligence algorithms but also that of remote sensing data sources. From acquiring raw data to inputting them into the network, considerations extend beyond complex scenes to factors like imaging conditions, resolution, and storage processes. During imaging, sensor size, atmospheric conditions, observation time, and lighting affect results. For instance, adverse weather like rain and fog significantly degrade imaging quality, necessitating image dehazing during processing. Additionally, image denoising is crucial in preprocessing due to signal transmission, dark current, and random noise effects.

For algorithms, considerations during training involve sample quantity, scene types, target categories, and annotation quality to select high-quality datasets. Enhancing recognition involves designing new network structures, adjusting training strategies, and selecting parameters. Since algorithms often deploy on hardware with low computing power and memory, compressing network volume and parameter size is vital. Given these considerations, selecting a suitable algorithm framework and dataset is paramount.

Currently, remote sensing object detection faces several challenges. Firstly, it suffers from a single detection perspective, limiting the useful information gathered from an overhead view. Secondly, remote sensing targets tend to be small in size with minimal differences between types, making fine-grained identification challenging. Most significantly, the vast range of object detection on remote sensing images leads to substantial computing power consumption and inefficient detection processes. To address these challenges, various algorithms with exceptional performance have emerged in remote sensing object detection, such as the R-CNN series [1][2][3], YOLO series [4][5][6][7][8], and DETR [9]. Nonetheless, deploying these algorithms on mobile hardware platforms with constrained computing and memory resources presents significant challenges due to their large network size and complex structures. In high-altitude environments where real-time imaging and object detection are crucial, available memory and computational resources are severely limited. Addressing actual memory usage, after training these algorithms on the same dataset, the resulting model files typically range from a few megabytes to hundreds of megabytes. Some exceed even hundreds of megabytes, which is unacceptable to a certain extent. From a computational perspective, existing remote sensing aircraft object detection networks prioritize performance metrics, often leading to increased network depth and width. While this may improve performance, it also substantially increases resource consumption in terms of memory and computation. For instance, the actual FLOPs of

YOLOv5n can reach as high as 4.3 G, rendering existing algorithms impractical for deployment on some actual mobile hardware platforms.

In response, several lightweight object detection networks have been proposed. Examples include the MobileNet series [10][11][12], Ghost-Net series [13], PP-LCNet series [14], and Shufflenet series [15][16]. However, each of these algorithms has shortcomings in terms of detection performance, network size, and resource consumption. They fail to strike a balance between detection performance and memory and computational resource utilization. For example, Ghost-Net and Mobile-Net have excessively large network parameter sizes, while PP-LCNet exhibits slightly inferior detection performance. This makes it challenging to meet practical application requirements. Therefore, while ensuring detection performance remains unchanged or slightly decreased, the focus should be on significantly reducing network parameters and computational load to achieve a perfect balance between detection performance and resource consumption.

2. Remote Sensing Object Detection Based on Deep Learning

Around 2014, deep learning-based approaches began to dominate the field of object detection, subsequently extending their influence to the entire domain of remote sensing object detection. Object detection algorithms based on deep learning are typically categorized into two types: single-stage object detection algorithms and two-stage object detection algorithms. [17] Two-stage algorithms mainly include Region-CNN (R-CNN) [1], Fast R-CNN [2], Faster R-CNN [3], and others. They typically start by using a Region Proposal Network (RPN) [3] to generate candidate boxes based on the texture, color, and size features associated with the objects. These candidate boxes undergo a filtering process to reduce their number before being sent to a deep learning network, typically utilizing a Convolutional Neural Network (CNN) [18] for feature extraction. The obtained feature vectors are compared with predefined target categories to confirm the presence of a target and perform target classification. Simultaneously, each candidate box undergoes position regression to obtain corresponding position coordinate information. Due to its excellent detection speed and relatively small resource consumption, single-stage networks have gained dominance in recent years. The most prevalent network among single-stage networks is the YOLO series algorithm, known for its fast detection speed and high accuracy, particularly suitable for real-time scenarios. The single-stage algorithm mainly includes the YOLO series and the Single Shot MultiBox Detector (SSD) [19] series. They generally pass the entire image as input to the CNN, skipping the step of generating candidate boxes. The CNN directly outputs category and location information of the target. Single-stage networks typically use-predefined anchor boxes with different scale sizes and aspect ratios to process targets, judging the presence of a target in each anchor box and predicting its location and category for object detection. With ongoing research, the detection accuracy of single-stage networks continues to improve. With superior detection speed, they have gradually replaced two-stage algorithms in many engineering practices, becoming the mainstream in practical applications.

Traditional methods for object detection in remote sensing images have limited representation power. Recently, many deep learning-based networks specifically for remote sensing images have emerged. Reference [20] proposed a method that redesigns the feature extractor, utilizes a multi-scale object proposal network (MS-OPN) for object-like region generation, and employs an accurate object detection network (AODN) for object detection based on fused feature maps. [21] Reference [22] introduced the CSand-Glass module to replace the residual module in the backbone feature extraction network of YOLOv5, achieving higher accuracy and speed in remote sensing images. Liu et al. proposed the YOLO-extract algorithm [23], which optimizes the model structure of YOLOv5 in two main ways. Firstly, it integrates a new feature extractor with stronger feature extraction ability. Secondly, it incorporates Coordinate Attention into the network.

In recent years, the Transformer [24] has had a profound impact on deep learning. In the field of object detection, Facebook introduced the end-to-end object detection network called DEtection Transformer (DETR) [9], which is based on the Transformer architecture. DETR can be viewed as a transformation process from an image sequence to a set sequence. This is due to the inherent nature of the Transformer as a sequence-to-sequence transformer. The approach taken by DETR involves unfolding the pixels of the output feature map from the backbone into a one-dimensional sequence, treating it as the sequence length, while maintaining the definitions of batch and channel. Consequently, DETR is capable of computing the correlations between each pixel of the feature map and all other pixels, unlike in CNNs, where this is achieved through the receptive field. The Transformer demonstrates the ability to capture a larger perceptual range than CNNs.

In addition to the previously mentioned methods for remote sensing object detection, there is a particular significance in introducing arbitrary-oriented remote sensing object detection methods. From a training perspective, the primary distinction between arbitrary-oriented object detection and regular object detection is in how object bounding boxes are represented in the dataset. Convolutional neural networks struggle to capture variations in scale and orientation of objects in remote sensing images. This struggle arises from the limited generalization ability of convolutional operations to target rotation and scale changes. As a result, the detection performance of convolutional neural networks tends to decrease, especially when dealing with dense objects and remote sensing targets with centrally symmetric features.

Optimizing the loss function for representing object bounding boxes is currently a focus of research. The early work of DRBox [25] has significantly advanced arbitrary-oriented remote sensing object detection. DRBox identified various challenges in this field and introduced three different models along with their parameters tailored for cars, ships, and aircraft, encompassing a wide range of differently sized targets and distinguishing their heads and tails. R3det [26] adopts single-stage object detection and suggests re-encoding the position information of refined bounding boxes into corresponding feature points. This process reconstructs the entire feature map to achieve feature alignment. The ROI-Transformer [27] introduces a module named RoI Transformer, which detects directed and dense objects using supervised RRoI learning and position-sensitive alignment-based feature extraction within a two-stage framework. The innovative application of polar coordinates in the P-RSDet [28] reduces parameter volume and introduces a novel loss function, Polar Ring Area Loss, leading to enhanced detection performance.

3. Lightweight Methods for Object Detection Networks Based on Deep Learning

Deep learning methods have demonstrated significant advancements in remote sensing object detection in recent years. As previously mentioned, the wide scope of object detection in remote sensing images leads to substantial computational overhead and significantly lowers detector efficiency. Consequently, lightweighting network models has become a focal point in current research. The primary goal of lightweighting network models is to reduce model complexity and decrease the number of model parameters. Four primary technical approaches exist for lightweighting network models: compressing pre-trained large models, redesigning lightweight models, accelerating numerical operations, and hardware acceleration.

In current practice, the first three technical approaches are widely employed. Knowledge distillation and model pruning are both well-established methods for compressing models. Knowledge distillation, a common method for model compression, reduces model volume and parameter count by transferring the knowledge of a complex model to a lightweight one. The literature points out that this method extracts the knowledge contained in the complex “teacher” model that has been trained into another lightweight model, the “student” model. In addition, model pruning is also a common method. Model pruning reduces model size by removing unimportant weights or neurons. It can be categorized into structured pruning and unstructured pruning based on different methods. Furthermore, adjusting the number and size of the network’s convolution kernels can achieve model lightweighting. Generally, a larger convolution kernel size can enhance feature extraction. However, the literature indicates that large convolution kernels increase computational requirements and the number of parameters, leading to unstable gradients. Therefore, scholars often employ multiple small convolution kernels instead of a single large one to compress models, as seen in the Visual Geometry Group (VGG) network proposed in [29].

Lightweight networks find extensive applications in remote sensing image processing. The design of lightweight network architecture originated with SqueezeNet [30] in 2016 and MobileNet [10][11][12] in 2017. Subsequently, new and improved networks, such as SqueezeNext [31] and MobileNetV2 [11], have emerged. MobileNetV1 can essentially be viewed as replacing the standard convolutional layer in VGG with depthwise separable convolution [32]. MobileNetV2 [11] introduces shortcut connections and replaces part of ReLU with a linear activation function. They employ pointwise convolution to increase the dimension prior to depth convolution, extract features through depth convolution, reduce the dimension, and add the input and output to form the residual structure. SqueezeNet proposed the Fire module, which replaces the 3 × 3 convolution kernel with a 1 × 1 convolution kernel. By adjusting the number of 1 × 1 convolutions, the number of channels in each layer in the convolution operation can be flexibly controlled, thereby reducing the amount of model calculations. The Shufflenet v1 [15] network, proposed at the same time, is one of the relatively mature lightweight networks. This article uses the improved Shufflenet v2 [16] to initially achieve network lightweighting. In addition to network simplification, numerical calculations have also become a new focus area, with numerical quantification being a typical representative. Quantization is the process of converting a model’s weights and activations from floating-point numbers to lower bit-width integers or fixed-point numbers. Quantization can reduce the memory footprint and computational requirements of a model. The literature indicates that model parameters are the primary memory access objects in CNNs; thus, parameter quantization is an effective means of reducing memory access and power consumption. Hardware acceleration involves utilizing specialized hardware accelerators to expedite the inference process of deep learning models, consequently alleviating the computational load on the model.

4. Attention Mechanism

The attention mechanism, originating from the study of human visual cognition characteristics, represents a significant breakthrough in neural network development. In human visual information processing, due to input information characteristics and human brain processing limitations, it is essential to selectively focus on certain information while ignoring redundant data. For example, in images, focus is on vibrant colors and distinct textures, while in text, attention is on sentence beginnings, endings, and specific keywords. In remote sensing, the attention mechanism adjusts the network's focus on different targets. Moreover, addressing the scarcity of remote sensing datasets, the attention mechanism enriches data, aiding the network in learning more valuable information. Additionally, for multi-source remote sensing images, it integrates diverse information efficiently, enhancing network performance. Therefore, attention mechanisms prioritize which input information to focus on and optimize resource allocation for information processing.

In existing object detection algorithms, the Squeeze-and-Excitation (SE) [33] attention mechanism is widely used in mobile networks. By employing SE modules, the network evaluates relationships between feature channels, determining the importance of each channel through the learning process. It then enhances crucial features for the current task while suppressing less important ones. However, this approach only considers inter-channel information and disregards positional information. Subsequent efforts, like BAM BAM [34] and CBAM [35], aim to incorporate positional information, but convolutional operations capture only local relationships, failing to model crucial long-distance dependencies for visual tasks. To address these limitations and achieve multidimensional information integration, attention mechanisms like Coordinate Attention (CA) and Efficient Channel Attention (ECA) [36] have emerged. Besides focusing on channel and positional information, researchers have explored new approaches, such as evaluating neuron weights and subsequently suppressing or focusing on them based on evaluation results, thus achieving more efficient computations. Representative examples include NAM Attention [37] and SimAM [38]. Furthermore, numerous efficient attention algorithms continue to emerge, including Sequential Attention [39], which emphasizes logical attention, Co-attention [40], which focuses on spatial attention, and the prevalent Transformer based on self-attention.

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv 2015, arXiv:1506.01497.
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
5. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
6. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767.
7. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.
8. Jocher, G. YOLOv5 by Ultralytics. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 August 2023).
9. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
10. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv 2017, arXiv:1704.04861.
11. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
12. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul,

Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.

13. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
14. Cui, C.; Gao, T.; Wei, S.; Du, Y.; Guo, R.; Dong, S.; Lu, B.; Zhou, Y.; Lv, X.; Liu, Q. PP-LCNet: A lightweight CPU convolutional neural network. arXiv 2021, arXiv:2109.15099.
15. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
16. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
17. Jiang, Y.; Tang, Y.; Ying, C.J.E. Finding a Needle in a Haystack: Faint and Small Space Object Detection in 16-Bit Astronomical Images Using a Deep Learning-Based Approach. Electronics 2023, 12, 4820.
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. ACM 2012, 60, 84–90.
19. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14. pp. 21–37.
20. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Multi-scale object detection in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 2018, 145, 3–22.
21. Cheng, S.; Cheng, H.; Yang, R.; Zhou, J.; Li, Z.; Shi, B.; Lee, M.; Ma, Q.J.P. A High Performance Wheat Disease Detection Based on Position Information. Plants 2023, 12, 1191.
22. Luo, S.; Yu, J.; Xi, Y.; Liao, X.J.I.A. Aircraft target detection in remote sensing images based on improved YOLOv5. IEEE Access 2022, 10, 5184–5192.
23. Liu, Z.; Gao, Y.; Du, Q.; Chen, M.; Lv, W.J.I.A. YOLO-extract: Improved YOLOv5 for aircraft object detection in remote sensing images. IEEE Access 2023, 11, 1742–1751.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. arXiv 2017, arXiv:1706.03762.
25. Liu, L.; Pan, Z.; Lei, B. Learning a rotation invariant detector with rotatable bounding box. arXiv 2017, arXiv:1711.09405.
26. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; pp. 3163–3171.
27. Ding, J.; Xue, N.; Long, Y.; Xia, G.-S.; Lu, Q. Learning RoI transformer for oriented object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2849–2858.
28. Zhou, L.; Wei, H.; Li, H.; Zhao, W.; Zhang, Y.; Zhang, Y. Arbitrary-oriented object detection in remote sensing images based on polar coordinates. IEEE Access 2020, 8, 223373–223384.
29. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556.
30. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv 2016, arXiv:1602.07360.
31. Gholami, A.; Kwon, K.; Wu, B.; Tai, Z.; Yue, X.; Jin, P.; Zhao, S.; Keutzer, K. Squeezenext: Hardware-aware neural network design. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1638–1647.
32. Sifre, L.; Mallat, S. Rigid-motion scattering for texture classification. arXiv 2014, arXiv:1403.1687.
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
34. Park, J.; Woo, S.; Lee, J.-Y.; Kweon, I.S. Bam: Bottleneck attention module. arXiv 2018, arXiv:1807.06514.
35. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.

36. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
37. Liu, Y.; Shao, Z.; Teng, Y.; Hoffmann, N. NAM: Normalization-based attention module. arXiv 2021, arXiv:2111.12419.
38. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 11863–11874.
39. Chen, Q.; Wang, W. Sequential attention-based network for noetic end-to-end response selection. arXiv 2019, arXiv:1901.02609.
40. Feng, G.; Hu, Z.; Zhang, L.; Lu, H. Encoder fusion network with co-attention embedding for referring image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15506–15515.

Retrieved from https://encyclopedia.pub/entry/history/show/126539