2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

QuerySOD: A Small Object Detection Algorithm Based on Sparse Convolutional Network and Query Mechanism

Zhengcai Cao, *Senior Member*, *IEEE*, Junnian Li, Jie Niu, and MengChu Zhou, *Fellow*, *IEEE*

***Abstract*— Although remarkable advances have been achieved in generic object detection, small object detection (SOD) remains challenging owing to small objects' information loss and noisy representation caused by their non-uniform distribution. Their limited width and height, scale variations, and redundant computation make SOD hard. To overcome them, this work proposes a new SOD method based on sparse convolutional network (SCNet) and Query Mechanism called QuerySOD. First, an extended feature pyramid network is constructed for extracting feature maps of small objects with more regional details. Then, a Sparse Head is neatly designed by using SCNet for accelerating the interfering speed and obtaining weights of each layer. After that, a Query Mechanism is innovatively introduced for harvesting the benefit of sparse value feature maps from the Sparse Head. QuerySOD is evaluated on public benchmarks including COCO and VisDrone. Finally, we apply it on 'Jinghai' unmanned survey vehicles and receive excellent SOD performance from this real-world application.**

***Index Terms*— Small object detection, extended feature pyramid network, sparse convolutional network, Query Mechanism.**

I. INTRODUCTION

With the rapid advances of deep learning, great progress has been made in common object detection. However, due to the complexity of environments, small object detection (SOD) remains far from being satisfactory in both accuracy and efficiency. It faces two main challenges: 1) small objects typically have small scales (taking only less than 1% of the original image size); and 2) they are generally sparsely and non-uniformly distributed, especially over high-resolution images. When the length and width of objects are much lower than those of other objects (generic ones), recognizing such small objects becomes extremely difficult [1].

Compared with generic object detection [2]–[4], small objects are more difficult to detect due to their small size, scale variations, and few features. They occupy much fewer pixels than generic objects do, which leads to insufficient feature extraction and noisy representation for anchor-based object detectors [5]. They may appear in any position of an image, even at corners, which leads to less attention and can easily be neglected. Furthermore, because of the complicated backgrounds of environments, e.g., sky, it is difficult to distinguish them from the generic ones, and tell their boundaries. In summary, accurate and efficient SOD remains an open problem.

Generally, small object detectors can be roughly classified into six classes, i.e., feature-imitation, sample-oriented, scale-aware, focus-and-detect, context-modeling, and attention-based ones [6]. Context-modeling and attention-based detectors are mainly designed for remote sensing. This work focuses on the rest:

Feature-imitation detectors including similarity learning-based, super-resolution-based, and GAN-based ones [7]–[9], are able to combine semantic information from high-level feature maps and location details from low-level ones of small objects for their detection. However, the first two types of detectors must maintain the diversity of features for enhancing the semantic representations of small objects. The last one cannot generate real textures and artifacts, which negatively impacts SOD accuracy.

Sample-oriented detectors concentrate on increasing the number of small objects by using data augmentation or optimising assignment strategies to generate more samples for training [10]–[12]. Nevertheless, they exhibit inconsistent performance due to insufficient positive samples. Objects' small size and low-quality are main factors influencing their performance.

Focus-and-detect detectors can reduce the computation cost by filtering out the regions which do not include small objects when processing high-resolution pictures [13]–[15]. All of them face the question: *where to focus*? They either increase additional annotations or use auxiliary frameworks, thereby making the end-to-end optimization of SOD highly complicated.

Scale-aware detectors adopt multi-branch frameworks for specialized training and combining the hierarchical features for improving the representations of small objects [16]–[18]. However, when processing small objects at different layers, they sometimes suffer from insufficient information of each convolutional layer. Furthermore, the information flow of the networks in existing scale-aware detectors does not contain accurate representations of small objects, which inevitably influences SOD performance.

Based on our literature review, compared to commonly used SOD detectors constructed by using feature pyramid networks (FPN) [16]–[18], we identify two key problems: 1) the computation on low-resolution feature maps is not

This work is supported in part by the National Natural Science Foundation of China under Grant (92148202, 52175002), the Beijing Natural Science Foundation (L223019, 3242011), and FDCT under Grant No. 0047/2021/A1. (Corresponding author: Zhengcai Cao and MengChu Zhou)

Zhengcai Cao is with State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, 150080, China. (caozc@hit.edu.cn)

Junnian Li and Jie Niu are with College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, 100029, China. (junnian_L@163.com and niujie2988@163.com)

MengChu Zhou is with Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA. (mengchu@gmail.com)

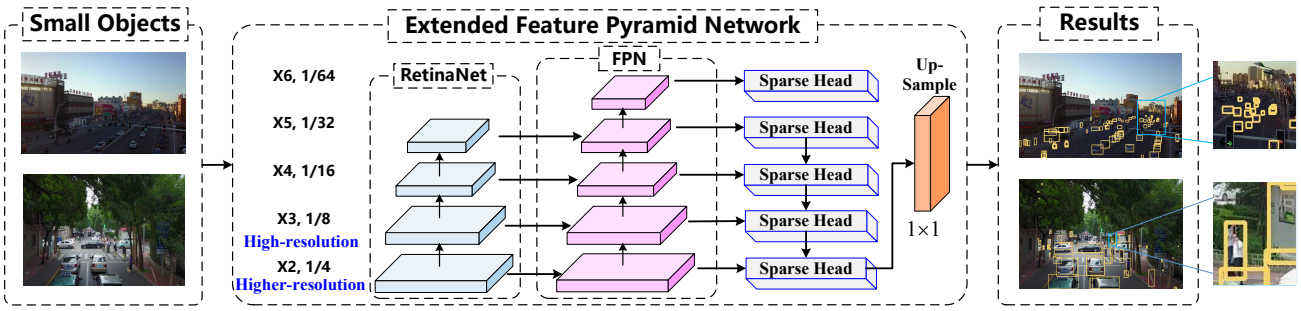

Fig. 1. The overall framework of our proposed QuerySOD.

always necessary due to the non-uniform distribution of small objects; and 2) small objects cannot be detected in low-resolution feature maps due to the highly structured feature pyramids.

To tackle these two problems, we propose a Scale-aware SOD detector based on sparse convolutional network (SCNet) and Query Mechanism, called QuerySOD, to improve SOD accuracy and speed. We intend to make the following novel contributions to the field of SOD:

1) An Extended Feature Pyramid Network (EFPN) is constructed. High and low-resolution features map of small objects are generated for classification and regression.

2) A Sparse Head is newly proposed to predict the accurate locations and associated scales of small objects. The context information of small objects is preserved by using SCNet.

3) A Query Mechanism is innovatively designed for generating the sparse value feature maps by applying the rough locations of small objects.

In Section II, our overall framework is introduced and EFPN, Sparse Head, Query Mechanism, and overall training process are described. Experimental results are reported in Section III. Conclusions are made in Section IV.

II. Proposed Methods

Small objects have limited width and height, which increases the difficulties of their accurate detection. In this work, a novel SOD method, called QuerySOD, is proposed. Its overall framework is shown in Fig. 1. It consists of EFPN, Sparse Head, and Query Mechanism.

A. EFPN

EFPN is constructed for SOD. It consists of RetinaNet as a backbone network, FPN module [19] and several newly designed Sparse Heads. A small object image is first fed into the backbone network. Then a variety of low and high-resolution feature maps of small objects are generated from the FPN module. From layer X_5 in Fig. 1, each FPN layer obtains a set of rough locations from its previous layer. The bounding boxes containing corresponding scales and accurate positions of small objects are forecasted by using Sparse Heads on high-resolution feature maps. An extra 1×1 convolutional layer is added at last for up-sampling low-resolution features to the same size, which increases the possibility of recognizing small objects.

RetinaNet [20], which includes an FPN backbone network and two detection subnets, is a common one-stage anchor-based object detector. The FPN backbone network in RetinaNet generates rich and multi-scale features of small objects, which detects these objects at different scales. The two detection subnets are the classification subnet and the box regression one, respectively. The former consists of four 3×3 convolutional layers with C filters and a ReLU module, followed by a 3×3 convolutional layer and sigmoid activations for final predictions. In the latter, a small FCN [21] is attached to each pyramid level for regressing the offsets between a ground-truth object and its corresponding bounding box.

We define the size of an input small object picture as $H\times W$. Then the size of FPN features is computed as

$$\mathbb{X}=\{X_l\in\mathbb{R}^{H^{'}\times W^{'}\times C}\} \tag{1}$$

where l represents the level of the FPN, and $(H^{'},W^{'})$ is set as $(\lfloor\frac{H}{2^l}\rfloor,\lfloor\frac{W}{2^l}\rfloor)$ in this work.

B. Sparse Head

Small objects can be classified by adequately applying high-resolution feature maps in EFPN. In this work, we added a higher-resolution X_2 in original RetinaNet for increasing the number of small objects during training. Nevertheless, this may lead to a new problem: high-resolution feature maps at X_2 and X_3 incur high computation cost, which is undesired. The dense computation mechanism in RetinaNet is able to handle this difficulty. Based on this observation, we design a Sparse Head to mitigate the information loss and reduce the computation cost of X_2 and X_3. Fig. 2 shows its overall structure.

In the Sparse Head, inspired by [18], a Query Mechanism is newly introduced. A coarse feature map X_l with stride 2^l which only activates the rough locations is constructed. It generates a heatmap $Y_l\in\mathbb{R}^{H^{'}\times W^{'}}$, which contains the rough location information of small objects. Y_l represents the possibility at location (i,j) which contains a small object. The small object's neighbouring context area is activated for extracting feature maps of small objects, thereby improving the detection accuracy. During the training process, the size of small objects on each level of FPN needs to be smaller than a threshold z_l, where z_l is the minimum anchor scale on

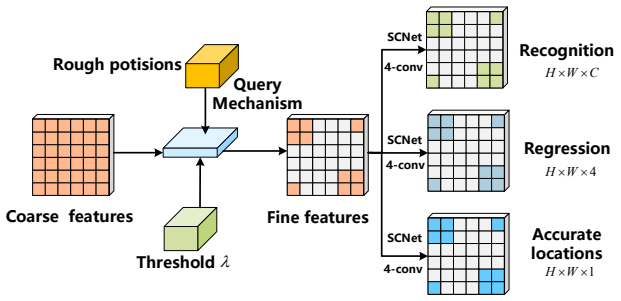

Fig. 2. The structure of the newly designed Sparse Head.

X_l. It can be pre-defined as the minimum regression range at layer l.

For each small object o, Sparse Heads' target feature maps are encoded by calculating the distance between its center point location (m_l, n_l) and other locations on X_l. Rough location information on X_{l-2} can be directly generated from X_{l-1}. By using the rough location of o, QuerySOD only needs to obtain its classification result during the prediction process on coarse feature maps to efficiently locate it, thus reducing the overall computational cost.

SCNet (u, v, f, s) [22] is constructed by using 4-conv for obtaining prediction results on layer $l-1$, which has u input feature maps, v output feature maps, filter size f, and stride s. Parameters f and s are normally odd integers. However, non-square filters can be generated, e.g., $f = 7 \times 1$ or $f = 1 \times 7$. Similar to regular convolution, the set of active sites is obtained by using SCNet. For acquiring more context information from high-resolution feature maps, the kernels of SCNet is defined to be of size 3×3. The size of the output image L_o is computed as:

$$L_o = (L_i - f + s)/s \tag{2}$$

where L_i represents the input image size of a small object. In this work, $s = 1$.

C. Query Mechanism

Small objects are typically distributed non-uniformly over high-resolution images, which makes SOD inefficient [23]. Based on this observation, within the Sparse head, a coarse-to-fine approach is innovatively proposed. At first, the rough locations are anticipated by using the Sparse Head. Then the corresponding accurate locations are intensively computed. As shown in Fig. 3, this step can be fulfilled by using a Query Mechanism: rough locations where small objects may exist in the low-resolution features are predicted as inputs, and sparse value feature maps are computed by using the high-resolution features as outputs.

Small objects' ground-truth bounding box is defined as $\bar{k}_l = (\bar{m}_l, \bar{n}_l, \bar{w}_l, \bar{h}_l)$, where $(\bar{m}_l, \bar{n}_l)$ represents the ground-truth center point position of each small object on X_l, and $(\bar{w}_l, \bar{h}_l)$ represents the width and height of the ground-truth bounding box. The minimum distance map D_l between each location (m, n) on X_l and the existing small objects' ground-truth

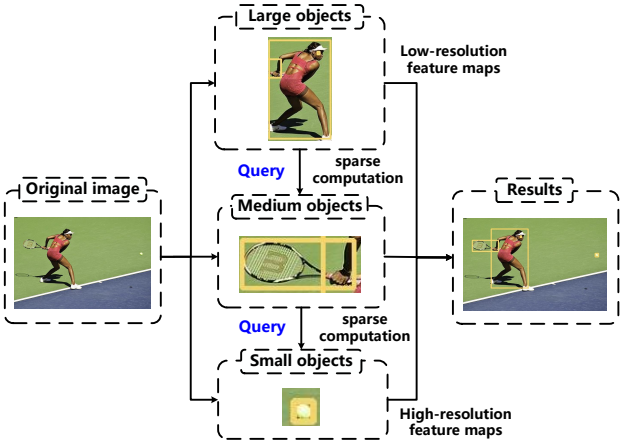

Fig. 3. The structure of the newly designed Query Mechanism.

center point location $(\bar{m}_l, \bar{n}_l)$ can be obtained as

$$D_l[m][n] = min \sum_l \sqrt{(m - \bar{m}_l)^2 + (n - \bar{n}_l)^2} \tag{3}$$

The ground-truth feature maps of the Query Mechanism can be obtained as

$$\bar{Q}_l[m][n] = \begin{cases} 1, & if\ D_l[m][n] < z_l \\ 0, & if\ D_l[m][n] \geq z_l \end{cases} \tag{4}$$

where z_l is pre-defined as the minimum regression range on X_l. When the distance $D_l[m][n]$ is bigger than z_l, the location is set to 0, and otherwise 1.

D. Training Process

In this work, FocalLoss [20] is applied to train our newly designed Sparse Head and Query Mechanism. During the whole training process, the locations of small objects whose predicted scores η are bigger than a pre-defined threshold λ are defined as a query value q_l. After that, q_{l-1} is mapped to its four neighboring features on X_{l-1} as accurate positions

$$\{\bar{a}_{l-1}\} = \{(2\bar{m}_l + i, 2\bar{n}_l + j), \forall \{i, j \in 0, 1\}\} \tag{5}$$

The accurate positions $\{\bar{a}_{l-1}\}$ on X_{l-1} are obtained to form the accurate position set $\{\bar{a}_{l-1}\}$. Then the Sparse Head processes $\{\bar{a}_{l-1}\}$ to recognize small objects and calculates next level's query score q_l. Value features on layer X_{l-1} applying $\{\bar{a}_{l-1}\}$ as indices are extracted as guidances to obtain a sparse tensor X_{l-1}.

The loss function of layer X_l in FPN is defined as:

$$\mathbb{L}_l(C_l,\ S_l,\ Q_l) = \mathbb{L}_f(C_l, \bar{C}_l) + \mathbb{L}_r(S_l, \bar{S}_l) + \mathbb{L}_f(Q_l, \bar{Q}_l) \tag{6}$$

where C_l, S_l, and Q_l define the output of the classification, the regression, and the Query score, while, $\bar{C}_l$, $\bar{S}_l$, and $\bar{Q}_l$ define their corresponding ground-truth features. $\mathbb{L}_r$ and $\mathbb{L}_f$ represent the bounding box regression loss and focal loss functions, respectively.

The overall loss of QuerySOD can be obtained as

$$\mathbb{L}_a = \sum_l \mathbb{L}_l \times \delta_l \tag{7}$$

where δ_l defines the weights of each layer's loss in FPN. In this work, the weights $\delta_l \in \{1, 3\}$ are linearly growing from 1 to 2.6, which can reasonable allocate the loss of each layer to make QuerySOD learn from both high and low-resolution feature maps.

E. Relationship with Previous Work

Different from QueryDet [18], our newly designed Sparse Head predicts the rough locations of small objects (as input of the Query Mechanism) in low-resolution feature maps and apply these locations as prior information to generate sparse value feature maps (as output of the Query Mechanism). Information of small objects is easily lost at high-resolution levels of FPN. This motivates us to fully apply such information. As a result, the final performance of SOD can be further improved. QueryDet only designs an auxiliary loss to generate recall predictions. In this work, both low-resolution and high-resolution feature maps are generated to extract semantic information of small objects for computing their accurate location coordinates, which leads to higher SOD performances than QueryDet. The experimental results demonstrate that QuerySOD is able to recognize small objects under dark scenarios.

III. EXPERIMENTS

A. Datasets

Public benchmarks COCO [24] and VisDrone [25] are adopted in this work:

COCO benchmark contains 118287 images for training, 5000 images for validating, and 40670 images for testing. 860001 objects from 80 classes are annotated for classification and regression. About 41.43% of objects are defined as small objects ($\leq 32 \times 32$ pixels) [10]. Only about 50% of the training images contain these small objects, which is obviously fewer than these images containing medium and large objects. This leads to a problem: there are fewer images with small objects for training. However, COCO is widely used because it is well annotated and has the reasonable evaluation metrics, which encourages each object to be located more accurately.

VisDrone benchmark is a large-scale dataset containing drone-captured images throughout 14 cities of China. Images in it are captured by using drones under multiple real-world urban/suburban scenarios with viewpoint variations and heavy occlusions. This dataset concentrates on providing high-quality images for 4 tasks, i.e., object detection given images, object detection given videos, visual object tracking, and multi-object tracking. There are 10209 images with 2000×1500 pixels in this dataset, including 6471, 548 and 3190 images for training, validating, and testing, respectively.

B. Hardware Devices

QuerySOD is trained on our PC with Intel(R) Xeon(R) Silver 4214R GPU @2.40GHz $\times$ 48, 4 NVIDIA GeForce GTX 3090 and 128 GB memory. It is implemented in Python 3.7, PyTorch 1.9.0, and Detectron2 0.5 [26]. Ubuntu 22.04.01 environment is constructed for performing all the experiments.

TABLE I

COMPARISON RESULTS ON COCO TEST-DEV [24]. THE BEST TWO RESULTS ARE HIGHLIGHTED IN RED AND BLUE FONTS. (3X) MEANS THAT THE NUMBER OF ITERATIONS IS SET AS 3 TIMES COMPARED TO ITS ORIGINAL VERSION.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
EfficientDet [28]	33.8	52.2	35.8	12.0	38.3	51.2
DETR-DC5$^+$ [29]	36.2	57.0	37.4	16.3	39.2	53.9
RefineDet [3]	36.4	57.5	39.5	16.6	39.9	51.4
FP-DETR-Lite [30]	37.9	57.5	41.1	21.7	40.6	50.7
RHF-Net [17]	37.7	59.8	40.1	19.9	42.9	51.5
QueryDet [18]	38.2	58.6	40.9	23.7	42.0	49.5
QuerySOD (1x)	35.3	54.8	37.9	20.5	38.2	45.6
QuerySOD (3x)	39.4	60.0	42.0	23.5	42.3	51.4

C. Implementation Details

We initially set *Learning rate* $= 1 \times 10^{-2}$ and *Batch size* $= 8$. The Query threshold λ as 0.25, which begins from layer X_4 in FPN. The backbone network RetinaNet is constructed based on ResNet-50 [27]. Default data augmentation methods in Detectron2, including Transform and Augmentation, are directly adopted during the training.

All batch normalization layers within the RetinaNet are frozen while training. During training, for each image within Visdrone, it is split into four patches which are non-overlapping and processed independently. QuerySOD is trained for 50000 iterations with Visdrone and initially *Learning rate* $= 1 \times 10^{-2}$. Then the learning rate is decayed at the 30000th and 40000th iterations by 10. Otherwise, QuerySOD is trained for 90000 iterations with COCO, by noting that COCO has more images for training.

D. Comparisons

Table I reports the comparison results between QuerySOD and several recent SOD detectors, e.g., Efficientdet [28], DETR-DC5$^+$ [29], RefineDet [3], FP-DETR-Lite [30], RHF-Net [17], and QueryDet [18] on COCO Test-Dev [24]. It can be observed that QuerySOD receives 39.4 AP, 60.0 AP_{50}, and 42 AP_{75}, which is obviously higher than its peers'. Although QuerySOD yields lower AP_L than its peers, it obtains the second highest AP_S and the highest AP_M, which demonstrates that by using low-level high-resolution feature maps and rough locations, QuerySOD is able to achieve outstanding detection performance in both generic and small objects.

From Table I, it can be found that the performance of QuerySOD is influenced by the set of total iterations. QuerySOD (1x) obtains 35.3 AP, 54.8 AP_{50}, and 20.5 AP_S, while, QuerySOD (3x) receives 39.4 AP, 60.0 AP_{50}, and 23.5 AP_S, which improves AP, AP_{50}, and AP_S by 4.1, 5.2, and 3.0 compared to QuerySOD (1x), respectively.

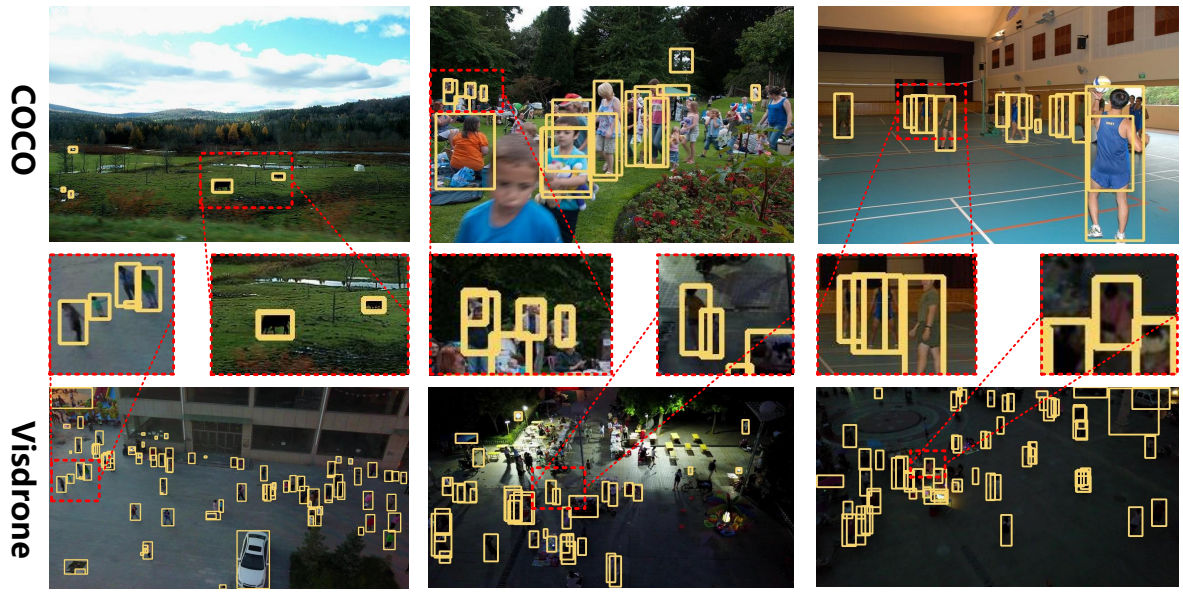

Fig. 4. Qualitative results of our method on COCO [24] and VisDrone [25]. All class labels are removed for better distinguishing the bounding boxes of small objects. The detection results with the red rectangle should be especially noted.

TABLE II

COMPARISON RESULTS ON VISDRONE2019-DET [25]. THE BEST TWO RESULTS ARE HIGHLIGHTED IN RED AND BLUE FONTS.

Method	Backbone	Input size	AP	AP_{50}	AP_{75}	FPS
ScaleKD [31]	ResNet-101	1920	29.4	49.3	30.0	-
HRDNet [16]	ResNeXt-101	3800	31.4	53.3	31.6	2.8
CDMNet [32]	ResNeXt-101	1920	31.9	52.9	33.2	-
PRDet [15]	ResNeXt-101	1920	32.0	53.9	33.2	7.9
GLSAN [14]	ResNet-50	600	32.5	55.8	33.0	1.3
QuerySOD	ResNeXt-101	1920	32.4	56.0	32.4	3.5

TABLE III

COMPARISON RESULTS ON COCO TEST-DEV [24]. (3X) MEANS THAT THE NUMBER OF ITERATIONS IS SET AS 3 TIMES COMPARED TO ITS ORIGINAL VERSION. QM REPRESENTS THE QUERY MECHANISM.

Method	QM	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS
RetinaNet	-	33.5	52.3	35.8	19.0	36.8	42.7	5.2
RetinaNet (3x)	-	34.3	54.6	36.6	20.3	38.5	43.5	5.2
QuerySOD	-	35.4	55.0	38.0	20.7	38.2	45.6	1.5
QuerySOD	✓	35.3	54.8	37.9	20.5	38.2	45.6	10.1
QuerySOD (3x)	-	39.4	60.1	42.1	23.5	42.4	51.5	1.0
QuerySOD (3x)	✓	39.4	60.0	42.0	23.5	42.3	51.4	6.8

Table II reports the comparison results of average precision (AP) and frame per second (FPS) between QuerySOD and several recent SOD detectors, e.g., ScaleKD [31], HRDNet [16], CDMNet [32], PRDet [15], and GLSAN [14] on VisDrone2019-DET [25]. It can be observed that QuerySOD receives the highest AP_{50}, and the second highest AP, which demonstrates its effectiveness. Although QuerySOD does not obtain the best score on AP_{75}, it takes the crown on the overall metric AP_{50}, demonstrating its superiority for SOD. It implies that accurate position predictions of small-scale objects is critical in the complicated scenario.

In Fig. 4, we visualize the QuerySOD's detection results on COCO [24] and VisDrone2019-DET [25]. It can be observed that QuerySOD can recognize almost every object, especially the small ones. The main reason is that our newly designed Sparse Head can accurately locate the small objects by using their corresponding rough ones. The Query Mechanism can provide sufficient context information. With the help of it, high-resolution features can be successfully incorporated for SOD, which significantly improves the SOD performance of QuerySOD.

E. Ablation Study

Table III lists the comparison results of AP and FPS between the baseline RetinaNet and QuerySOD on COCO Test-Dev [24]. It can be observed that the baseline RetinaNet (3x) maintains 5.2 mean FPS, and obtains 34.3 overall AP and 20.3 AP_S. With the help of higher resolution feature maps than it, QuerySOD (3x) runs 6.8 mean FPS, and receives 39.4 AP and 23.5 AP_S, which improves AP and AP_S by 5.1 and 3.2 compared to the baseline RetinaNet (3x), respectively. The importance of using rough locations of small objects as guidances for obtaining accurate locations while performing SOD is well revealed.

Table IV shows the comparison results of AP and FPS between the baseline RetinaNet, GLSAN [14], and QuerySOD on Visdrone2019-DET [25]. It can be seen that GLSAN maintains 1.3 mean FPS, and obtains 32.5 overall AP. Meanwhile the baseline RetinaNet maintains 2.4 mean FPS, and obtains 31.4 overall AP. QuerySOD runs 3.5 mean FPS, and receives 32.4 AP, which improves the accuracy of different models at a range of 0.3%-2% compared to

TABLE IV
COMPARISON RESULTS ON VISDRONE2019-DET [25]. QM REPRESENTS THE QUERY MECHANISM.

Method	QM	Input size	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning-tri	Bus	Motor	*AP*	FPS
GLSAN [14]	-	600	24.2	17.3	13.7	49.7	31.5	24.5	17.5	11.2	40.2	24.1	32.5	1.3
RetinaNet	-	1920	31.5	21.6	17.3	59.2	38.0	28.6	22.1	11.2	47.9	29.0	31.4	2.4
QuerySOD	-	1920	33.6	23.3	17.9	60.6	38.1	28.3	22.4	12.8	50.1	29.8	32.5	0.8
QuerySOD	✓	1920	33.5	23.3	18.0	60.5	38.1	28.2	22.4	12.7	50.1	29.8	32.4	3.5

baseline RetinaNet. These results efficiently demonstrate that QuerySOD performs better on SOD tasks compared to its peer, which is mainly because the position predictions of QuerySOD becomes more accurate by using the newly designed coarse-to-fine Query Mechanism. As shown in Fig. 4, QuerySOD possesses capabilities for recognising small objects under nighttime scenarios, which makes it effective for real-world challenging scenarios.

As shown in Tables III-IV, with the help of the Query Mechanism, the inference speed of QuerySOD is significantly improved. Specifically, in Table III, the speed of QuerySOD and QuerySOD (3x) with Query Mechanism is improved to about 6.8 times for high-resolution detection compared to its original version. In Table IV, the speed of QuerySOD with the Query Mechanism is increased to about 4.4 times compared to its original version on average. Compared to the baseline RetinaNet, QuerySOD is able to gain more speed improvement, which demonstrates that it can be deployed on the embedded systems for real-time applications. The main reason is that accurate location computation on high-resolution X_2 and X_3 is partly carried out by using input rough location information. Compared with the baseline RetinaNet with higher-resolution X_2, total FLOPs is decreased by 1% with the help of the Query Mechanism.

F. Applications

QuerySOD has been successfully deployed on a 'Jinghai' unmanned survey vehicle (USV) [33]. The realized total length, width, and molded depth of USV are 4.50m, 2.50m, and 0.55m, respectively. The full loaded USV weights about 0.60t in the air and the designed draft of the hull is usually kept lower than its normal status. An Nvidia Jetson Xavier NX, a photoelectric sensor system, and an intelligent Wi-Fi camera are installed on this USV.

Because of the complexity of sea environments, it is very challenging for sailors to keep focusing on the screen of 'Jinghai' for a long time. Fig. 5. shows the qualitative results when 'Jinghai' is performing maritime patrolling after the proposed QuerySOD is successfully deployed. All these images are captured by using the photoelectric sensor system installed on this USV. From Fig. 5, it can be seen that with the help of QuerySOD, 'Jinghai' is able to recognize maritime small objects of different scales from 600m $\sim$ 700m, which validates that QuerySOD is ready to be widely used for inshore and offshore maritime patrolling. Furthermore, compared to baseline RetinaNet, QuerySOD can recognise small-scale USV more accurately.

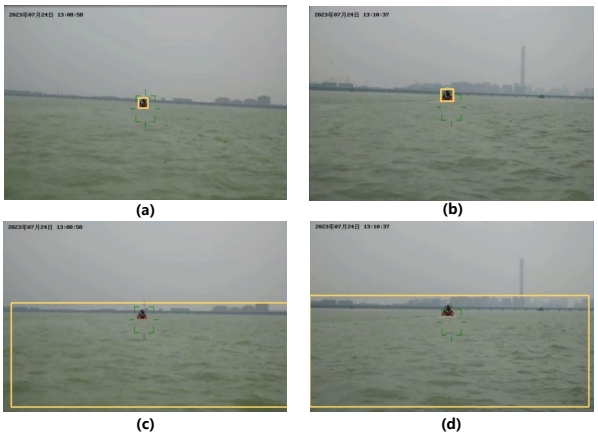

Fig. 5. Different scales of the same USV when 'Jinghai' is performing maritime patrolling. (a) and (b) are generated through QuerySOD, (c) and (d) are generated through baseline RetinaNet.

IV. CONCLUSIONS

This work introduces QuerySOD, a scale-aware SOD detector based on a sparse convolutional network and Query Mechanism. An EFPN is constructed for obtaining high and low-resolution features maps of small objects for classification and regression. For predicting the small objects' corresponding bounding boxes containing accurate scales and positions within the high-resolution feature maps, a Sparse Head is neatly designed. A Query Mechanism is newly introduced to obtain the accurate locations by effectively utilizing high and low-resolution features. Experimental results on COCO and Visdrone demonstrate that QuerySOD receives outstanding SOD performance, by especially reducing the computation amount and efficiently improving the detection accuracy of small objects. QuerySOD is successfully used on 'Jinghai' USV. It can be observed that this USV is able to recognize boats with limited width and height under real-world maritime scenarios, particularly in relatively high sea scale, which illustrates its excellent generalization ability to fulfill maritime SOD. Our next work includes extending QuerySOD to more challenging sea environments for detecting unseen maritime small objects, which can be used for inshore and offshore security examinations of ports, and islands, lakes, as well as other scenarios [34]–[44].

REFERENCES

[1] G. Chen *et al.*, "A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 936–953, 2022.

[2] Z. Huang *et al.*, "Feature map distillation of thin nets for low-resolution object recognition," *IEEE Transactions on Image Processing*, vol. 31, pp. 1364–1379, 2022.

[3] S. Zhang *et al.*, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4203–4212.

[4] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.

[5] S. Shrestha, S. Pathak, and E. K. Viegas, "Towards a robust adversarial patch attack against unmanned aerial vehicles object detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 3256–3263.

[6] G. Cheng *et al.*, "Towards large-scale small object detection: Survey and benchmarks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13 467–13 488, 2023.

[7] J. Li *et al.*, "Perceptual generative adversarial networks for small object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1222–1230.

[8] J. Noh *et al.*, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9725–9734.

[9] C. Deng *et al.*, "Extended feature pyramid network for small object detection," *IEEE Transactions on Multimedia*, vol. 24, pp. 1968–1979, 2021.

[10] M. Kisantal *et al.*, "Augmentation for small object detection," *arXiv preprint arXiv:1902.07296*, 2019.

[11] C. Xu, J. Wang, W. Yang, and L. Yu, "Dot distance for tiny object detection in aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1192–1201.

[12] C. Xu *et al.*, "Rfla: Gaussian receptive field based label assignment for tiny object detection," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 526–543.

[13] F. Ozge Unel, B. O. Ozkalayci, and C. Cigla, "The power of tiling for small object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 582–591.

[14] S. Deng *et al.*, "A global-local self-adaptive network for drone-view object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 1556–1569, 2020.

[15] J. Leng *et al.*, "Pareto refocusing for drone-view object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 3, pp. 1320–1334, 2022.

[16] Z. Liu, G. Gao, L. Sun, and Z. Fang, "Hrdnet: High-resolution detection network for small objects," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2021, pp. 1–6.

[17] P.-Y. Chen, J.-W. Hsieh, C.-Y. Wang, and H.-Y. M. Liao, "Recursive hybrid fusion pyramid network for real-time small object detection on embedded devices," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 402–403.

[18] C. Yang, Z. Huang, and N. Wang, "Querydet: Cascaded sparse query for accelerating high-resolution small object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 668–13 677.

[19] T.-Y. Lin *et al.*, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[22] B. Du, Y. Huang, J. Chen, and D. Huang, "Adaptive sparse convolutional networks with global context enhancement for faster object detection on drone images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 435–13 444.

[23] A. Meethal, E. Granger, and M. Pedersoli, "Cascaded zoom-in detector for high resolution aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2046–2055.

[24] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[25] P. Zhu *et al.*, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7380–7399, 2021.

[26] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," *https://github.com/facebookresearch/detectron2*, 2019.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 781–10 790.

[29] X. Dai *et al.*, "Dynamic detr: End-to-end object detection with dynamic attention," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2988–2997.

[30] W. Wang, Y. Cao, J. Zhang, and D. Tao, "Fp-detr: Detection transformer advanced by fully pre-training," in *Proceedings of the International Conference on Learning Representations*, 2022, pp. 1–14.

[31] Y. Zhu *et al.*, "Scalekd: Distilling scale-aware knowledge in small object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 723–19 733.

[32] C. Duan *et al.*, "Coarse-grained density map guided object detection in aerial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2789–2798.

[33] Y. Peng *et al.*, "Development of the USV jinghai-iand sea trials in the southern yellow sea," *Ocean Engineering*, vol. 131, pp. 186–196, 2017.

[34] Z. Cao, X. Xu, B. Hu, and M. Zhou, "Rapid detection of blind roads and crosswalks by using a lightweight semantic segmentation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6188–6197, 2020.

[35] Z. Cao *et al.*, "A multi-object tracking algorithm with center-based feature extraction and occlusion handling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4464–4473, 2022.

[36] J. Li, D. Zhang, M. Zhou, and Z. Cao, "A motion blur qr code identification algorithm based on feature extracting and improved adaptive thresholding," *Neurocomputing*, vol. 493, pp. 351–361, 2022.

[37] H. Mu *et al.*, "Dynamic obstacle avoidance system based on rapid instance segmentation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 4578–4592, 2023.

[38] Y. Sun, Z. Ma, M. Zhou, and Z. Cao, "A topological semantic mapping method based on text-based unsupervised image segmentation for assistive indoor navigation," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, p. 2531513, 2023.

[39] Y. Shi, J. Xia, M. Zhou, and Z. Cao, "A dual-feature-based adaptive shared transformer network for image captioning," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, p. 5009613, 2024.

[40] E. Spyrakos-Papastavridis and J. S. Dai, "Minimally model-based trajectory tracking and variable impedance control of flexible-joint robots," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 7, pp. 6031–6041, 2020.

[41] L. Huang *et al.*, "Multirobot cooperative patrolling strategy for moving objects," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 5, pp. 2995–3007, 2022.

[42] L. Huang, M. Zhou, and K. Hao, "Non-dominated immune-endocrine short feedback algorithm for multi-robot maritime patrolling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 362–373, 2019.

[43] J. Zhang *et al.*, "Pso-based sparse source location in large-scale environments with a uav swarm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5249–5258, 2023.

[44] L. Butera *et al.*, "Precise agriculture: effective deep learning strategies to detect pest insects," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 246–258, 2021.