Spatial Attention Based Real-Time Object Detection Network for Internet of Things Devices

YONGXIN ZHANG1, (Member, IEEE), PENG ZHAO1, DEGUANG LI1, (Member, IEEE), AND KOSTROMITIN KONSTANTIN2

1School of Information Science, Luoyang Normal University, Luoyang 471022, China
2Department of Physics of Nanoscale Systems, South Ural State University, 454080 Chelyabinsk, Russia

Corresponding author: Yongxin Zhang (tabo126@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802162, and in part by the Scientific and Technological Project of Henan Province under Grant 192102210122 and Grant 202102210370.

ABSTRACT Target detection algorithms for Internet of things (IoT) devices often require both high real-time performance and low computational complexity. Real-time object detection network: You Only Look Once Version 3 (YOLOv3) makes full use of multi-scale features to detect objects by using feature pyramid network structure, and achieves good performance on the premise of guaranteeing fast detection speed. The feature pyramid network of YOLOv3 includes bottom-up feature extraction, top-down sampling and lateral connection of low-level detail features and high-level semantic features. But not all features are useful for object detection. In this article, a novel object detection network Spatial Attention based YOLOv3 (SA-YOLOv3) is proposed. The proposed method adds spatial attention network to the top-down sampling process. The spatial attention network calculates the feature weight matrix based on the up-sampling feature map. SA-YOLOv3 uses the feature weight matrix to filter low-level features and retain more valuable features. Finally, the selected low-level feature map and high-level feature map are concatenated together and feature maps with both spatial information and rich semantic information are obtained. The experimental results on PASCAL VOC2012 datasets and RSOD datasets show that SA-YOLOv3 outperforms YOLOv3.

INDEX TERMS Object detection, feature pyramid network, spatial attention, Internet of Things.

I. INTRODUCTION

Object detection is a key step for IoT devices to realize intelligent perception and recognition of objects and processes, such as autonomous driving [1], [2] and robot vision [3]. The early object detection methods require the following steps. Firstly, traversing the entire image with sliding window and marking the location of the object to be detected. Secondly, extracting features from the marked areas. In this regard, some handcrafted features such as Histogram of Oriented Gradient (HOG) [4], Haar-like [5] and Scale-invariant feature transform (SIFT) [6] are commonly used. Finally, judging whether there are objects in the windows corresponding to these features by trained classifier, and then classifying the object. However, the early methods produced many redundant windows, and the handcrafted features were not robust enough for variable backgrounds, which resulted low detection efficiency.

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

Krizhevsky proposed AlexNet [7] based on Convolutional Neural Network (CNN) in 2012. It proved that the features extracted by deep neural network are more abundant and accurate than the handcrafted features. After that, researchers committed to studying the object detection methods based on deep neural network. These methods are divided into two categories: the proposal-based approaches and the end-to-end-based approaches.

The proposal-based approaches include two stages. Firstly, a series of region proposals are generated from the images, and then the targets in the candidate regions are classified by using CNN. In [8] and [9], Selective Search (SS) algorithm is used to generate region proposals in the first stage. SS algorithm is faster and more accurate than traditional sliding window methods. However, the SS algorithm extracted about 2000 region proposals, which resulted huge calculation and low detection speed. To further improve the detection speed, RPN (Regional Proposal Network) [10] is applied to simplify the generation of region proposal in Faster R-CNN (Region-Convolutional Neural Network).
In the following years, several detection methods, including Region-based Fully Convolutional Networks (R-FCN) [11], Rotation Region Proposal Networks (RRPN) [12], and R-FCN-3000 [13] have been proposed. These methods have improved the accuracy of target detection in varying degrees. However, the generation of region proposals makes this kind of detection algorithm consume a lot of time and space, which affects the real-time performance of the algorithm.

Lin et al. [14] analyzed and pointed out the reason why the end-to-end method lagged behind the two-stage detection method. From then on, the end-to-end detection methods began to develop rapidly. In 2016, Redmon et al. [15] firstly proposed an end-to-end-based detection algorithm: you only look once (YOLO). This algorithm could simultaneously predict category and coordinates of objects in the image by a single convolutional network. The detection speed of YOLO is much higher than the proposal-based approaches, while it has poor performance on some smaller objects in the image. The following research results improve the detection effect of the end-to-end detection algorithm for small size targets. In [16], Single Shot Multi Box Detector (SSD) algorithm was proposed. This algorithm learned from the idea of anchors in Faster R-CNN, and fused the feature maps of different resolutions. Fu et al. [17] replaced the backbone VGG16 [18] in SSD with the deeper ResNet-101 [19] to improve the feature extraction capabilities of the backbone, and then added some additional deconvolution to extract richer context information. Shen et al. [20] designed a network structure without loading pre-trained weights. This structure learned from the idea of dense connection in DenseNet [21], and then applied the Stem Block structure to SSD algorithm, which enabled the network to improve the accuracy while reducing the model parameters. In [22], Redmon and Farhadi proposed YOLOv2. The accuracy and detection speed of YOLOv2 are both higher than YOLO. However, YOLOv2’s Darknet 19 basic network features extraction ability is not strong, and it does not make full use of the multi-scale features of the target, making the performance of YOLOv2 slightly inferior to that of R-CNN. Lin et al. [23] proposed Feature Pyramid Network (FPN) with a top-down and laterally connected. This structure fuses multi-layer feature information and makes predictions on feature maps of different scales, and improves detection performance. In 2018, Redmon et al. proposed YOLOv3 [24]. YOLOv3 introduced Darknet53, which has strong ability of feature extraction, as its backbone. The research results of FPN [23] prove that deep features contain more abundant semantic information and are suitable for object classification. Low-level features contain detail spatial structure details and are suitable for locating object boundaries. YOLOv3 drew on the idea of FPN and combined the high-level semantic features with the low-level detail features, improved the ability of classification and location on small objects. However, YOLOv3 fuses unselected low-level features directly with high-level features. In fact, not all the low-level detail features are beneficial to detection. In order to solve this problem, we proposed Spatial Attention based YOLOv3 (SA-YOLOv3) network. Our contributions in this work are as follows:

1. We proposed Spatial Attention based YOLOv3 (SA-YOLOv3) network. The proposed SA-YOLOv3 has more accurate detection accuracy and similar speed than the YOLOv3.

2. The spatial attention network is used to filter the low-level detail features, so as to retain the effective features and eliminate the redundant features.

3. We combine the filtered shallow detail features with deep semantic features, and make full use of the advantages of the two features and achieve better detection results.

The remainder of this article is organized as follows: Section 2, we give a brief review of spatial attention network and YOLOv3 network; Spatial Attention based YOLOv3 network is proposed in Section 3; Section 4 proves the effectiveness of SA-YOLOv3; Section 5 is the conclusions of this article.

II. RELATED WORKS

A. SPATIAL ATTENTION NETWORK

Zhao et al. [25] proposed a spatial attention network (SA) and used it to calculate the saliency of pictures. SA can better locate the boundary of the object in the picture. The structure of SA network is shown in Fig. 1. The network is divided into two branches. The inputs of the two branches are the high-level semantic feature map (such as G in Fig. 1, where H, W, and C represent the height, width, and number of channels of the feature map, respectively) and the low-level detail feature map (see L in Fig. 1). The upper branch called location network in Fig. 1 is the core of SA network. Firstly, location network gets spatial features from high-level semantic feature map using asymmetric convolution structure 1 × k and k × 1, and then outputs two feature maps of U and U’ in Fig. 1. The number of channels of U and U’ is half of the original input, that is C/2 and C’/2. After that, location network reduces the channel dimension of U and U’ to 1 using k × 1 and 1 × k convolution, and the feature maps V and V’ are obtained. Finally, location network fuses V with V’ by feature map elements point-to-point addition and sigmoid operation to produce a weight matrix Z of size W × H. The other branch multiplies the low-level detail feature map L by the weight matrix Z extracted from the location network to generate the attention map Y of the spatial domain. The function of spatial attention network is to select the features in the low-level detail feature map L by using the weight matrix Z.

B. YOLOv3 NETWORK

YOLOv3 network consists of two parts, namely bottom-up feature extraction network and top-down multi-scale detection network. The former is responsible for extracting low-level detail features of the input image. The latter extracts high-level semantic features from the low-level detail features. Then these high-level features are concatenated with the corresponding low-level features to detect objects in
Spatial attention based real-time object detection network for IoT devices

**FIGURE 1.** Spatial attention network structure.

**FIGURE 2.** YOLOv3 network structure.

As shown in Fig. 2, Darknet53 with orange dotted frame on the left side is used as backbone in feature extraction network. The depth of Darknet53 is 75 layers and it is formed by stacking 53 convolution layers and 22 shortcut connection layers. The 75-layer Darknet53 network can be divided into a single DBL module and five residual blocks including res1, res2, res8, res8, and res4. DBL is a basic component of YOLOv3. DBL consists of convolution layer, Batch Normalization (BN) operation and Leaky ReLU. Resn indicates that each residual block contains n residual units. Each residual unit consists of two DBL modules connected in series. For the convenience of analysis, we divide the Darknet53 into three stages. The first stage is the blue box part in Fig. 2. In the first stage, Darknet53 uses three blue residual blocks (including res1, res2, and res8 in Fig. 2) to extract the features of the input image with the size of $416 \times 416 \times 3$, and gets the feature graph C1 with the size of $52 \times 52 \times 256$. In the second stage, Darknet53 uses the res8 module of the purple box in Fig. 2 to downsample C1 and get the feature map B1 with the size of $26 \times 26 \times 512$. In the third stage, feature map A1 is generated by res4 block with orange box in Darknet53. The size of A1 is $13 \times 13 \times 1024$.

In Fig. 2, the dotted line box (red) on the right is a top-down multi-scale detection network. This network is mainly used to predict the output feature maps of three different scales. The prediction process is as follows.

The output feature map A1 of the Darknet53 generates feature map A2 after passing through the DBL module.
in the multi-scale detection network. The size of $A_2$ is $13 \times 13 \times 512$. Then, the DBL and convolution operations in the predict1 module (blue dotted boxes in Fig. 2) change the feature graph $A_2$ into the output1 of the first scale (size $13 \times 13 \times 75$).

The block1 module (the first green dotted line frame in Fig. 2) performs DBL and up-sample operation on feature map $A_2$, and connate it with feature map $B_1$. The concatenated feature map is then processed through the second DBL $\times 5$ module in the multi-scale detection network, and the feature map $B_2$ is obtained, which is $26 \times 26 \times 256$ in size. The second scale output output2 (size $26 \times 26 \times 75$) is obtained via the predict2 module (whose operation is the same as predict1).

The process of obtaining feature map $C_2$ and output3 of the third scale is the same as that of $B_2$ and output2.

The outputs of the above three scales (Output1, Output2, and Output3) are used to detect large, moderate, and small objects in the input image, respectively.

III. SPATIAL ATTENTION BASED YOLOv3

A. THE LOCATION OF SPATIAL ATTENTION NETWORK IN SA-YOLOv3

We use a simplified structure of YOLOv3 network to illustrate the location of the SA network in SA-YOLOv3. As shown in Fig. 3, we simplify the network structure of YOLOv3 to the form of FPN [22]. The left half of Fig. 3 is the bottom-up feature extraction process. Each layer in the left represents the output feature map of each stage of Darknet53 network in Fig. 2. Input corresponds to the input of YOLOv3, that is, the input with yellow box of the Darknet53 network in Fig. 2. $C_1$ corresponds the output feature map of the first stage (blue box in Fig. 2, namely DBL module, res1 module, res2 module and res8 module) of Darknet53 network. $B_1$ corresponds the output feature map of the second stage (purple box in Fig. 2, namely res8 module) of Darknet53 network. $A_1$ corresponds to the output feature map of the third stage (orange box in Fig. 2, namely res4 module) of Darknet53 network.

The right half of Fig. 3 shows the top-down up-sampling process. In this process, $A_2$, $B_2$, and $C_2$ layers correspond the output feature maps of three DBL $\times 5$ modules in the multi-scale detection network in Fig. 2, respectively. The two blocks with the green dotted line correspond to the two blocks in Fig. 2, respectively. In Fig. 3, Predict1, Predict2, and Predict3 correspond to the predict1, predict2, and predict3 modules in the multi-scale detection network in Fig. 2, respectively.

The three lateral connections in Fig. 3 combine the low-level detail features with the up-sampled features. To remove the redundant features in the low-level detail
feature maps, we introduce SA network [24] to the two blocks in Fig. 3. We call the block after joining SA as SA-block. The proposed SA-block can select the low-level detail features, and then enable high-level semantic features in the detection network to be combined with more efficient detail features.

**B. SA-BLOCK**

The structure of SA-block is shown in Fig. 4. The high-level semantic feature map (A₂ or B₂) from the upper layer is processed through two DBL modules and an up-sample operation to serve as input to the SA network (red solid wireframe in Fig. 4). The SA obtains the weight matrix W by passing the input feature map through the location network. The location network obtains spatial features from the semantic feature map, using the asymmetric convolution structure \(1 \times k\) and \(k \times 1\). After that, the location network reduces the channel dimension of the spatial features to 1 dimension using the \(k \times 1\) and \(1 \times k\) convolution structures, respectively. Finally, the location network fuses the two branches’ output by feature map elements point-to-point addition to get the weight matrix \(W\). The weight matrix \(W\) is multiplied by the corresponding detail feature map (B₁ or C₁ in Fig. 3) to get the filtered low-level feature map \(SA_{out}\), that is, the output of the SA network. Finally, we fuse the high-level semantic feature
map with the selected feature map \( SA_{\text{out}} \). The purpose of doing this is to combine high-level semantic features with more effective detail features.

C. ARCHITECTURE OF SA-YOLOv3

The proposed SA-YOLOv3 network structure is shown in Fig. 5. The location of the SA-block in the SA-YOLOv3 network is shown by the red boxes of SA1 and SA2 in Fig. 5.

In SA-block1, the feature map A2 is passed through two DBL modules (where the second DBL module is used for increasing channel dimensions to get richer features) and an up-sample operation to serve as the input of the SA1 network, and its size is \( 26 \times 26 \times 512 \). The SA1 network is divided into two branches. The left branch uses conv1 operation with a kernel of \( 1 \times k \) (\( k = 3 \)) and conv2 operation with a kernel of \( k \times 1 \), and then obtains a feature map of size \( 26 \times 26 \times 1 \). The right branch uses the conv2 operation and the conv1 operation in sequence, and also obtains a feature map of size \( 26 \times 26 \times 1 \). The weight matrix \( W1 \) is obtained by adding and activating (by sigmoid function) the output of the two branches. The weight matrix \( W1 \) is multiplied by the low-level detail feature map \( B1 \) in Darknet 53, and then the map \( SA1_{\text{out}} \) (size \( 26 \times 26 \times 512 \)) is obtained as the output of SA1 network. Finally, we concatenated \( SA1_{\text{out}} \) and the high-level semantic feature map together.

In SA-block2, the feature map B2 is passed through two DBL modules and an up-sample operation to serve as the input of the SA2 network, and its size is \( 52 \times 52 \times 256 \). The output of SA2 is the weight matrix \( W2 \). Then \( W2 \) is multiplied by the low-level detail feature map \( C1 \) and the output feature map \( SA2_{\text{out}} \) (size \( 52 \times 52 \times 256 \)) of the SA2 network is obtained. Finally, the \( SA2_{\text{out}} \) and the high-level semantic feature map are concatenated and generate the output feature map \( C2 \) of the SA-block2 module.

IV. EXPERIMENTS

In this subsection, we carry out experiments on PASCAL VOC2012 [26] datasets and RSOD [27], [28] datasets to evaluate effectiveness of the proposed SA-YOLOv3 network. The PASCAL VOC2012 datasets includes 17,125 images with 20 categories, including airplanes, bicycles, birds and so on. Fig. 6(a) shows some sample images of PASCAL VOC2012 datasets. The RSOD data set includes 936 images with 4 categories, including aircraft, oiltank, overpass and playground. Fig. 6(b) shows some sample images of the RSOD datasets.

The training parameters of the SA-YOLOv3 network were set as follows. The SGD optimizer was adopted to update the weights of the network, momentum: 0.9, weight decay: 0.0005, the initial learning rate was 0.001, and batch size: 2. We set the size of the pictures in PASCAL VOC2012 datasets to \( 416 \times 416 \), and the size of the pictures in RSOD datasets to \( 608 \times 608 \). In the RSOD datasets, the size of input image is set to \( 608 \times 608 \). Our experiment was performed on Ubuntu 18.04LTS with an NVIDIA GeForce GTX 1050Ti GPU.

In the experiment, we use mIoU (mean Intersection-over-Union), mAP (mean Average Precision) [25] and fps (frames per second) to evaluate performance of network. The IoU value is the ratio of the bounding box of the object marked by the datasets to the bounding box of the object predicted by the network. The mIoU is the average of all IoUs \( (IoU > IoU_{\text{thresh}}, IoU_{\text{thresh}} = 0.5) \) on the premise that the network correctly predicts the object category. The mAP is the average of the average precision values for all categories. The mAP can be computed by Eq. (1) and (2).

\[
AveP = \sum_{r=0}^{1} \left( r_{n+1} - r_n \right) \cdot p_{\text{interp}} (r_{n+1}) \cdot \max_{r'} \left( p_{\text{interp}} (r_{n+1}) \right)
\]

\[
\text{mAP} = \frac{\sum_{q=1}^{Q} AveP (q)}{Q}
\]

The average precision can be obtained by (1). Where \( p = \text{Object} \cap \text{Detected} / \text{Detected} \) which represents the precision, is the number of correct results identified by the network. Where \( r = \text{Object} \cap \text{Detected} / \text{Object} \) which represents the recall, is the proportion of the number correctly recognized by the network for each category. The \( p_{\text{interp}} (r_{n+1}) \) is the maximum precision of \( r \) greater than or equal to \( r_{n+1} \). The mAP can be obtained by (2). Where \( Q \) is the total number of categories. Fps is the detection rate of network (frames per second).

| Method       | mAP (%) | mIoU  | Speed(fps) |
|--------------|---------|-------|------------|
| YOLOv3       | 44.9    | 0.77  | 16.9       |
| SA-YOLOv3    | 46.4    | 0.784 | 17         |

TABLE 1. Results of YOLOv3 and SA-YOLOv3 on pascal VOC2012 datasets.
TABLE 2. The mAP values of each category of YOLOv3 and SA-YOLOv3 on pascal VOC2012 datasets.

|        | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow |
|--------|------|------|------|------|--------|-----|-----|-----|-------|-----|
| YOLOv3 | 68.2 | 38.7 | 52.8 | 30.7 | 21.7   | 64.5| 42.5| 70.8| 24.8  | 43  |
| SA-YOLOv3 | 70.7 | 43   | 55.3 | 33.3 | 25.2   | 66.1| 48.3| 71.4| 21.4  | 37  |

FIGURE 7. Detection results of SA-YOLOv3 on PASCAL VOC2012 test set.

A. EXPERIMENTS ON PASCAL VOC2012 DATASETS

In the experiment, we randomly selected 60% of the images in PASCAL VOC2012 datasets as training set (including 10275 images), and the remaining 40% as test set (including 6850 images). Tables 1 and 2 show the results of YOLOv3 and SA-YOLOv3 on VOC2012 test set. Fig. 7 shows detection results of SA-YOLOv3 network on PASCAL VOC2012 test set.

From the results shown in Tables 1 and 2, we can make the following conclusions:

1. The mAP value of SA-YOLOv3 is 1.5% higher than that of YOLOv3 while maintaining the same speed as YOLOv3. At the same time, the mIoU value of SA-YOLOv3 is also 1.4% higher than that of YOLOv3.

2. The accuracy of SA-YOLOv3 is slightly lower than that of YOLOv3 in chair, cow, horse and train categories. However, in other categories, the mAP value of SA-YOLOv3 is the same as that of YOLOv3, and even has a substantial increase (such as bike, bus, car, and tv categories).

The results of the above two experiments indicate that the filtered detail features after spatial attention network can better locate the boundary of the object in image.

B. EXPERIMENTS ON RSOD DATASETS

Target detection in remote sensing images [26], [27] is also one of the important applications of object detection. We use RSOD datasets to evaluate the effectiveness of SA-YOLOv3 in remote sensing image target detection task.

TABLE 3. Results of YOLOv3 and SA-YOLOv3 on RSOD datasets.

|        | mAP (%) | mIoU | Speed(fps) |
|--------|---------|------|------------|
| YOLOv3 | 89.7    | 0.794| 10.9       |
| SA-YOLOv3 | 91.1 | 0.825| 7.5        |

TABLE 4. The mAP values of each category of YOLOv3 and SA-YOLOv3 on RSOD datasets.

|        | aircraft | oiltank | overpass | playground |
|--------|----------|---------|----------|------------|
| YOLOv3 | 88.6     | 94.5    | 75.9     | 99.9       |
| SA-YOLOv3 | 88.5 | 95.3    | 81.4     | 99.3       |

In the experiment, we used 60% of RSOD datasets as training set (including 655 pictures), and the remaining 40% as test set (including 281 pictures). The experimental results are shown in Tables 3 and 4. Fig. 8 shows detection results of SA-YOLOv3 network on RSOD test set.

From the results shown in Tables 3 and 4, we can get the following conclusions:

1. Although the fps of SA-YOLOv3 is slightly lower than that of YOLOv3, it still has real-time detection capability. The mAP value of SA-YOLOv3 is 1.4% higher than that of YOLOv3. At the same time, the mIoU value of SA-YOLOv3 is also increased by 3.1% compared to YOLOv3.

2. The performance of SA-YOLOv3 in the two categories of aircraft and playground is similar to that of YOLOv3, but the mAP value in the two categories of oiltank and overpass is significantly improved.
The results of the above two experiments indicate that the filtered detail features after spatial attention network can better locate the boundary of the object in image.

V. CONCLUSION

In this article, we proposed Spatial Attention based YOLOv3 network (SA-YOLOv3). The proposed SA-YOLOv3 network uses spatial attention network to filter the low-level detail features in the top-down up-sampling process, retaining more valuable detail features. Then the filtered detail features are concatenated with the high-level semantic features to obtain the features of both spatial information and rich semantic information. Experiment results show that the SA-YOLOv3 outperforms YOLOv3 on the premise of ensuring similar speed to YOLOv3.

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their comments and suggestions. They would also like to express their thanks to Li Bin for providing the codes and source images.

REFERENCES

[1] Y. Zhou and O. Tuzel, “VoxelNet: End-to-end learning for point cloud based 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4490–4499.

[2] B. Li, “3D fully convolutional network for vehicle detection in point cloud,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Sep. 2017, pp. 1513–1518.

[3] C.-J. Liang, K. M. Lundeen, W. McGee, C. C. Menassa, S. Lee, and V. R. Kamat, “A vision-based marker-less pose estimation system for articulated construction robots,” Autom. Construct., vol. 104, pp. 80–94, Aug. 2019.

[4] X. Wang, T. X. Han, and S. Yan, “An HOG-LBP human detector with partial occlusion handling,” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 32–39.

[5] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Dec. 2011, pp. 511–518.

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

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2012, pp. 1097–1105.

[8] 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., Jun. 2014, pp. 580–587.

[9] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. 27th IEEE Trans. Pattern Anal. Mach. Intell. (CVPR), Jul. 2017, pp. 91–99.

[11] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. 30th Conf. Neural Inf. Process. Syst. (NIPS), 2016, pp. 379–387.

[12] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, “Arbitrary-oriented scene text detection via rotation proposals,” IEEE Trans. Multimedia, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[13] B. Singh, H. Li, A. Sharma, and L. S. Davis, “R-FCN-3000 at 30fps: Decoupling detection and classification,” in Proc. 31st IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 1081–1090.

[14] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.

[15] 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), Jun. 2016, pp. 779–788.

[16] W. Liu, D. Anguelov, and D. Erhan, “SSD: Single shot MultiBox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37.

[17] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “DSSD: Deconvolutional single shot detector,” Jan. 2017, arXiv:1701.06659. [Online]. Available: https://arxiv.org/abs/1701.06659

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[20] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, “DSOD: Learning deeply supervised object detectors from scratch,” in Proc. 16th IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1919–1927.

[21] 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), Jul. 2017, pp. 2261–2269.

[22] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6517–6525.

FIGURE 8. Detection results of SA-YOLOv3 on RSOD test set.
Y. Zhang et al.: Spatial Attention Based Real-Time Object Detection Network for IoT Devices

[23] 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), Jul. 2017, pp. 936–944.

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

[25] T. Zhao and X. Wu, “Pyramid feature attention network for saliency detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3085–3094.

[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, Jun. 2010.

[27] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, “Accurate object localization in remote sensing images based on convolutional neural networks,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 5, pp. 2486–2498, May 2017.

[28] Z. Xiao, Q. Liu, G. Tang, and X. Zhai, “Elliptic Fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images,” Int. J. Remote Sens., vol. 36, no. 2, pp. 618–644, Jan. 2015.

PENG ZHAO received the master’s degree in computer applied technology from Northwestern Polytechnical University, Xi’an, China, in 2006. She is currently pursuing the Ph.D. degree in computer software and theory with Northwest University, Xi’an. She is also an Associate Professor with Luoyang Normal University, Luoyang, China. Her research interests include image processing and machine learning.

DEGUANG LI (Member, IEEE) received the B.S. degree in computer science from PLA Information Engineering University, in 2010, the M.S. degree in computer science from Northeastern University, in 2012, and the Ph.D. degree from Sichuan University, in 2017. He is currently a Lecturer with the School of Information Science, Luoyang Normal University. His research interests include green computing and mobile computing. He is a member of CCF.

KOSTROMITIN KONSTANTIN received the bachelor’s and master’s degrees in physics with a specialization in chair physics of condensed matter and the Ph.D. degree from Chelyabinsk State University, in 2008, 2010, and 2013, respectively.

YONGXIN ZHANG (Member, IEEE) received the B.E. degree in computer science and technology from Henan Polytechnic University, Jiaozuo, China, in 2003, and the M.S. and Ph.D. degrees in computer software and theory from Northwest University, Xi’an, China, in 2009 and 2014, respectively. He is currently an Associate Professor with Luoyang Normal University, Luoyang, China. His main research interest includes intelligent information processing.