IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 9, 2020, accepted April 29, 2020, date of publication May 11, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993998

An Efficient Feature Pyramid Network for Object Detection in Remote Sensing Imagery

FANG QINGYUN[1], ZHANG LIN[2], AND WANG ZHAOKUI[1], (Member, IEEE)
[1]School of Aerospace Engineering, Tsinghua University, Beijing 100084, China
[2]Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA

Corresponding author: Wang Zhaokui (wangzk@tsinghua.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant No. 11872034 and the Key Laboratory of Intelligent Infrared Perception, Chinese Academy of Sciences.

ABSTRACT Scale diversity, small target, and power limitation have made remote sensing imagery a challenging field in object detection on satellites. Aiming at the aspects of scale diversity and small target, this paper provides a novel feature pyramid network with Adaptive Residual Spatial Bi-Fusion (ARSF) as a solution. ARSF nets introduce a robust fusion of multi-scale semantic information and fine spatial details. A spatial feature fusion module designed in networks with ARSF adapts to object size variation by learning the most crucial feature maps. Comparing to the original feature pyramid network, a shorter critical path for information transmission is formed in our method. Experiments show that a validation instance of YOLOv3-ARSF can achieve a state-of-the-art performance of 85.8 mAP on the NWPU-VHR10 dataset. YOLOv3-ARSF only 3MB larger than YOLOv3 but far exceeds YOLOv3 by 2.3% mAP, which shows our ARSF is efficient. As for the last challenge, two lightweight versions, ARSF(lite) and ARSF(lite+) are also validated for future research of online object detection on satellites in aerospace engineering. Visualizations and details are provided for a more comprehensive understanding.

INDEX TERMS Computer vision, object detection, remote sensing, satellites, aerospace engineering.

I. INTRODUCTION

With the rapid development of remote sensing technology, massive remote sensing image data have been generated by satellites. Object detection in remote sensing imagery has then kept being a hot topic in academic research with broad applications due to the critical value of those images. Object detection not only determines the class of interest but also gives the location information of each prediction. Objects to be detected in this field are generally human-made targets such as aircraft, ships, vehicles, etc., and they have noticeable differences with the background. Besides, for major emergency tasks such as fire alarms, search and rescue of marine vessels, and assessment of earthquakes, volcanoes, and tsunami disasters, it will take too long, if the ground station processes the information returned by the satellites, to miss the golden time of search and rescue. Therefore, online detecting on satellites has become a vital development direction of remote sensing technology in the future.

However, different from general object detection, remote sensing object detection on satellites still yields challenge by [1], [2]:

- **Scale Diversity**: Remote sensing images are taken from a few hundred kilometers to tens of thousands of kilometers, and the ground objects are of different sizes, even for the same class of objects. For example, a large ship in the port is more than 300 meters long, and a small vessel only has tens of meters;
- **Small Object**: Many of the objects in remote sensing images are small size (tens or even a few pixels), which leads to a small amount of information.
- **Power Limitation**: The future development direction of remote sensing object detection is online on satellites, but due to the power limitation, the processor (especially ARM CPU) and memory of satellites are minimal. Therefore, the network scale and calculation volume cannot be large.

The state-of-the-art object detection solutions [3]–[7] adopted the method of increasing the network depth and designing better feature pyramid network(FPN) [3] to

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

enhance the fine details and meaningful semantic information of the multi-scale objects, thereby improving the detection accuracy of the CNN.

Although these advanced networks provide a more robust feature pyramid network in multi-scale detection, however, they are still not sufficient for small object detection. To make the detector suitable for objects of various sizes, especially small objects, a new and effective feature pyramid network structure, named Adaptive Residual Spatial Bi-Fusion (ARSF), is proposed in this paper.

Like PANet [6], the ARSF feature pyramid network is bidirectional to fuse deep semantic features and shallow spatial features robustly. A shortcut module, also called residual module, is adopted on ARSF lateral connection to shorten the information transmission path further. Because of this "residual" learning nature, the ARSF network becomes more accessible to train and converge. Besides this, an innovative Adaptive Spatial Fusion(ASF) module is introduced in the ARSF network, which can adaptively learn the most useful features map for the head of detectors.

As for the limited power, although many papers are applying deep learning methods to achieve object detection in remote sensing images [8]–[11], the network scale and calculation volume of these papers are plentiful, and it is still challenging to complete the efficient detection on satellites with limited onboard memory and computing power. To handle this problem, we take advantage of efficient convolution and quantization to achieve ARSF(lite) and ARSF(lite+) two versions for edge devices such as satellites to save the computational cost. In our experimental results, the parameters and floating-point operations (Flops) of these two versions are four times smaller than that of the previous FPN [3] and SPP [7], and the memory overhead is reduced by two times.

Furthermore, to better understand how the ARSF network proposed in this paper adaptively fuse various features, we visualize some examples to help analyze and understand it.

The rest of this paper is organized as follows. Some related work is introduced in section II. Our ARSF architecture and feature visualization for remote sensing visions are given in section III. The experimental results and discussions are presented in section IV. Finally, the conclusions are drawn in section V.

II. RELATED WORK

A. ADVANCED DETECTORS

In recent years, deep learning has achieved great success in many computer vision tasks, including object detection with its deep semantic features. R-CNN [12], Faster R-CNN [13], Mask R-CNN [14], etc., which pursue the accuracy of the ‘‘two-stage’’ network, as well as SSD [15], DSSD [16], YOLO [17] and YOLOv3 [18], which pursue higher efficiency and ‘‘single-stage’’, have significantly promoted the development of this field.

Since this article aims at edge devices with limited memory and computing power, the following work focuses on the ‘‘one-stage’’ detectors that pay more attention to the trade-off between efficiency and performance. In particular, YOLOv3 [18], which can achieve almost accuracy as the ‘‘two-stage’’ network, had a clear advantage in speed. Compared with the previous YOLO [17] and YOLO-9000 [19], YOLOv3 designed a powerful backbone DarkNet53 and a multi-scale object detection structure. Darknet53 was much faster than ResNet152 while their accuracy was close. The multi-scale network structure alleviated the problems of coarse and weak detection on small objects.

B. FEATURE PYRAMID NETWORK

Since some pooling layers are repeatedly applying in CNN to extract advanced semantics, the information of small objects can be filtered out during the downsampling process.

To cope with this problem, FPN [3] utilized a top-down path module to fuse different level features, which noticeably increases the performance of detectors. Top features preserved semantic discrimination, while bottom features retained spatial information. In the top-down pathway, the high-level semantic information is integrated into the low-level spatial information through upsampling, which enhances the features of different levels. Nevertheless, for large and medium objects, due to the invariance of convolutional translation and the long pathway between high-level features and low-level features (as shown by the gray dotted line in Fig.1), it is difficult to locate them accurately.

Since then, subsequent variations of FPN such as PFPN [4], Panoptic FPN [5], PANet [6], etc. were proposed. PFPN was inspired by SPP net [7], and applied the multi-scale context aggregation (MSCA) module for feature fusion to improve the accuracy of detection. Panoptic FPN [5] supplied a semantic segmentation branch using a finely-designed feature pyramid network for Mask R-CNN to complete the tasks of instance segmentation and semantic segmentation. PANet [6] created a bottom-up path enhancement module based on the top-down path module in FPN, which was designed to shorten the path of information transmission and maintain the accurate positioning information in low-level features. This creative bidirectional structure strengthened the ability of the feature pyramid network.

C. LIGHTWEIGHT NETWORK

Although the learning feature capabilities of advanced CNNs (such as AlexNet [21],ResNet [20], GoogLeNet [22], and DenseNet [23], etc.) are being continuously enhanced by the deeper the network layers. However, in engineering, model size and computational cost also need to be considered. Deep convolutional neural networks include dozens or even hundreds of layers with a large number of weight parameters. Saving these huge weight parameters has high requirements on the device memory. In order to handle this problem, a normal method is model compression; however, lightweight network blazes a new trail. Many efficient convolutional calculations are designed in lightweight networks to reduce parameters without compromising performance.

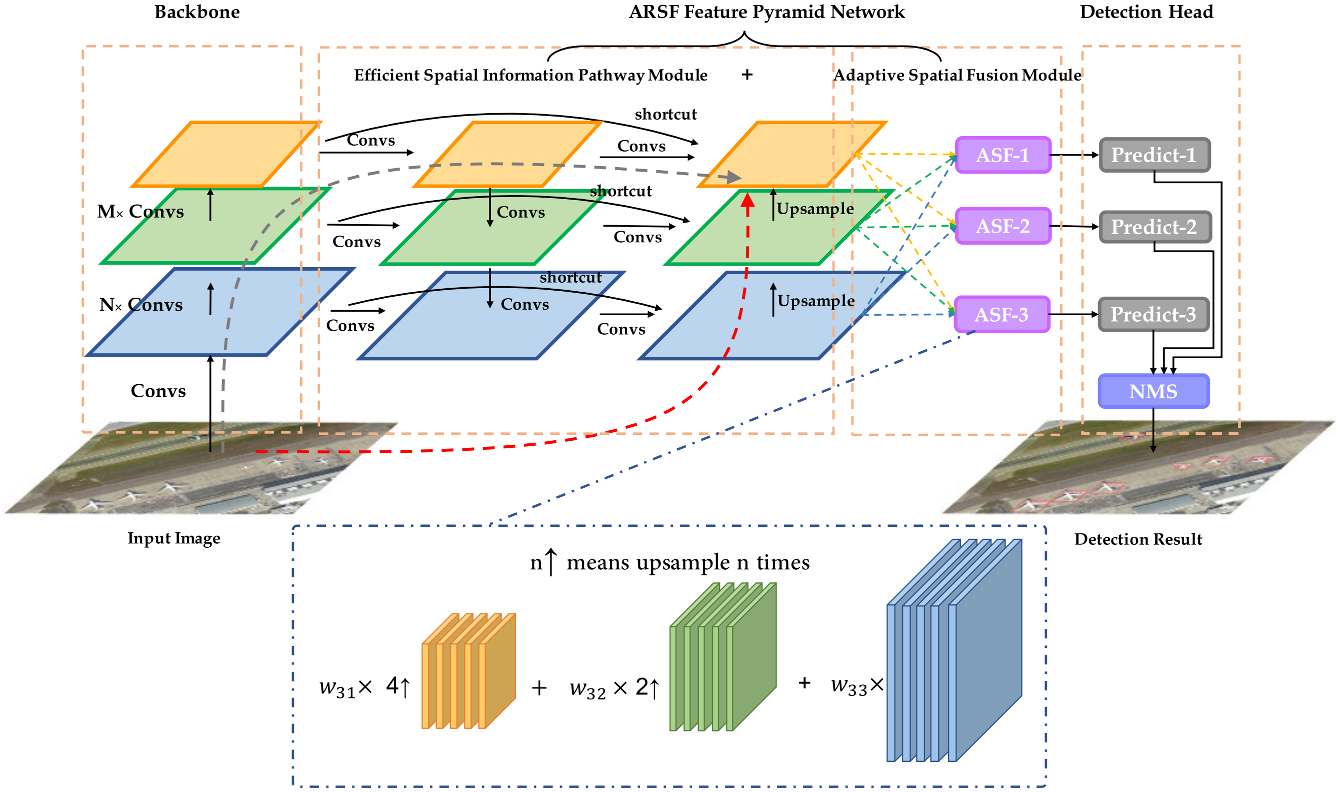


FIGURE 1. The architecture of YOLOv3 with ARSF Network. If the arrow is not marked, it means a simple transfer, no additional operations.

In recent years, many lightweight networks have been proposed, but mainly the following four lightweight networks: SqueezeNet [24], ShuffleNet [25], [26] MobileNet [27]–[29], Xception [30].

A more efficient convolutional implementation in a lightweight network can significantly reduce parameters and Flops. Therefore, in future engineering, lightweight networks will become mainstream.

III. THE ARCHITECTURE OF ARSF

Compared with the previous FPN and variants, our ARSF has more effective spatial information propagation and more powerful semantic extraction and can adaptively learn the most useful features for different size objects. In this article, we only take ARSF combined with YOLOv3 as an example to show that it is powerful and effective. In fact, ARSF can be worked as a plug-in to embed in most mainstream detectors, like SSD, Faster R-CNN, Mask R-CNN, etc.

A. ADAPTIVE RESIDUAL SPATIAL BI-FUSION

As shown in Fig.1, the main structure of ARSF is divided into two parts: an efficient spatial information pathway(ESIP) module and an adaptive spatial fusion(ASF) module.

1) EFFICIENT SPATIAL INFORMATION PATHWAY MODULE

In addition to the original top-down pathway(gray dashed in Fig. 1) introduced in FPN [3], we propose a novel bottom-up pathway as indicated by the red dashed line in Fig. 1 is much shorter than the gray one that needs to go through N + M convolutions. The shorter spatial information pathway can make better use of the accurate location information stored in the low-level features. We adopt a ‘‘residual’’ shortcut module in the lateral connection to further shorten the pathway since this kind of module has been proved to improve the networks' overall performance in many studies [18], [20], [24], [28].

2) ADAPTIVE SPATIAL FUSION MODULE

Recent object detection networks [31], [32] usually utilize feature fusion to improve the accuracy of detecting different sizes object. However, this research resizes different features to the same resolution and then adds them up simply. This method of treating different resolution features without discrimination is too rough. It ignores that the contribution of different resolution features to the object detection of various sizes is inconsistent.

For example, low-resolution features that have sizeable receptive field and high-level semantic information contribute more to identifying large-scale objects than high-resolution features. In contrast, high-resolution features have smaller receptive fields but accurate localization for small-scale objects. Additional weight is introduced in this paper to reconcile the different resolution features. By training this weight, the network can adaptively learn the importance of different resolution features for detectors.

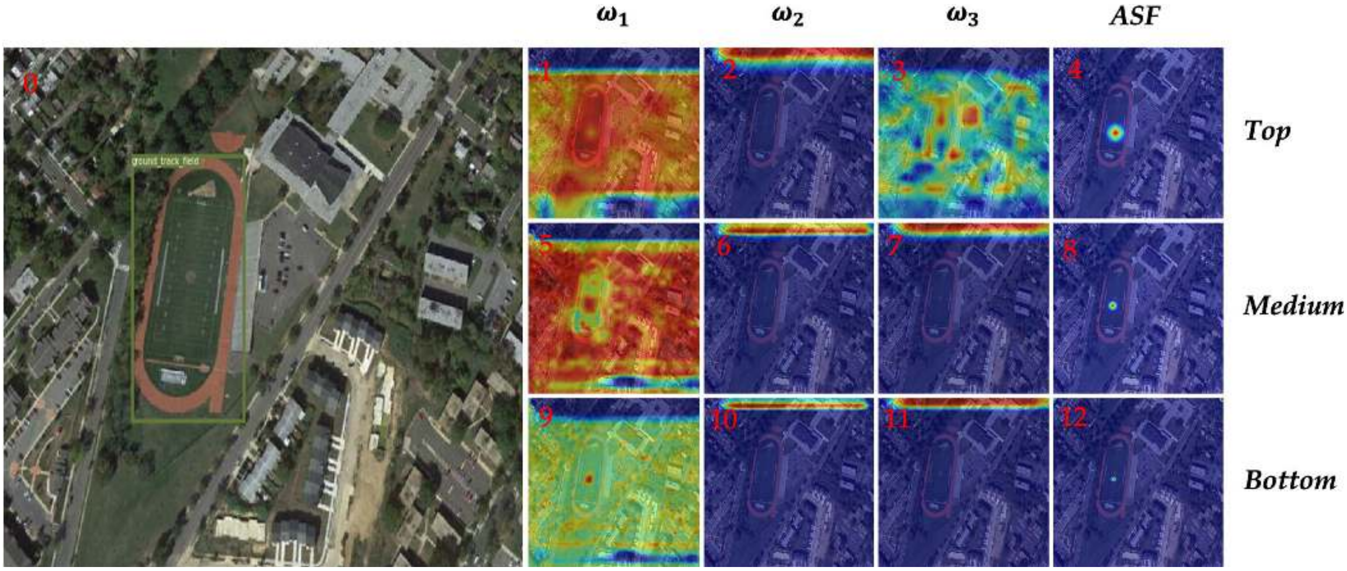


FIGURE 2. Grad-CAM image of large-scale object detection by ARSF net.

Based on this idea, we have the following formula:

$$O_i = \sum_{j=1}^{n} w_{ij} \cdot I_j \tag{1}$$

where O_i is the i-th detection output, I_j is the j-th feature fusion input, and w_{ij} is the weight of the j-th input to the i-th output. Taking ASF-3 as an example in Fig.1, $O_3 = w_{31} \times I_1(4 \uparrow) + w_{32} \times I_2(2 \uparrow) + w_{33} \times I_3$, where $n \uparrow$ means feature maps should upsample n times.

3) LIGHTWEIGHT VERSION

ARSF(lite) has the same structure as ARSF, except that the regular convolutions are replaced with the Depthwise Separable Convolution [27]–[30], which makes the feature network more compact. Depthwise Separable Convolution is a lightweight convolution, whose parameter amount is only 1/9 of the regular convolution, and the amount of multiplication is only $1/c + 1/9$, where c is the input channel number. ARSF(lite) based on this efficient convolution will much streamline scale and computational load. This scheme is memory, CPU, and GPU-saving for most embedded devices.

To further reduce the computational cost, we propose ARSF(lite+) that is a version of the 16-bit quantization technology taken on ARSF (lite). This quantization converts weights to 16-bit floating-point values during model conversion from 32-bit. In theory, this results in a 2x reduction in model size. Some hardware, like GPUs, can compute natively in this reduced precision arithmetic, realizing a speedup over traditional floating-point execution.

B. VISUAL EXPLANATION

In deep learning, convolutional neural networks are often treated as black boxes, and it is difficult for people to explore the mathematical nature behind them. Therefore, in recent years, the interpretability of neural networks has become an important research direction in the computer vision field.

For a better understanding of how the ARSF performs adaptive feature fusion, some examples are visualized using Grad-CAM [33] technology to help analyze and understand it.

Grad-CAM essentially takes gradients as weight factors to measure which pixels in the feature maps have the greatest impact on detection. Assuming that our detection target is an airplane, its steps are divided into three steps:

1. First, calculate the partial derivative of the probability y^{plane} of the plane with respect to all pixels A_{ij} in the feature map of the last layer, that is,

$$\frac{\partial y^{plane}}{\partial A_{ij}^{k}} \tag{2}$$

   where k is the number of the channel of the feature maps, and i,j represent the i-th row and j-th column,respectively.
2. Then, average the partial derivatives of each pixel in the feature maps which similarly to the global average pooling(GAP),

$$\alpha_k^{plane} = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^{plane}}{\partial A_{ij}^{k}} \tag{3}$$

   where α_k^{plane} is the weight of the k-th channel of the feature maps when discriminating as plane targets, and Z is a constant (number of pixels in the feature map).
3. Finally, linearly combine each channels of the feature maps A with α_k^{plane} as the weight, and then pass a ReLU

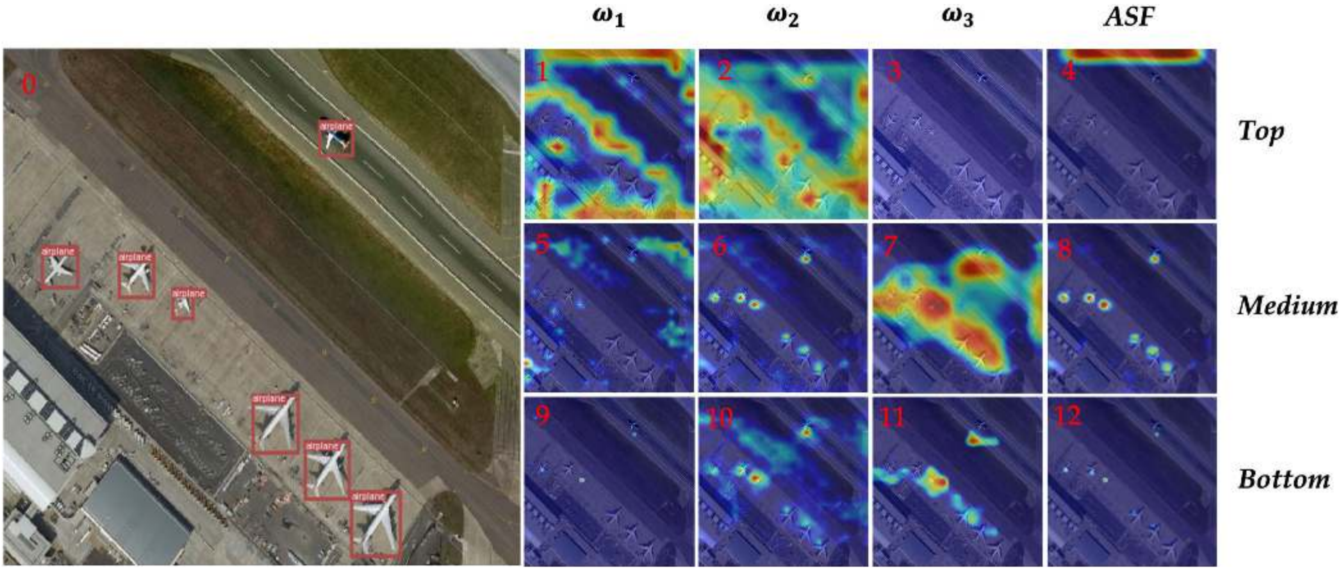


FIGURE 3. Grad-CAM image of medium-scale object detection by ARSF net.

function, that is:

$$L^{plane}_{Grad-CAM} = \mathrm{ReLU}(\sum_{k} \alpha_k^{plane} A^k) \qquad (4)$$

where $L^{plane}_{Grad-CAM}$ is an activation map, just like a heat map, through Grad-CAM for planes. Notice, we should resize this activation map to the size of the input image, and then overlay it with the input image to get the final result.

Here we show three examples (Fig.2, Fig.3, and Fig.4) representing the detection of three sizes objects: large, medium, and small. In Fig.2 and Fig.3, there are 13 subfigures, and the number of each subfigure is given by the red number in the upper left corner.

Subfigure 1, 6, and 11 are activation maps of three sizes original feature maps(i.e., yellow, green, and blue feature maps in the rightmost column of ESIP module in Fig.1). Subfigure 2, 3, 5, 7, 9, and 10 are up-sampled or down-sampled from the original feature maps to get the same size as the original feature maps in the same row. In the remaining right column, subfigure 4, 8, and 12 are activation maps obtained by the ASF module of three feature maps in the same row.

In Grad-CAM images, the red areas represent the key pixels that the network pays more attention to. More specifically, these pixels have a big influence on detecting whether there are objects in the vicinity. Naturally, the closer these red pixels are to objects, the better. It should be noted that the actual sizes of the top, medium, and bottom feature maps are different, and they are unified to the same size here only for better visualization.

YOLOv3 only takes the center point of the object in the corresponding feature maps as the detection point [18]. For a large-scale object such as the playground in Fig.2, the network also only detects the center part. In the first row of Fig.2, it is clear that the gradients learned in subfigure 1 and 3 are negative because the center of the object is relatively weak to the surroundings. According to the ASF activation map of subfigure 4, it only makes sense when the weights w_1 and w_3 are also negative.

Compared with the medium and bottom levels, the top-level activation map(i.e., subfigure 4) has more red pixels. It illustrates that the top-level features are more suitable for detecting large-sized objects.

For the medium-sized aircraft detection in Fig.3, all the airplanes of the top-level fused features are filtered as background. The detection of bottom-level features depends on too few pixels to omit some planes. In contrast, the medium-level activation map has highlighted pixels around all airplanes, clearly showing the critical areas that the network pays attention to. These key areas are also near the center position of the airplanes. It is verified again that YOLOv3 is a center-point-based detection method, and it also reflects the power of the ARSF net to locate the objects accurately.

Different from Fig.2 and Fig.3, in Fig.4 we have taken a part of the original image with a red dashed frame to enlarge it for better visualization, as well as 3 level ASF activation maps. The aircraft object in this figure is much smaller than before, so the aircraft object will be filtered by the network as the background in the top-level and medium-level fused features, resulting in missed detection. However, the bottom-level features are more accurate because of its larger size and higher resolution, which is suitable for small object detection. The activation map of the bottom-level features also verifies it.

In general, these three-level feature maps are targeted at different sizes of large, medium, and small objects. The adaptive feature fusion and multi-scale detection methods fully utilize semantic and spatial information to improve the final performance of the detectors.

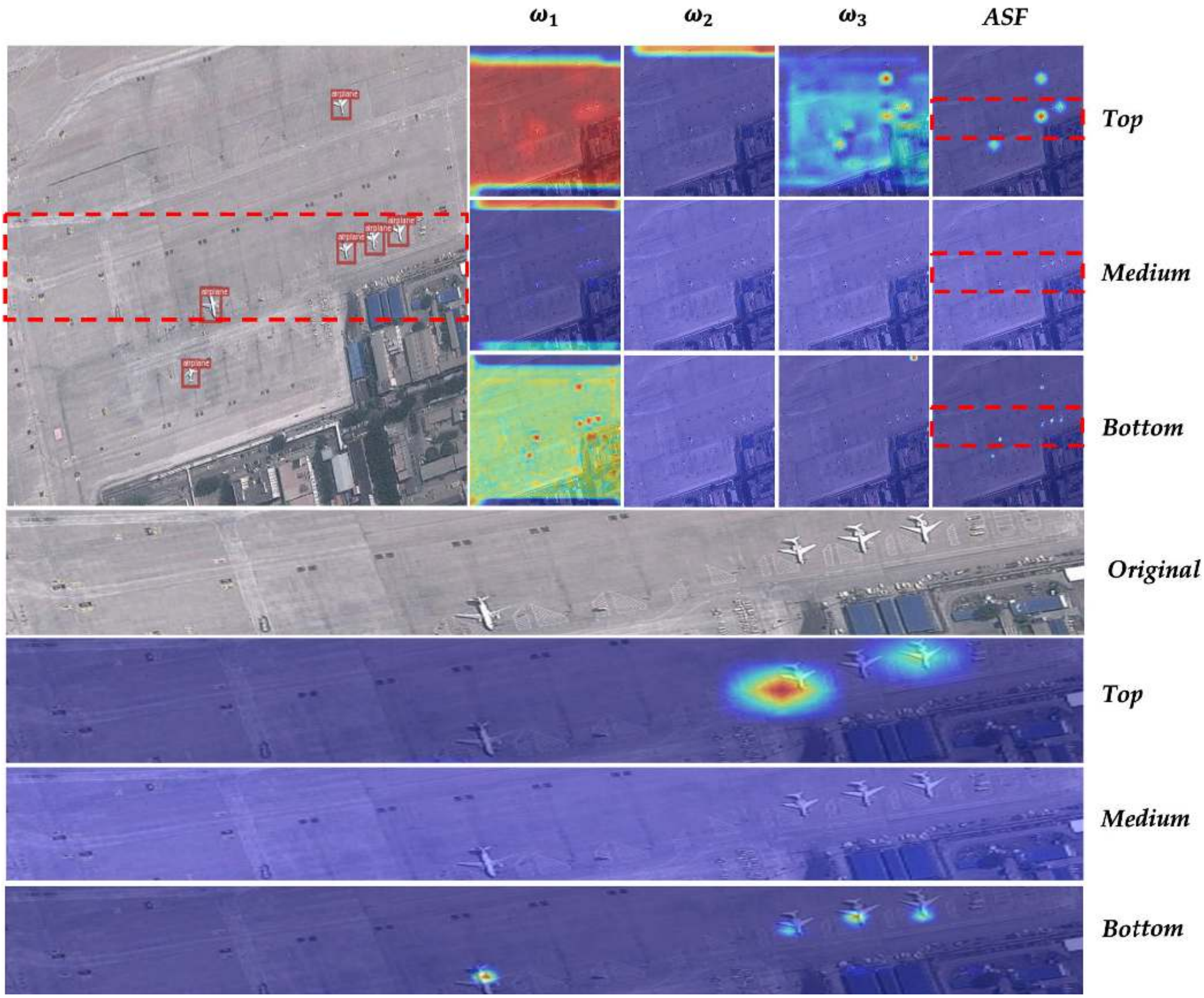


FIGURE 4. Grad-CAM image of small-scale object detection by ARSF net.

IV. PERFORMANCE EXPERIMENTS

A. DATASET AND EXPERIMENTAL SETTINGS

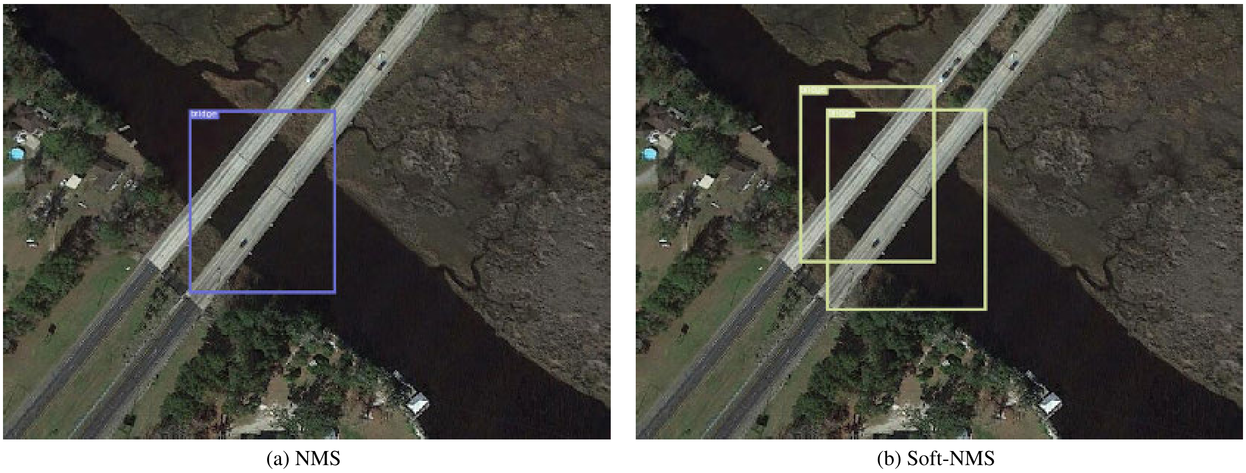
In experiments, we take the NWPU-VHR10 [34] to prove that our proposed ARSF network achieves the performance of SoTA detectors. The NWPU VHR-10 dataset contains 650 annotated images, 757 airplanes, 302 ships, 655 storage tanks, 390 baseball diamonds, 524 tennis courts, 150 basketball courts, 163 ground track fields, 224 harbors, 124 bridges, and 598 vehicles, totally ten classes and 6,686 objects. To verify that our proposed ARSF is suitable for multiple sizes of objects, especially for small size, we have stipulated three sizes of large, medium, and small for the objects in the dataset. The specified standards are in Table 1, and the area of objects is relative to the entire image.

The hardware and software platforms applied in the experiments are configured as follows, Intel (R) Core (TM) i9-7900X @3.30 GHz(CPU), NVIDIA Titanxp 12G (GPU), Gloway DDR4 16G (Memory), Samsung 960 Pro 512G (SSD), Ubuntu16.04 LTS (System), and Pytorch(Deep learning framework).

TABLE 1. A description of the object sizes in the NWPU VHR-10.

	min rectangle area	max rectangle area
small object	0×0	0.05×0.05
medium object	0.05×0.05	0.2×0.2
large object	0.2×0.2	1.0×1.0

Our all detectors were trained by using the Adam algorithm, where the initial learning rate was 1e-4 for Warmup Cosine Annealing Learning Rate Schedule (WCALR), and the other initial learning rates are 1e-3. The total epoch is 200, and the batch size is 8 for all training.



(a) NMS

(b) Soft-NMS

FIGURE 5. soft-NMS and NMS comparison example.

B. EVALUATION INDICATORS

In order to evaluate the performance of network detection, Precision, Recall, Average Precision(AP), and Mean Average Precision (mAP) are adopted.

Precision refers to the proportion of true positives of detection.

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{5}$$

Recall measures the ratio of positives that are correctly detected to total positives.

$$\text{Recall} = \frac{TP}{(TP + FN)} \tag{6}$$

If the IoU between the prediction bounding box and the ground truth is larger than 0.5, it will be considered as true positive (TP); otherwise, it will be considered as false positive (FP). False negative(FN) means that there is a right target here, but the network does not detect it.

Average Precision(AP) is the integral of the precision-recall curve for each category.

$$AP = \int_0^1 \frac{FP}{TN + FP} d \frac{TP}{TP + FN} \tag{7}$$

mean Average Precision(mAP) computes the mean of all the AP values for all categories.

C. COST-FREE TRICKS

In the field of object detection, one can improve the network detection performance by increasing the input size of images, selecting a deeper and stronger backbone, or constructing a much more sophisticated feature network. However, these methods for improving precision have some costs, and it will increase the network parameters and Flops, make network training difficult, and extend the forward inference time of the detector.

Besides, there are also some cost-free tricks in object detection tasks. They improve the ability of detection without increasing the model size of the network and naturally will not extend the inference time. In this article, we take K-means [18], soft-NMS [35], and Warmup Cosine Learning Rate schedule [36] three cost-free tricks on the YOLOv3 baseline to improve the precision.

1) K-MEANS

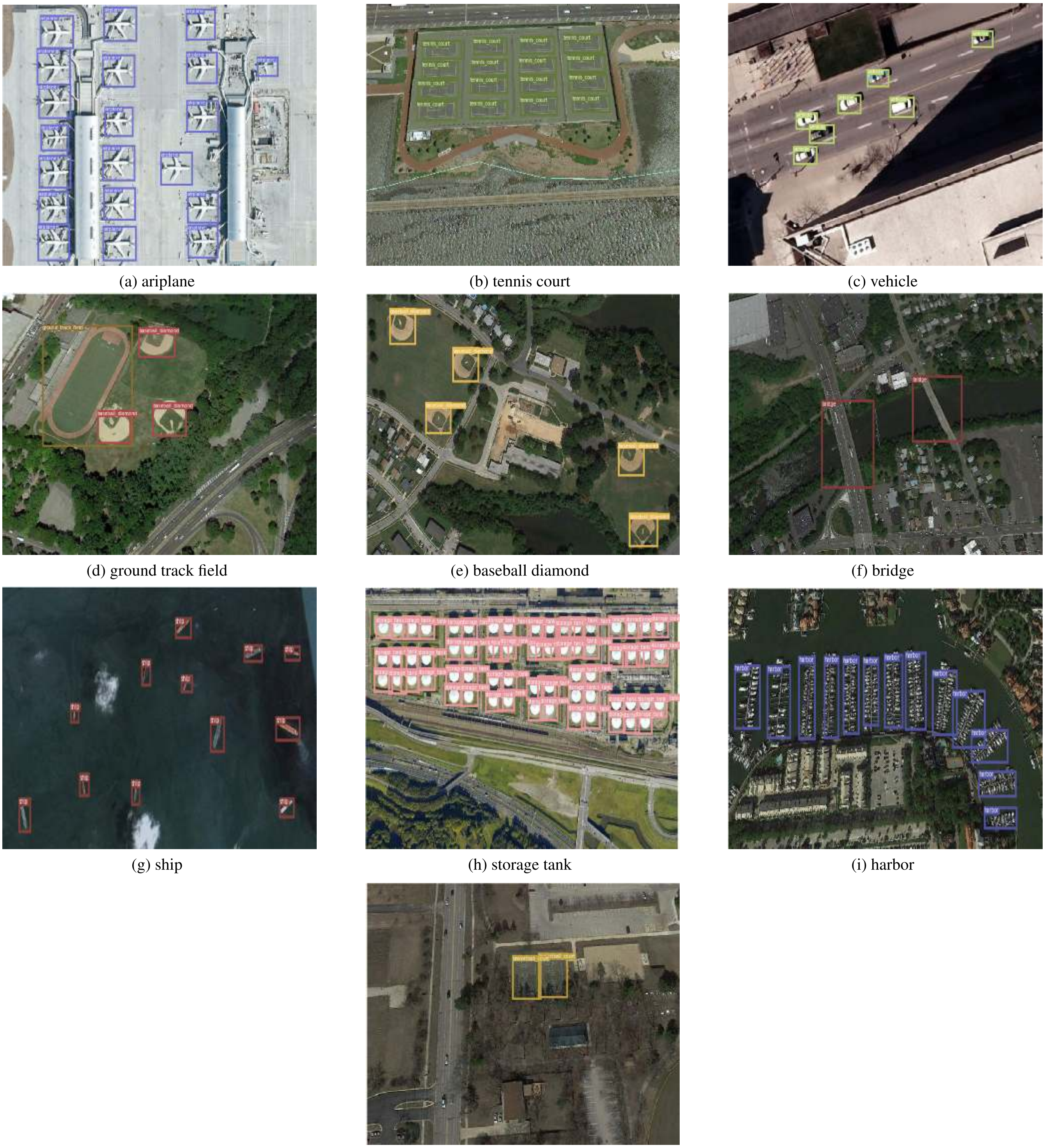
In the previous Faster-RCNN [13] and SSD [15] algorithms, we need to select the prior scales of bounding boxes manually. Obviously, manual selection is too subjective. The K-means method can automatically select more accurate and representative bounding boxes through clustering the scales of all targets in the training set, making it better for the convolutional neural network to detect objects.

2) SOFT-NMS

Non-Maximum Suppression (NMS) algorithms are necessary for current object detection algorithms to eliminate a large number of redundant candidate bounding boxes. The NMS algorithm simply sets a threshold, forcing the confidence of candidate boxes to zero when IoU values larger than the threshold. Obviously, this method is too rough to cause overlapping but true targets to be missed, thereby increasing the rate of false alarms. The soft-NMS algorithm [35] sets an overlap area-based Gaussian penalty function for adjacent detection bounding boxes, rather than simply setting their confidence to zero. Take the two overlapping bridges in Fig. 5 as an example, soft-NMS can detect the two bridges very well. However, the traditional NMS simply and rudely set the threshold, so that the other bridge was dropped.

3) WARMUP COSINE ANNEALING LEARNING RATE SCHEDULE

From experience and intuition, during training, when the loss is getting smaller and smaller, the hyperparameter learning rate should be correspondingly reduced. Especially when it is near the optimum point, the learning rate should be lower to avoid oscillation. The common method is Piecewise Constant



(a) ariplane

(b) tennis court

(c) vehicle

(d) ground track field

(e) baseball diamond

(f) bridge

(g) ship

(h) storage tank

(i) harbor

(j) basketball court

FIGURE 6. Visualization of 10 classes object detection examples by YOLOv3-ARSF networks on NWPU VHR-10.

Decay, also called Step Decay. After reaching the pre-defined training epochs or iterations, the learning rate is multiplied by a constant less than 1 (usually 0.1) to decrease. This schedule is adopted in training FasterRCNN [13] and YOLOv3 [18]. However, it has a sharp learning transition period, which may cause the optimizer to re-stabilize the learning momentum in the next some iterations [36]. In contrast, the Warmup Cosine Annealing Learning Rate (WCALR) schedule consists of two stages, Warmup and Cosine Annealing Learning. In the first stage, the model can gradually stabilize under the low

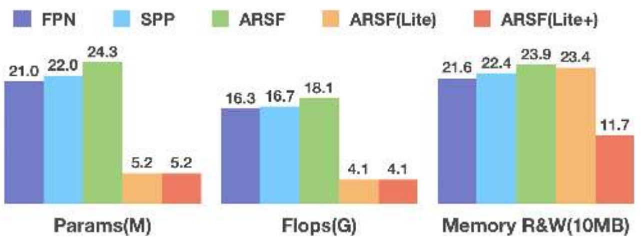


FIGURE 7. Comparison of ARSF, SPP and FPN in parameters, floating point operations, and memory read and write.

TABLE 2. Cost-free tricks for improving YOLOv3 baseline on NWPU VHR-10.

Baseline	Kmeans	Soft-NMS	WCALR	mAP
✓				80.5
✓	✓			82.4
✓	✓	✓		82.7
✓	✓	✓	✓	83.5

warmup learning rate to avoid violent oscillation. In the second stage, since the model is relatively stable, a smoother cosine-shape learning rate adjustment strategy is taken. The two-pronged approach makes the model convergence faster and performs better. The following formula can describe it:

$$\alpha_t = \begin{cases} \dfrac{t}{T'}\alpha_0 & 1 \leq t \leq T' \\ \dfrac{1}{2}\alpha_0 \left(1 + \cos\left(\dfrac{(t-T')\pi}{T}\right)\right) & T' \leq t \leq T \end{cases} \tag{8}$$

where t is the current epoch or iteration, α_t is the learning rate of the tth epoch or iteration, T' is total epochs or iterations of warmup-stage, T is total epochs or iterations, and α_0 is the defined max learning rate.

Table 2 shows the improvement of the stronger YOLOv3 baseline by three cost-free tricks. These three tricks together improve the detection precision by 3%, which is a considerable increase. Kmeans and WACLR increase mAP by 1.9% and 0.8% respectively, while soft-NMS increases a little by 0.3%.

D. EXPERIMENTS RESULT

To demonstrate that our proposed ARSF network is powerful, we combine it with the YOLOv3 detector called YOLOv3-ARSF. Experiments on the test set prove that our ARSF network can handle the two challenges of scale diversity and small objects in remote sensing images, especially to improve the detection of small and densely packed objects. Fig. 6 is the visualization of some results of test set detection. In general, the bounding boxes generated by the new detector proposed in this paper can cover almost all targets well, especially in Fig. 6 (h, i), when the targets are densely packed, small, or both, the detector can still detect them without missing.

TABLE 3. The AP value of objects of different sizes.

Item	YOLOv3	YOLOv3-SPP	YOLOv3-ARSF
AP_s	51.3	46.8	**53.0**
AP_m	74.1	77.2	**77.5**
AP_l	81.8	**84.6**	82.9
mAP	83.5	84.4	**85.8**

For further proof, we show that the AP values of the small, medium and large objects (described in Table 1) are in the following Table 3. The best AP value of each item is bold in Table. Compared with YOLOv3 and YOLOv3-SPP, our proposed YOLOv3-ARSF has advantages in the detection of small and medium objects, especially in small objects. It is 1.7 higher than YOLOv3 and 6.2 higher than YOLOv3-SPP, which is a significant improvement.

In Table 4, we show the quantitative comparisons measured by AP values from SSD, DSSD, YOLOv3, YOLOv3 improved by tricks and our proposed detector YOLOv3-ARSF. The best AP value of each category is bold in Table. Among all the methods in Table, the proposed YOLOv3-ARSF, once fine-tuned on the Darknet53 ImageNet pre-trained model and boosted up by cost-free tricks, achieved the best performance of one-stage detectors and pushed the benchmark into 85.8. Owing to the ESPI module and the ASF module, YOLOv3-ARSF performs better than original YOLOv3 on objects of various sizes, including airplane(95.4 to 96.1), ship(87.1 to 88.7), storage tank(70.9 to 84.5), tennis court(73.2 to 81.1), basketball court(81.2 to 83.0), ground track field(96.2 to 98.3), harbor(85.6 to 87.2), barge(60.6 to 76.4) and vehicle(56.1 to 63.2).

To illustrate the efficiency of our method, FPN, SPP, ARSF, ARSF (lite), and ARSF (lite+) are compared in terms of parameters, Flops, and memory read and write, as shown in Fig. 7. ARSF slightly exceeds the FPN and SPP in these three terms. However, using both Depthwise Separable

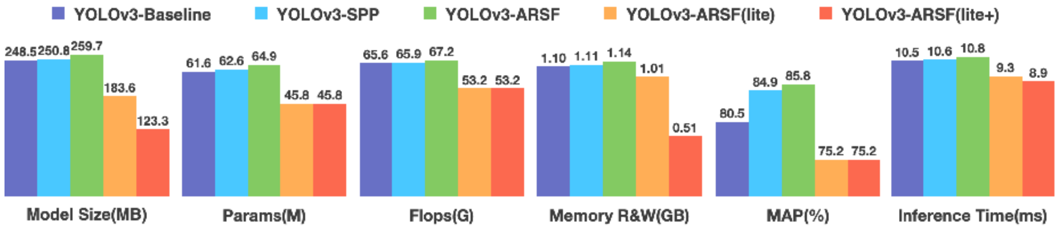


FIGURE 8. Comparison of YOLOv3 baseline, YOLOv3-SPP, YOLOv3-ARSF,YOLOv3-ARSF(lite) and YOLOv3-ARSF(lite+) on NWPU VHR-10.

TABLE 4. The mAP (mean Average Precision) values of our proposed method and others on NWPU VHR-10.

Category	SSD300 [8], [15]	SSD512 [8], [15]	DSSD321 [8], [16]	YOLOv3 416 [18]	YOLOv3 416 + 3 cost-free tricks	YOLOv3 416 + 3 cost-free tricks + SPP	YOLOv3 416 + 3 cost-free tricks + ARSF
Airplane	95.8	90.4	86.5	95.4	94.6	94.3	**96.1**
Ship	78.4	60.9	65.4	87.1	88.5	**90.7**	88.7
Storage tank	85.5	79.8	**90.3**	70.9	75.2	78.0	84.5
Baseball diamond	89.9	89.9	89.6	**99.1**	98.7	97.7	**99.1**
Tennis court	**89.4**	82.6	85.1	73.2	78.6	72.3	81.1
Basketball court	82.9	80.6	80.4	81.2	**85.6**	**90.7**	83.0
Ground track field	1.2	98.3	78.2	96.2	99.5	**99.9**	98.3
Harbor	68.8	73.4	70.5	85.6	86.9	**89.3**	87.2
Bridge	70.2	**76.7**	68.2	60.6	63.5	77.0	76.4
Vehicle	**81.9**	52.1	74.2	56.1	63.7	59.4	63.2
Mean AP	74.4	78.4	78.8	80.5	83.5	84.9	**85.8**

convolution and 16-bit quantization technology (ARSF(lite+)), the parameters and Flops can be reduced by four times, and the memory read and write can be reduced by two times, which is a huge improvement.

Furthermore, we design an experiment to compare YOLOv3 and YOLOv3 variants using SPP, ARSF, ARSF (lite), and ARSF (lite +) on model size, parameters, Flops, memory read and write and mAP aspects. The results are shown in Fig.8. YOLOv3-ARSF can achieve the best performance. Compared with the baseline, YOLOv3-ARSF (lite+) decreases 50% model size and memory reads and writes, 26% parameters, 19% Flops, 9% inference time but only lost 5% of the mAP. Trading off efficiency and performance makes it more suitable for edge devices. However, YOLOv3-ARSF(lite) and YOLOv3-ARSF (lite+) did not compare with other lightweight technologies, such as model pruning and knowledge distillation. So for edge devices, using depthwise separable convolution and half-precision quantization may not be the best way.

V. CONCLUSION

In order to overcome the difficulties of scale diversity and small targets in remote sensing detection, a novel pyramid-structured network called ARSF is finely designed. This advanced structure balances the importance of low-leveled spatial information and high-leveled semantic information during the learning process. The adaptive learning behavior in ASF consolidates with robustness and increase flexibility when facing data heterogeneity. A shorter path in ESIP is provided, while a better performance is established. Experiment results show that the proposed detecting methods succeed in a wide range of object sizes, and the best performance of 85.8 mAP is achieved in the NWPU-VHR10 dataset. Besides, two lightweight versions, ARSF (Lite) and ARSF (lite+) provide a technical basis for the realization of online remote sensing object detection on the satellite in the future.

REFERENCES

[1] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," 2018, *arXiv:1805.09512*. [Online]. Available: http://arxiv.org/abs/1805.09512

[2] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, and J. Sun, "ThunderNet: Towards real-time generic object detection on mobile devices," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul South Korea, Oct. 2019, pp. 6717–6726, doi: 10.1109/ICCV.2019.00682.

[3] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[4] S. Kim, H. Kook, J. Sun, M. Kang, and S. Ko, "Parallel feature pyramid network for object detection," in *Proc. ECCV* (Lecture Notes in Computer Science), vol. 11209, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Munich, Germany: Springer, Sep. 2018, pp. 239–256, doi: 10.1007/978-3-030-01228-1_15.

[5] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," 2019, *arXiv:1901.02446*. [Online]. Available: http://arxiv.org/abs/1901.02446

[6] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, *arXiv:1805.10180*. [Online]. Available: http://arxiv.org/abs/1805.10180

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2014.

[8] S. Chen, R. Zhan, and J. Zhang, "Geospatial object detection in remote sensing imagery based on multiscale single-shot detector with activated semantics," *Remote Sens.*, vol. 10, no. 6, p. 820, 2018.

[9] J. Liu, S. Yang, L. Tian, W. Guo, B. Zhou, J. Jia, and H. Ling, "Multi-component fusion network for small object detection in remote sensing images," *IEEE Access*, vol. 7, pp. 128339–128352, 2019, doi: 10.1109/ACCESS.2019.2939488.

[10] Y. Wang, Z. Dong, and Y. Zhu, "Multiscale block fusion object detection method for large-scale high-resolution remote sensing imagery," *IEEE Access*, vol. 7, pp. 99530–99539, 2019, doi: 10.1109/ACCESS.2019.2930092.

[11] W. Zhao, W. Ma, L. Jiao, P. Chen, S. Yang, and B. Hou, "Multi-scale image block-level F-CNN for remote sensing images object detection," *IEEE Access*, vol. 7, pp. 43607–43621, 2019, doi: 10.1109/ACCESS.2019.2908016.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

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

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. ECCV* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[18] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[19] J. Redmon and A. Farhadi, ''YOLO9000: Better, faster, stronger,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[20] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''Imagenet classification with deep convolutional neural networks,'' in *Proc. NIPS*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Lake Tahoe, NV, USA: Springer, Dec. 2012, pp. 1106–1114. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ''Going deeper with convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[24] J. Hu, L. Shen, and G. Sun, ''Squeeze-and-excitation networks,'' in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html

[25] X. Zhang, X. Zhou, M. Lin, and J. Sun, ''ShuffleNet: An extremely efficient convolutional neural network for mobile devices,'' 2017, *arXiv:1707.01083*. [Online]. Available: http://arxiv.org/abs/1707.01083

[26] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, ''ShuffleNet v2: Practical guidelines for efficient CNN architecture design,'' 2018, *arXiv:1807.11164*. [Online]. Available: http://arxiv.org/abs/1807.11164

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, ''MobileNetV2: Inverted residuals and linear bottlenecks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520.

[29] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, ''Searching for MobileNetV3,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul South Korea, Oct. 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[30] F. Chollet, ''Xception: Deep learning with depthwise separable convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[31] Z. Li and F. Zhou, ''FSSD: Feature fusion single shot multibox detector,'' 2017, *arXiv:1712.00960*. [Online]. Available: http://arxiv.org/abs/1712.00960

[32] M. Tan, R. Pang, and Q. V. Le, ''EfficientDet: Scalable and efficient object detection,'' 2019, *arXiv:1911.09070*. [Online]. Available: http://arxiv.org/abs/1911.09070

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ''Grad-CAM: Visual explanations from deep networks via gradient-based localization,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[34] G. Cheng, J. Han, P. Zhou, and L. Guo, ''Multi-class geospatial object detection and geographic image classification based on collection of part detectors,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 98, pp. 119–132, Dec. 2014.

[35] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, ''Soft-NMS—Improving object detection with one line of code,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5562–5570, doi: 10.1109/ICCV.2017.593.

[36] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, ''Bag of freebies for training object detection neural networks,'' 2019, *arXiv:1902.04103*. [Online]. Available: http://arxiv.org/abs/1902.04103

FANG QINGYUN was born in 1994. He received the bachelor's degree in communication engineering from the Nanjing University of Science and Technology, in 2017. He is currently pursuing the Ph.D. degree with the School of Aerospace Engineering, Tsinghua University. His current research interests include deep learning and object detection.

ZHANG LIN is currently a Senior Research Associate with the Intelligent Robotics and Autonomous System Laboratory, University of Cincinnati. His major research interests include reinforcement learning and application in robotics.

WANG ZHAOKUI (Member, IEEE) was born in Anhui, China, in 1978. He received the B.S. and Ph.D. degrees in space engineering from the National University of Defense Technology, Changsha, China, in 1999 and 2006, respectively. Since 2009, he has been working as an Associate Professor with Tsinghua University. He has authored or coauthored more than 100 journal articles and conference papers. He has authored one text book and coauthored two text books. He holds more than 20 patents in China and one patent in USA and one patent in Germany. His research aims at small satellite of scientific exploration and enabling technology of distributed space systems, such as dynamics and intelligent control for satellite cluster, inner-formation based gravity measurement satellite, human–robot collaborations, and multi-robots collaborations formations in space. He has been a Council Member of the Chinese Association of Automation (CAA) and a member of the American Institute of Aeronautics and Astronautics (AIAA) and the Space University Administrative Committee of International Astronautical Federation (IAF).