International Journal of Applied Earth Observations and Geoinformation 104 (2021) 102548

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag

Sparse anchoring guided high-resolution capsule network for geospatial object detection from remote sensing imagery

Yongtao Yu [a,*], Jun Wang [a], Hao Qiang [a], Mingxin Jiang [a], E Tang [a], Changhui Yu [a], Yongjun Zhang [a], Jonathan Li [b]

[a] Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian, JS 223003, China
[b] Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L3G1, Canada

ARTICLE INFO

Keywords:
Object recognition
Sparse anchoring
Region proposal
Capsule attention
Capsule network
Remote sensing imagery

ABSTRACT

As the optical remote sensing techniques keep developing with a rapid pace, remote sensing images are positively considered in many fields. Accordingly, a great number of algorithms have been exploited for remote sensing image interpretation purposes. Thereinto, object recognition acts as an important ingredient to many applications. However, to achieve highly accurate object recognition is still challengeable caused by the orientation and size diversities, spatial distribution and density variations, shape and aspect ratio irregularities, occlusion and shadow impacts, as well as complex texture and surrounding environment changes. In this paper, a sparse anchoring guided high-resolution capsule network (SAHR-CapsNet) is proposed for geospatial object detection based on remote sensing images. First, formulated with the multibranch high-resolution capsule network architecture assisted by multiscale feature propagation and fusion, the SAHR-CapsNet can extract semantically strong and spatially accurate feature semantics at multiple scales. Second, integrated with the efficient capsule-based self-attention module, the SAHR-CapsNet functions promisingly to attend to target-specific spatial features and informative channel features. Finally, adopted with the capsule-based sparse anchoring network, the SAHR-CapsNet performs efficiently in generating a fixed number of lightweight, high-quality sparse region proposals. Quantitative assessments and comparative analyses on two challenging remote sensing image datasets demonstrate the applicability and effectiveness of the developed SAHR-CapsNet for geospatial object detection applications.

1. Introduction

Employing the bird-view surveying means, optical remote sensing imaging sensors have the superiorities of large perspectives, less ground condition restrictions, and convenience in data acquisition. They can rapidly and cost-effectively acquire high-quality, varying-resolution remote sensing images reflecting the details and changes of the land covers. Consequently, remote sensing images are widely and intensively used in many applications ranging from environmental monitoring (Rishikeshan and Ramesh, 2018), land use mapping (Xu and Somers, 2021), agricultural management (Sagan et al., 2021) to intelligent transportation systems (Lu et al., 2021). To date, intensive attentions have attracted to conduct intelligent interpretation of remote sensing images aiming at promoting the automation level, the processing efficiency, and the accuracy of the output products (Ma et al., 2019). Among the various researches, geospatial object detection is a hot topic and behaves as important prerequisite to many applications. In the literature, numerous algorithms and techniques have been developed for geospatial object detection tasks, especially the recent breakthroughs achieved by the deep learning models (Li et al., 2020). The output of geospatial object detection pipelines usually involves the accurate localization parameters and the correct object category labels.

Unlike the upright features of the objects exhibiting in ground-shooting images, geospatial objects in remote sensing images usually demonstrate top views with arbitrary orientations (Cheng and Han, 2016). A typical issue is that some geospatial objects exhibit severe incompleteness resulted in by the occlusions of nearby high-rise land covers. In addition, the illumination condition changes often generate different-level and different-area shadow covers on the geospatial objects. Furthermore, the size variations, shape irregularities, spatial distribution diversities, texture inconsistencies, and inter-category similarities of the geospatial objects are also common phenomena in the

* Corresponding author.
E-mail address: allennessy@hyit.edu.cn (Y. Yu).

https://doi.org/10.1016/j.jag.2021.102548
Received 12 July 2021; Received in revised form 10 September 2021; Accepted 13 September 2021

remote sensing images. In fact, to achieve highly accurate recognition of geospatial objects on a par with human-level identification qualities is still challengeable, especially for the small-size, irregular-shape, densely-distributed, and severely-contaminated objects. Therefore, exploiting advanced solutions to further promote the detection efficiency and accuracy is significantly necessary.

In this paper, a novel two-stage anchor-free architecture is presented to detect oriented geospatial objects based on remote sensing images. This architecture comprises a high-resolution capsule network for multiscale feature extraction, a sparse anchoring network for lightweight region proposal generation, and a multi-category classification network for object detection. On account of the formulation of the high-resolution capsule network as the feature extraction backbone, the integration of the efficient self-attention module as the feature promotion mechanism, and the development of the sparse anchoring network as the region proposal generation strategy, the proposed architecture behaves competitively to detect geospatial objects of varied sizes and orientations, different densities and distributions, diverse shapes and aspect ratios, and complex surface and environmental conditions. The main contributions are embodied in the following aspects: (1) A multi-branch high-resolution capsule network architecture functioned with multiscale feature propagation is designed to extract semantically strong and spatially accurate feature representations for well encoding geospatial objects of varying sizes. Compared with the high-resolution network (HRNet) (Sun et al., 2019; Wang et al., 2021b) made of scalar neurons, the proposed high-resolution capsule network with capsule primitives and modified multiscale feature fusion strategy performs better in high-quality entity-aware feature representation. (2) A capsule-form efficient self-attention module composed of spatial feature and channel feature attention units is designed to well attend to target-specific spatial features and informative channel features for further boosting the feature encoding quality and robustness. Specifically, the channel feature attention unit serves positively to emphasize geospatial objects of different sizes and the spatial feature attention unit functions excellently to suppress the impacts of the background features. (3) A capsule-based sparse anchoring network is designed to automatically produce a set of lightweight, high-quality sparse region proposals for highly efficient object recognition. Without the burdensome regression and selection of high-quality region proposals from the numerous candidates, the sparse anchoring guided implementation effectively accelerates the processing efficiency.

The rest of the paper has the following section organizations. Section 2 reviews the deep learning models for geospatial object recognition tasks. Section 3 details the implementation of the sparse anchoring guided high-resolution capsule network. Section 4 presents the experimental analyses and comparisons. Section 5 provides the conclusions.

2. Related works

2.1. One-stage models

The one-stage models are formulated with a single network architecture, which directly conduct object recognition and regression based on the extracted feature maps. Yao et al. (2021b) proposed a multiscale convolutional neural network (CNN) architecture to alleviate the detection accuracy degradation of small-size objects by exploiting multiscale features and contextual cues. Courtrai et al. (2020) combined the super-resolution technique with a generative adversarial network (GAN) to enlarge the small-size objects, thereby improving the object details. Lei et al. (2020) designed a region-enhanced CNN model directed with a saliency map aiming at enhancing the object region saliencies to achieve better detections. Shi et al. (2021) integrated a geometric transform module and a global contextual feature fusion module into a one-stage model to, respectively, capture the rotation and flip transformations and boosted the feature semantics via a spatial attention mechanism. Hou et al. (2021) proposed a self-adaptive aspect ratio anchor formulation to depict the orientations of objects. In this formulation, each category of objects was determined with appropriate aspect ratios for regressing objects of varying orientations. Liu et al. (2021b) developed a center-boundary dual attention network (CBDA-Net) for detecting oriented objects with an anchor-free strategy. The dual attention mechanism served positively for attending to the center and boundary regions of objects, thereby suppressing the influence of the background. In addition, you only look once (YOLO) models (Pham et al., 2020), part-based CNN (Sun et al., 2021), Siamese graph embedding network (Tian et al., 2021), and fisher vectors (Wang et al., 2021a) were also exploited for geospatial object detections.

2.2. Two-stage models

An important component in two-stage object recognition architectures is the region proposal network (RPN), which functions to produce numerous dense object candidates for assisting object recognition and localization. On the issues of size and spatial distribution variations of objects, Zhang et al. (2019) designed a scale-sensitive proposal generation network, which comprised multilayer RPNs for generating object proposals at multiple scales. Jiang et al. (2020) combined the misplaced localization strategy into an encoder-decoder architecture to adapt to the identification of elongated-shape and small-size objects. For promoting the quality of the generated region proposals, Zhong et al. (2018) designed a location-aware balancing network for alleviating the position shift caused by convolution operations. Wang et al. (2021c) proposed a feature-reflowing pyramid network (FRPNet), which was integrated with a nonlocal block for between-region relevance exploitation, to detect multiscale, multiclass objects. Aiming at boosting the feature representation robustness at different scales, Cheng et al. (2021) developed a multiscale feature augmentation strategy, which comprehensively took into account the feature semantics from different scales. Differently, Zheng et al. (2020) constructed a hyper-scale object detection network to exploit hyper-scale feature semantics at different resolutions. Considering the image capturing height, Jin and Lin (2020) constructed a scale-aware network with the assistance of adaptive anchors. The height-based preset of the anchor scales effectively reduced the scale searching space. To enhance the feature representation quality, Chen et al. (2020b) embedded the spatial and channel attention mechanisms into the feature extraction backbone to, respectively, concentrate on the spatial regions related to the foreground and strengthen the useful feature channels.

2.3. Oriented bounding box based models

To effectively handle arbitrarily-oriented objects, some researches improved the RPN to generate more accurate region proposals by using oriented anchors. Fu et al. (2020) developed an orientation-aware CNN for enclosing objects with oriented bounding boxes. In this network, the RPN augmented the anchors with different orientations. Liu et al. (2021a) developed a multidirectional RPN, which can generate oriented region proposals based on a three-side formulation. Li et al. (2019) designed a residual network functioned with rotatable region proposals to detect vehicles of varying orientations. This network adopted a rotatable RPN to produce oriented anchors with a batch averaging rotatable anchor initialization strategy. Aiming at improving the anchor matching efficiency and quality, Xiao et al. (2021) suggested a self-adaptive anchor selection strategy. In their implementation, an adaptive thresholding module and a coordinate regression module were applied to regress accurate rotated bounding boxes. To exploit contextual properties of objects and suppress the background interferences, Ye et al. (2020) developed a feature aggregation and filtering network, where a feature filtering module was used to weaken the background impacts. As for the issue of coarsely labelled data, Shin et al. (2020) suggested a hierarchical multi-label object detection pipeline by using a clustering-guided cropping scheme.

2.4. Anchor-free models

With the purpose of improving the processing efficiency and reducing the task-dependent anchor design, some anchor-free strategies have been recently exploited to produce region proposals. Fang et al. (2020) proposed a semi-anchor-free detector (SAFDet) to handle oriented objects with the assistance of the region of interest (ROI) transformer and attention models. In the SAFDet, an anchor-free-based branch was integrated for highlighting the foreground properties. Yu et al. (2020) developed an orientation guided anchoring (OGA) mechanism for automatically generating lightweight, high-quality oriented region proposals. Through orientation ROI (OROI) pooling, arbitrarily-oriented objects can be recognized in a consistent way. To solve the orientation diversity issue of ships in complicated environments, Yang et al. (2021) presented a one-stage anchor-free architecture, which comprised a detection head for bounding box regression and a center localization head for detection result augmentation. Wang et al. (2019) designed a deconvolutional object detection network, which leveraged a deconvolutional RPN to generate reference boxes. By considering the multilevel feature semantics to guide the region proposal generation, Xu et al. (2020) presented a hierarchical feature propagation architecture for upgrading the object recognition accuracy. Chen et al. (2020a) designed a two-phase pipeline by using a couple of spatial density building nets (SDBNs), which functioned for region proposal generation and object categorization, respectively. In addition, weakly supervised model (Yao et al., 2021a), attention mask R-CNN (Nie et al., 2020), deep hash assisted network (Wang et al., 2020), and global density fused CNN (Zhang et al., 2020) were also designed to detect geospatial objects.

3. Methodology

The architecture of the proposed sparse anchoring guided high-resolution capsule network (SAHR-CapsNet) is presented in Fig. 1, which employs a two-stage processing pipeline to detect arbitrarily-oriented geospatial objects with the assistance of sparse region proposals. The SAHR-CapsNet involves three components: a feature extraction backbone network, a sparse anchoring network, and a geospatial object detection network. The feature extraction backbone network is formulated with a multibranch, high-resolution capsule network architecture, which follows the HRNet architecture (Sun et al., 2019; Wang et al., 2021b) and can provide semantically strong and spatially accurate feature representations at multiple scales. The sparse anchoring network can automatically output a set of sparse and oriented region proposals at each feature scale. The object detection network converts the different-size, varying-orientation region proposals into a consistent representation to conduct object recognition and fine-grained bounding box regression.

3.1. Revisit of capsule network

Capsules are structurized as a one-dimensional tensor formulation, which consists of a set of instantiation parameters. A significant property is that, by using the tensor-form capsules, capsule networks can simultaneously encode the feature existence probability based on the capsule length and the inherent properties through the instantiation parameters. More importantly, such a tensor formulation enables a capsule to recognize a feature and adapt to its variants by adjusting its instantiation parameters. Therefore, due to the advantageous characteristics, capsule networks are positively leveraged in different remote sensing applications, including object detection (Yu et al., 2019), object segmentation (Ren et al., 2020), change detection (Xu et al., 2021), and land cover mapping (Jiang et al., 2021).

In the capsule networks, a capsule takes the following transformed, weighted sum of the predictions from the prepositive capsules as the input:

$$\boldsymbol{C}_j = \sum_i a_{ij} \boldsymbol{W}_{ij} \boldsymbol{U}_i \tag{1}$$

where $\boldsymbol{C}_j$ denotes the input to a capsule j; $\boldsymbol{U}_i$ represents the output of a prepositive capsule i; $\mathbf{W}_{ij}$ acts as a transformation matrix; a_{ij} denotes a contribution coefficient reflecting the amount of prediction cast by capsule i. These coefficients can be computed by an improved version of the dynamic routing process (Rajasegaran et al., 2019).

Since the feature saliency is encoded by the capsule length, longer capsules should contribute more to the predictions, whereas shorter capsules should be considered less to the predictions. In this regard, the following squashing function (Sabour et al., 2017) is used to transform the aggregated predictions to a capsule:

$$\boldsymbol{U}_j = \frac{\|\boldsymbol{C}_j\|^2}{\|\boldsymbol{C}_j\|^2 + 1} \cdot \frac{\boldsymbol{C}_j}{\|\boldsymbol{C}_j\|} \tag{2}$$

where $\boldsymbol{C}_j$ and $\boldsymbol{U}_j$ denote the input and output of the capsule, respectively. Following this transformation, the lengths of long capsules are augmented to contribute more, whereas the lengths of short capsules are suppressed to contribute less.

3.2. Feature extraction backbone network

Differing from the common deep learning models that usually follow a cascaded pattern to mine multiscale/multilevel features, the HRNet opens up a new design pattern by paralleling multiple branches to simultaneously extract high-level features at different scales. Thus, taking advantage of the powerful high-order feature encoding capability of capsules and the superior multiscale feature representation property of the HRNet architecture, the feature extraction backbone network is designed with a high-resolution capsule network (HR-CapsNet) architecture aiming at extracting multiscale high-level feature

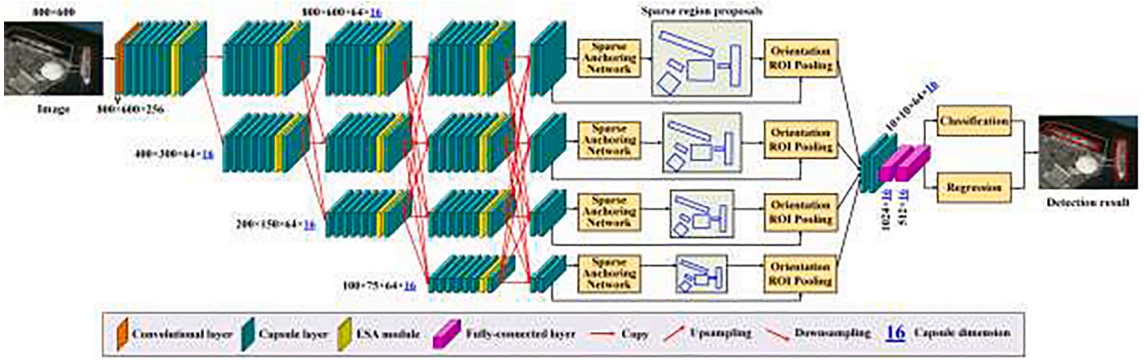

Fig. 1. Architecture of the proposed sparse anchoring guided high-resolution capsule network.

representations for effective multiscale object detection. As shown in Fig. 1, the HR-CapsNet backbone comprises multiple parallel branches of different resolutions, each of which dedicates to extracting high-level and strong feature semantics at a particular scale. An important property of the multibranch parallel architecture is that high-resolution feature representations can be maintained through the entire network to gradually achieve high-level feature encodings.

The HR-CapsNet backbone starts with a high-resolution branch for maintaining the identical spatial resolution to the input image. This is the key component to realize spatially accurate feature extraction. Then, high-to-low resolution branches are gradually added in a parallel manner to access larger feature contexts with the gradual enlargement of the receptive fields. In our architecture, the scaling factor is set as 0.5. Note that, the feature maps in the same branch maintain the same spatial resolutions and sizes, which is beneficial to avoiding the localization accuracy loss at a particular scale. Specifically, the initial high-resolution branch starts with two scalar convolutional layers for extracting low-level features, which are further leveraged to construct high-order capsule formulations.

To facilitate capsule feature computation, the high-to-low resolution branch addition and the multiscale feature fusion processes are modified. As shown by Fig. 2(a), to add a new lower-resolution branch, the different-resolution feature maps from the established branches are first downscaled to the same spatial size as configured by the newly added branch for facilitating feature concatenation. Then, the concatenated feature maps are operated by a 1×1 capsule convolution for feature fusion, resulting in the primary lower-resolution feature map. Meanwhile, multiscale feature propagation is carried out among the previously established branches to comprehensively take into account the multiscale, multiresolution features to promote the feature semantics at each scale. Concretely, as shown by Fig. 2(b) to (d), when propagating features from a higher-resolution branch, a feature map is downscaled to the spatial size as configured by the target branch. In contrast, when propagating features from a lower-resolution branch, a feature map is upscaled to the spatial size as configured by the target branch. Then, the feature map from the target branch is directly copied and concatenated with the scale-adjusted feature maps from the other branches. Finally, the concatenated feature maps are operated by a 1×1 capsule convolution for feature fusion, resulting in a feature map with strong semantics at the target branch. This multiscale feature propagation process is the key technique of the HR-CapsNet, which effectively boosts the feature semantics to achieve high-level feature representations at different scales.

As illustrated by Fig. 1, this multiscale feature propagation and fusion mechanism is carried out several times to continually boost the feature quality at each scale. Finally, the HR-CapsNet backbone outputs a group of multiscale semantically strong and spatially accurate feature maps, which perform excellently to characterize geospatial objects of varying sizes.

Note that, for each branch of the HR-CapsNet backbone, the capsule convolutions operate almost equally on the feature maps towards capsule feature extraction. On one hand, the informativeness of different feature channels are characterized weakly, thereby unfavorable to get high-quality feature representations. On the other hand, the spatial features covering the foreground areas are not well focused on to weaken the influences of the background areas, thereby not helpful to obtain class-specific feature encodings. Therefore, aiming at further boosting the quality of the multiscale feature semantics extracted by the HR-CapsNet backbone to emphasize the informative feature semantics and suppress the useless ones, we construct a capsule-based efficient self-attention (ESA) module. As illustrated by Fig. 3, the ESA module comprises a spatial feature attention (SFA) unit and a channel feature attention (CFA) unit for modulating the spatial and channel features, respectively.

In the CFA unit, the input multi-dimensional feature map is first operated by a 1×1 capsule convolution to get a one-dimensional feature map $F_A \in R^{H \times W \times 64}$ (H and W denote the height and width of the input feature map) that reflects the channel-wise feature saliencies. That is, the capsules with higher responses have more salient features in the corresponding channel. Considering the size variations of geospatial objects, the CFA unit is designed with two parallel branches for, respectively, exploiting the global and local channel-wise interdependencies with the purpose of effectively and simultaneously emphasizing the large-size and small-size objects. Concretely, the first branch starts with a global average pooling (GAP) operated on feature map F_A for collecting the channel-wise informativeness with a global perspective for emphasizing large-size objects; whereas, the second branch starts with a local average pooling (LAP), having the stride of 1 and the kernel size of $n \times n$, performed on feature map F_A to collect the channel-wise informativeness with a local perspective for emphasizing small-size objects. Then, for each branch, two point-wise convolution (PConv) based convolutional layers, with the kernel size of 1×1, are connected up for exploiting cross-channel interdependencies. The two branches produce two feature maps $C_G \in R^{1 \times 1 \times 64}$ and $C_L \in R^{H \times W \times 64}$, which can be treated as two attention maps and reflect the significance of the feature channels of the input feature map with global and local perspectives, respectively. To be specific, each element of C_G encodes the informativeness of the corresponding channel of the input feature map and the element in each channel of C_L encodes the significance of the feature at the corresponding position of the input feature map. Finally, these two attention maps are added in a channel-wise manner and activated with the sigmoid function for obtaining the channel attention map that is used as weight factors to promote the contributions of the useful feature semantics. The channel attention map $A_C \in R^{H \times W \times 64}$ is computed as follows:

$$A_C(i,j,c) = sigmoid(C_G(c) + C_L(i,j,c)) \tag{3}$$

where $A_C(i,j,c)$ is the element at position (i, j) in the c-th channel of A_C, $C_G(c)$ is the c-th element of C_G, $C_L(i,j,c)$ is the element at position (i, j) in the c-th channel of C_L, and $sigmoid(\cdot)$ denotes the sigmoid function.

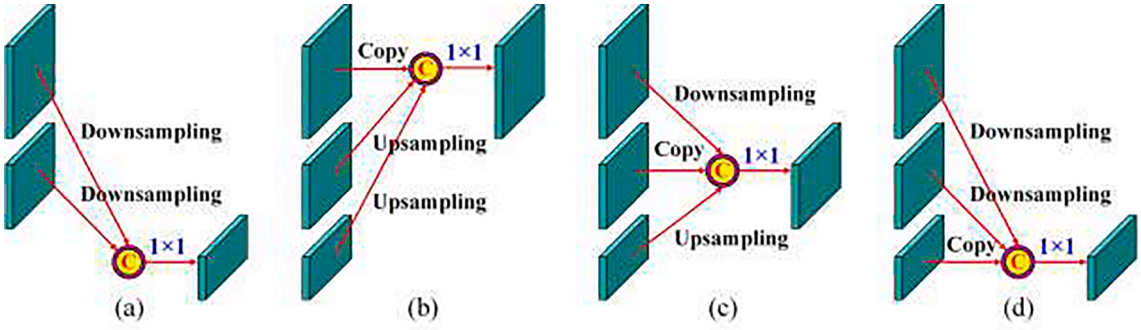

Fig. 2. Illustrations of (a) adding a new lower-resolution branch and multiscale feature propagation for generating (b) high-, (c) medium-, and (d) low-resolution feature maps.

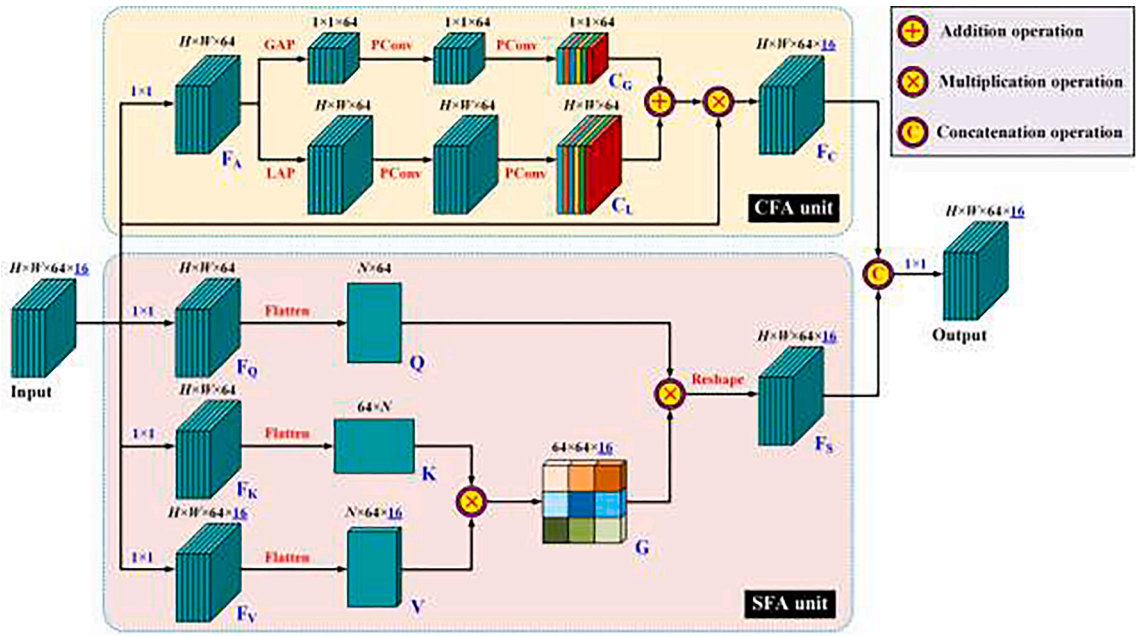

Fig. 3. Structure of the proposed efficient self-attention (ESA) module. *H* and *W* are the height and width of the input feature map.

Eventually, the channel-wisely recalibrated feature map F_C is obtained by multiplying the input feature map with the channel attention map A_C channel-wisely and element-wisely.

In the SFA unit, the input feature map is first operated by three 1×1 capsule convolutions to get the value feature map $F_V \in R^{H\times W\times 64\times 16}$, the key feature map $F_K \in R^{H\times W\times 64}$, and the query feature map $F_Q \in R^{H\times W\times 64}$, where W and H denote the input feature map's width and height. Specifically, each capsule of F_V represents a response at a position in the input feature map. That is, the longer the length of a capsule, the more the feature responses of the capsule. Each channel in F_K can be viewed as a spatial attention map that encodes a kind of spatial semantic property about the responses in each channel of F_V. The weights in the spatial attention map reflect the contribution levels of the corresponding capsules for calculating the spatial semantic property. The elements at each position of F_Q represent the spatial attention coefficients for the spatial semantic properties at that position. These coefficients cooperatively determine the feature saliency at the corresponding position of the input feature map. To facilitate computation, we channel-wisely flatten these three feature maps and reshape them to constitute the value matrix $V \in R^{N\times 64\times 16}$, the key matrix $K \in R^{64\times N}$, and the query matrix $Q \in R^{N\times 64}$, where $N = W \times H$ denotes the number of positions of the input feature map. Next, a global context matrix $G \in R^{64\times 64\times 16}$ is produced by performing matrix multiplication between K and V (i.e., KV). Here, K acts as weight factors covering all the positions of V to aggregate the responses to generate a set of spatial semantic properties. Therefore, a spatial semantic property is encoded in each row of G. To be specific, each row of K is activated by a softmax function before carrying out matrix multiplication. Afterwards, matrix multiplication is performed between Q and G (i.e., QG) to comprehensively and weightedly take into account the spatial semantic properties to obtain a spatially modulated feature at each position. Eventually, by rearranging each column of the product matrix into a feature channel, we obtain a spatially recalibrated feature map F_S. Note that, by adopting such a spatial feature attention mechanism (i.e., following the calculation sequence of Q(KV)), the computation complexity and the number of parameters are significantly reduced compared with the non-local block formulation (Wang et al., 2018), which adopts the calculation sequence of (QK)V to conduct spatial feature recalibration. As illustrated by Fig. 3, the feature maps F_S and F_C produced by the SFA and CFA units are concatenated for fusion by performing a 1×1 capsule convolution, resulting in a quality-boosted feature map that explicitly attends to both the channel-wisely informative and spatially class-specific features.

As illustrated by Fig. 1, the ESA module is integrated into each branch of the HR-CapsNet backbone to promote the feature representation quality at each scale. Concretely, for each branch, before carrying out feature propagation, the feature map is first fed into the ESA module to conduct feature recalibration. Then, the quality-boosted feature map output by the ESA module is used for cross-branch feature propagation and fusion.

3.3. Sparse anchoring network

As shown in Fig. 4, to characterize geospatial objects of arbitrary orientations, we leverage the following five-tuple representation: (x, y, h, w, α), where (x, y) represents the center of an object, h and w represent the height and width of the oriented bounding box of the object, and α represents the orientation of the bounding box. Specifically, h is defined as the bounding box's short side, w is defined as the bounding box's long side, and $\alpha \in [0,\pi)$ is defined as the angle included between the direction parallel to the bounding box's long side and the positive direction of the

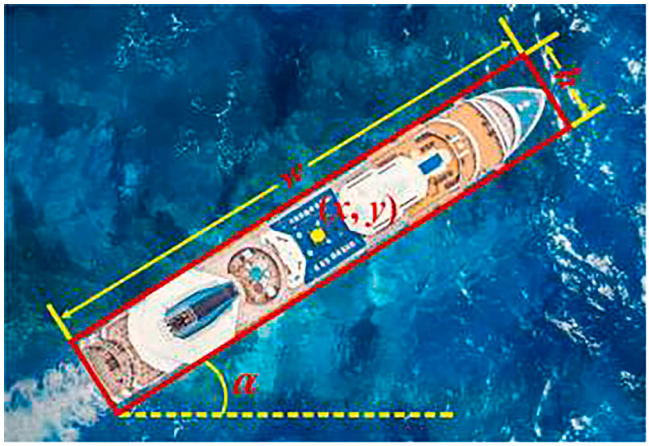

Fig. 4. Illustration of the five-tuple representation of an arbitrarily-oriented object.

x-axis along clockwise direction.

The RPN-based models generally deploy several predefined anchors having different aspect ratios and scales at each position of the feature map to direct the generation of object region proposals. The deficiency of the anchor-based strategy lies in the following two aspects: First, in most cases, the anchor design is task-dependent. That is, a new set of anchors are usually required to be designed for different detection tasks. Second, the processing efficiency is degraded caused by the regressions and selections of the large-volume anchors. Therefore, exploiting an effective technique to automatically produce a small quantity of region proposals of high qualities completely covering foreground regions with an anchor-free manner without the predetermination of anchors is significantly favorable to enhance the two-stage object recognition efficiency.

In this paper, we develop a sparse anchoring network (SAN) to conduct object region proposal generation based on the above five-tuple representation. The novelties of the SAN are embodied in the following two aspects: First, the generation of region proposals is task-independent and anchor-free without the predesign of anchors. Second, a small, fixed number of high-quality region proposals are automatically generated to encapsulate the objects of interest. Therefore, the object detection efficiency can be dramatically promoted and the detection accuracy can be well maintained in the meantime. As shown in Fig. 5, the SAN is designed with a lightweight capsule fully-connected network with a fixed number of outputs. Concretely, two capsule fully-connected layers are mounted on the input feature map to access the feature contexts with a global perspective. The output layer involves M sets of fully-connected layers, each of which contains five one-dimensional capsules for predicting the five parameters of an oriented region proposal. That is, only M region proposals distributing on the input feature map are generated rather than the selected ones from hundreds of thousands of candidates in the RPN. Note that, since the object sizes might vary greatly in a feature map, instead of directly predicting the large-range parameters (x, y, h, w), we adopt the following transformations to restrain them to a small range:

$$x = dxH_{\mathrm{F}} \tag{4}$$

$$y = dyW_{\mathrm{F}} \tag{5}$$

$$h = \frac{\sqrt{W_{\mathrm{F}}^2 + H_{\mathrm{F}}^2}}{2} e^{dh} \tag{6}$$

$$w = \frac{\sqrt{W_{\mathrm{F}}^2 + H_{\mathrm{F}}^2}}{2} e^{dw} \tag{7}$$

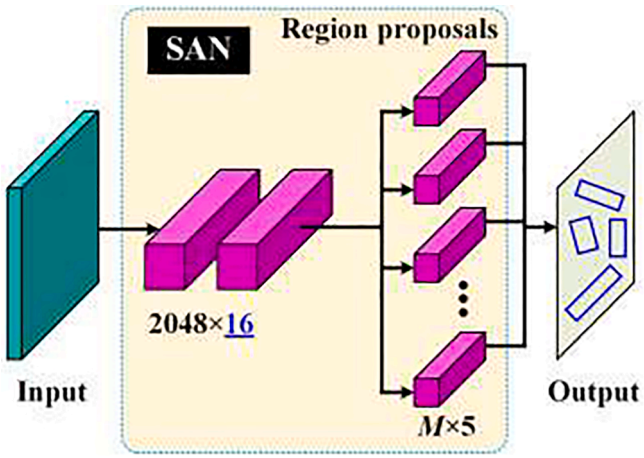

Fig. 5. Architecture of the sparse anchoring network for object region proposal generation.

where H_{F} and W_{F} denote the height and width of the input feature map. The SAN only requires to predict the small-range parameters (dx, dy, dh, dw) and the parameter α.

The SAN can be constructed using the set matching loss function as follows:

$$L_{\mathrm{SAN}} = \mu_1 L_{\mathrm{cls}} + \mu_2 L_{\mathrm{reg}} + \mu_3 L_{\mathrm{GIoU}} \tag{8}$$

where μ_1, μ_2, and μ_3 are the regularization factors for adjusting the significances of the loss terms. Specifically, L_{cls} is formulated as the focal loss (Lin et al., 2017) between the ground-truth category and the classification prediction of the region proposal. L_{reg} is formulated as the smooth-L_1 loss (Girshick, 2015) between the regressed parameters of the region proposal and the ground-truth parameters of its matched bounding box. L_{GIoU} is defined by the generalized intersection over union (GIoU) loss (Rezatofighi et al., 2019) between the generated region proposal and its matched ground-truth bounding box.

3.4. Object detection network

As illustrated by Fig. 1, the SAN is mounted on the multiscale feature maps provided by the HR-CapsNet backbone to produce a fixed quantity of region proposals at each scale. These region proposals, along with the feature semantics enclosed by the region proposals, are leveraged to recognize objects. Noteworthily, the shapes, sizes, and orientations of these region proposals vary greatly in different feature maps, even in the same feature map, which brings difficulties to carry out object identification in a consistent manner. Thus, in this paper, we adopt the OROI pooling strategy (Yu et al., 2020) to eliminate the orientations of the arbitrarily-oriented region proposals and convert the varying-shape and varying-size region proposals into a fixed size. The transformed region proposals with the same size are input to the object detection network to conduct object identification.

As illustrated by Fig. 1, the object detection network is designed with a lightweight capsule network, which comprises some capsule convolutional and fully-connected layers, as well as two parallel task-specific output layers used for, respectively, region proposal categorization and fine-grained object bounding box determination. The softmax classification layer involves $V + 1$ outputs, representing the background and the V categories of objects, respectively. When an object is confirmed to be enclosed in a region proposal, the regression layer outputs the fine-tuned five-tuple parameters of the object's bounding box.

The object detection network can be constructed by the multitask loss function as follows:

$$L_{det} = \lambda_1 L_{\mathrm{cls}} + \lambda_2 L_{\mathrm{reg}} \tag{9}$$

where λ_1 and λ_2 are the regularization factors for adjusting the significances of the loss terms. To be specific, the classification loss term L_{cls} is computed by the focal loss (Lin et al., 2017) of the softmax probability prediction output by the ground-truth category neuron.

To further refine the object bounding box associated with a region proposal, we leverage the following scale-insensitive parameterization strategy to regress the offset, rotation, and height and width shifts related to a region proposal:

$$\begin{gathered} d_x = (x - x_{\mathrm{r}})/\sqrt{w_{\mathrm{r}}^2 + h_{\mathrm{r}}^2},\ d_y = (y - y_{\mathrm{r}})/\sqrt{w_{\mathrm{r}}^2 + h_{\mathrm{r}}^2} \\ d_h = log(h/h_{\mathrm{r}}),\ d_w = log(w/w_{\mathrm{r}}) \\ d_\alpha = \alpha - \alpha_{\mathrm{r}} \end{gathered} \tag{10}$$

$$\begin{gathered} d_x^* = (x^* - x_{\mathrm{r}})/\sqrt{w_{\mathrm{r}}^2 + h_{\mathrm{r}}^2},\ d_y^* = (y^* - y_{\mathrm{r}})/\sqrt{w_{\mathrm{r}}^2 + h_{\mathrm{r}}^2} \\ d_h^* = log(h^*/h_{\mathrm{r}}),\ d_w^* = log(w^*/w_{\mathrm{r}}) \\ d_\alpha^* = \alpha^* - \alpha_{\mathrm{r}} \end{gathered} \tag{11}$$

where (d_x, d_y, d_h, d_w, d_α) denote the predicted regression parameters of a region proposal, $(d_x^*, d_y^*, d_h^*, d_w^*, d_\alpha^*)$ denote the ground-truth

parameters to regress the region proposal, $(x_r, y_r, h_r, w_r, \alpha_r)$, (x, y, h, w, α), and $(x^*, y^*, h^*, w^*, \alpha^*)$, respectively, represent the region proposal, the predicted bounding box, and the ground-truth bounding box. According to the above parameterization representations, the regression loss term L_{reg} is computed by the smooth-L_1 loss (Girshick, 2015) between (d_x, d_y, d_h, d_w, d_α) and $(d_x^*, d_y^*, d_h^*, d_w^*, d_\alpha^*)$.

4. Results and discussions

4.1. Datasets

In this paper, two large-scale remote sensing image datasets were examined to assess the detection effectiveness of the developed SAHR-CapsNet. The first dataset is the GOD^{2}18 dataset (Yu et al., 2020). This dataset consists of 22,000 images, including 4000 aerial images collected by a UAV system and 18,000 satellite images collected using the Google Earth service. A total of 69,207 instances covering four categories of geospatial objects are annotated by both oriented bounding boxes and horizontal bounding boxes. The images in the GOD^{2}18 dataset have the same size of 800 × 600 pixels. The second dataset is the DOTA dataset (Xia et al., 2018). This dataset includes 2086 satellite images covering fifteen categories of geospatial objects. In total, 188,282 instances are annotated with arbitrary quadrilaterals. The images in the DOTA dataset have different sizes ranging from about 800 × 800 pixels to about 4000 × 4000 pixels.

4.2. Network training and parameter configuration

The SAHR-CapsNet was trained with the Adam optimizer using a cloud computing environment configured with a 16-core CPU, ten 16-GB GPUs, and a 128-GB memory. The parameter volume of the SAHR-CapsNet is about 19 M. As for the SAHR-CapsNet, the SAN and the object detection network share the HR-CapsNet backbone. Furthermore, the object detection network relies on the region proposals generated by the SAN to proceed object identification. To solve this issue, we adopted a "divide-and-conquer" strategy to train them separately. Concretely, first, we trained the SAN along with the HR-CapsNet backbone based on the loss function in Eq. (8). The number of region proposals was set as $M = 400$, and the regularization factors μ_1, μ_2, and μ_3 were configured as 1.0, 0.2, and 1.0, respectively, after intensive performance evaluations. We configured 800 training epochs and distributed two images per batch to a GPU. Specifically, in the first 600 epochs, we configured the learning rate as 0.001, then, in the rest 200 epochs, decreased it to 0.0001. When the SAN was constructed, the parameters of the HR-CapsNet backbone and the SAN were fixed, and the object detection network was trained based on the loss function in Eq. (9). After intensive performance evaluations, the regularization factors λ_1 and λ_2 were configured as 1.0 and 0.2, respectively. For each training image, the region proposals produced by the SAN along with the feature maps extracted by the HR-CapsNet backbone were used to construct the object detection network. We configured 600 training epochs and distributed 50 region proposals per batch to a GPU. Specifically, in the first 400 epochs, we configured the learning rate as 0.001, then, in the rest 200 epochs, decreased it to 0.0001. Finally, the SAN and the object detection network were jointly optimized to refine the network parameters of the entire SAHR-CapsNet with the combination of the loss functions in Eqs. (8) and (9). For joint optimization, we configured the learning rate as 0.0001 and trained the SAHR-CapsNet for 200 epochs.

4.3. Parameter sensitivity analysis

In the proposed SAHR-CapsNet, there are two important parameters having great impacts on the object detection performance: the number of sparse region proposals M generated by the SAN and the size of the global context matrix G. To determine the optimal configurations for these two parameters, we conducted a set of experiments to analyze the sensitivities of their configurations to the object detection performance.

In our experiments, we tested the following seven configurations for M: 50, 100, 200, 300, 400, 500, and 600, and tested the following seven configurations for the size of G: 32, 48, 64, 96, 128, 256, and 512. The performances of different configurations of these two parameters were reported and analyzed using the precision-recall curves. As shown by Fig. 6(a), when the value of M increased from 50 to 400, the object detection accuracy enhanced dramatically. This is because, initially, the objects of interest in some images cannot be completely covered with a small number of region proposals, thereby leading to a low recall value. Then, as the number of region proposals increased, the objects of interest in the images can be better and better covered, thereby resulting in a promotion of the object detection accuracy. However, when the value of M exceeded 400, the improvement of the object detection accuracy was quite slight. The reason is that the number of 400 region proposals performed promisingly to well encapsulate the objects of interest in the images. Thus, the addition of more region proposals helped slightly to the promotion of the object detection accuracy. Furthermore, more region proposals would increase the computation overhead of the SAHR-CapsNet. Thus, by balancing the object detection accuracy and the computational performance, we configured the value of M as 400.

As shown by Fig. 6(b), the object detection accuracy kept upgrading as the size of G increased from 32 to 64. As a matter of fact, the size of G implies the amount of the spatial semantic properties encoded. Theoretically, the larger the size of G, the more the spatial semantic properties. Thus, with the enlargement of the spatial semantic properties encoded in G as the size of G increased, the feature representation quality was gradually promoted, thereby leading to the enhancement of the object detection accuracy. However, when the size of G was greater than 64, the object detection accuracy was improved quite slightly. This is because, a very large size of G might produce redundant and insignificant spatial semantic properties, which helped less to the promotion of the feature encoding quality. Moreover, the increase of the size of G also brought dramatic increase of the computation overhead. Thus, by trading off between the object detection accuracy and the computational performance, we configured the size of G as 64.

4.4. Geospatial object detection

To quantitatively examine the proposed SAHR-CapsNet in geospatial object detection tasks, the following three assessment measures were leveraged: precision, recall, and F_1-score. To be specific, precision and recall evaluate the capability of an object detection model in distinguishing the true targets and the false alarms. F_1-score provides an overall accuracy evaluation by comprehensively taking into consideration the precision and recall metrics. These assessment measures are formulated in the following forms:

$$precision = \frac{TP}{FP + TP} \times 100\% \tag{12}$$

$$recall = \frac{TP}{FN + TP} \times 100\% \tag{13}$$

$$F_1\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \times 100\% \tag{14}$$

where the quantities of true positives, false positives, and false negatives are, respectively, denoted by TP, FP, and FN. The object detection performances evaluated on the two datasets by using these three evaluation metrics are reported in Table 1.

As Table 1 reports, the proposed SAHR-CapsNet performed effectively in detecting geospatial objects from these two datasets. Concretely, for the GOD^{2}18 dataset, a detection accuracy with the precision, recall, and F_1-score of 98.23%, 94.16%, and 96.15%, respectively, was achieved. For the DOTA dataset, the SAHR-CapsNet obtained

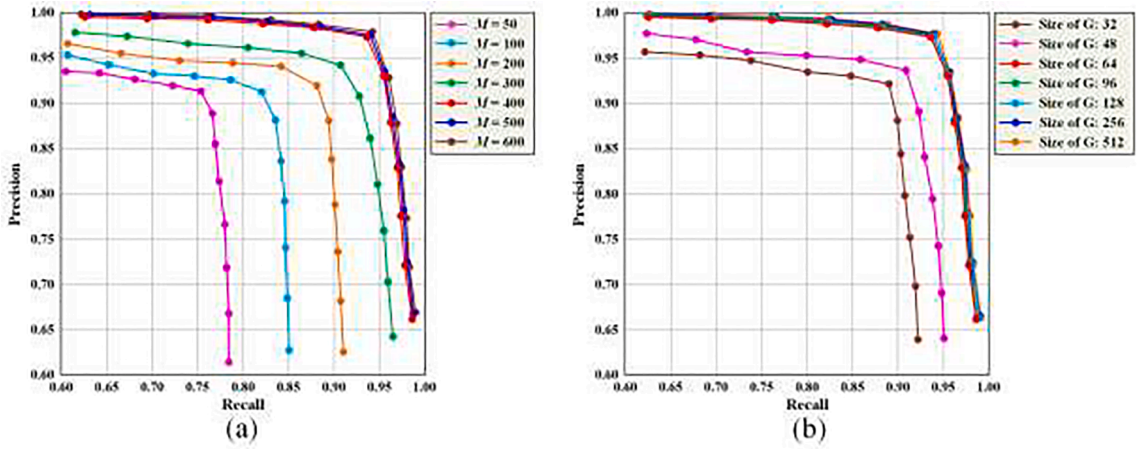

Fig. 6. Precision-recall curves of (a) different configurations of the number of sparse region proposals and (b) different configurations of the size of the global context matrix.

Table 1
Geospatial object detection performances of different models.

Models	Dataset	Precision (%)	Recall (%)	F_1-score (%)	Speed (images/s)
SAHR-CapsNet	GOD^{2}18	98.23	94.16	96.15	11
	DOTA	96.95	93.04	94.95	
SAHRNet	GOD^{2}18	97.25	93.12	95.14	12
	DOTA	95.92	91.96	93.90	
SAHR-CapsNet-N	GOD^{2}18	98.24	94.16	96.16	9
	DOTA	96.96	93.06	94.97	
SAHR-CapsNet-A	GOD^{2}18	97.94	93.82	95.84	11
	DOTA	96.41	92.79	94.57	
RPNHR-CapsNet	GOD^{2}18	98.27	94.22	96.20	5
	DOTA	96.97	93.11	95.00	
GTGCF-Net	GOD^{2}18	96.65	92.51	94.53	13
	DOTA	95.40	91.57	93.45	
SARA-Net	GOD^{2}18	92.74	90.12	91.41	11
	DOTA	91.95	89.36	90.64	
CBDA-Net	GOD^{2}18	91.52	88.86	90.17	50
	DOTA	90.65	88.47	89.55	
RM-CNN	GOD^{2}18	96.93	92.79	94.81	4
	DOTA	95.71	91.83	93.73	
SAS-Net	GOD^{2}18	93.36	90.93	92.13	4
	DOTA	92.62	89.91	91.24	
OGA-Net	GOD^{2}18	96.48	92.32	94.35	8
	DOTA	95.22	91.38	93.26	

a detection accuracy with the precision, recall, and F_1-score of 96.95%, 93.04%, and 94.95%, respectively. Comparatively, a better performance was obtained on the GOD^{2}18 dataset due to the more challenging scenarios of the DOTA dataset. Specifically, for the GOD^{2}18 dataset, the degradation of the recall was mainly caused by the densely-distributed, shadow-covered, and tree-occluded vehicles, resulting in the increase of the missing detections. In contrast, for the DOTA dataset, the recall degradation was mainly caused by the densely-distributed and varying-size ships parked along the harbors, leading to a failure in correctly separating some ships. Moreover, the false alarms were caused by the land covers showing extremely similar textural and geometrical properties to the objects of interest. As a whole, the proposed SAHR-CapsNet showed quite promising and competitive object detection performance on the two challenging datasets with a small quantity of missing detections and false identifications. Specifically, an average detection accuracy with the precision, recall, and F_1-score of 97.59%, 93.60%, and 95.55%, respectively, was obtained with respect to these two datasets.

The remarkable challenges of the GOD^{2}18 and DOTA datasets reflect in the following cases: (1) instances with various orientations due to the bird-view image capturing mode; (2) intra-class size variations of the instances; (3) interclass size variations of the instances; (4) instances with diverse spatial distributions and densities; (5) instances having different textural properties; (6) instances having different aspect ratios; (7) instances having arbitrary shapes; (8) instances suffering from different-level occlusions caused by overhead objects; (9) instances covered by different-level shadows caused by nearby high-rise objects; (10) similarities between the objects of interest and the non-targets; (11) differences in image qualities and exposure conditions caused by different imaging sensors and illumination condition changes; and (12) complicated surrounding environments of the objects of interest in the images. All of the above cases might cause the degradation of the object detection accuracy due to the failure in correctly locating and identifying some instances. However, the proposed SAHR-CapsNet still behaved competitively with low false detection rates and high recognition rates in handling the geospatial objects of diverse self-conditions and different surrounding environments. The advantageous performance was embodied in the following aspects. First, constructed with a capsule network architecture, the SAHR-CapsNet can abstract advanced high-order capsule representations to well encode entity features, which serves for promoting the feature saliencies and distinguishabilities. Second, formulated with a multibranch HR-CapsNet backbone assisted by the multiscale feature propagation and fusion technique, the SAHR-CapsNet can extract semantically strong and spatially accurate feature semantics at multiple scales, which favors significantly to the detection of varying-size objects. Third, assisted by the ESA module to recalibrate the spatial and channel features, the SAHR-CapsNet can focus on the target-specific spatial features and the informative channel features, which positively boosts the feature encoding capability and robustness. Last but not least, designed with the SAN for automatically generating high-quality oriented region proposals, the SAHR-CapsNet can perform effectively in detecting geospatial objects of arbitrary orientations and different aspect ratios.

For qualitative evaluations, Figs. 7 and 8 also present two subsets of the geospatial object detection results from the GOD^{2}18 and DOTA datasets. Overall, the majority of the geospatial objects exhibiting different sizes and orientations, varying densities and spatial distributions, diverse aspect ratios and shapes, and complex surface and environmental conditions were correctly located and identified. Specifically, as shown in Figs. 7 and 8, the vehicles parked in the parking lot and the ships parked along the harbor distributed closely in a parallel manner and exhibited high densities. Generally, horizontal bounding box based

Fig. 7. Sample geospatial object detection results from the $GOD^2$18 dataset.

Fig. 8. Sample geospatial object detection results from the DOTA dataset.

approaches might fail to guarantee the integrity of the detected instances. However, due to the use of oriented bounding boxes, our proposed SAHR-CapsNet showed high quality in handling such challenging scenarios. In addition, the instances showed extremely large size variations in the images, especially the existence of very small-size instances (e.g., vehicles, ships, storage tanks, etc.). Accurately recognizing the small-size instances is a difficult task due to the lack of informative and sufficient feature presence in the feature maps. Fortunately, benefitting from the design of the multibranch HR-CapsNet backbone for maintaining the high-resolution feature semantics through the entire network and the integration of the ESA module for highlighting the foreground features, our proposed SAHR-CapsNet performed competitively in detecting the varying-size instances. Furthermore, some instances not severely occluded by the nearby overhead land covers (e.g., vehicles parked under trees) and some instances contaminated by the shadows cast by nearby high-rise objects were also correctly identified due to the robust, high-order, and task-oriented capsule feature encodings extracted by the HR-CapsNet backbone. Last but not least, owing to the design of the SAN to automatically generate independent and oriented region proposals based on the multiscale high-quality feature maps, the instances of diverse aspect ratios and the overlapped instances (e.g., the roundabouts and the vehicles) were also accurately detected.

To evaluate the computational performance of the proposed SAHR-CapsNet, the processing time was also recorded on the test datasets at the detection stage. On average, the SAHR-CapsNet achieved a processing speed of about 11 image patches per second on a GPU. The computational performance gain benefitted from the sparse anchoring network by generating a fixed set of lightweight region proposals without the extra time cost in dense anchor regression and high-quality region proposal selection.

4.5. Ablation studies

For the purpose of verifying the effectiveness and efficiency of the proposed network architecture and the designed modules, a set of ablation studies were conducted for performance comparisons. First, to compare the performance between capsule formulation and scalar neuron formulation, we redesigned the model by replacing only the capsules with scalar neurons, leaving the entire network architecture unchanged. As a result, the feature extraction backbone was formulated as the HRNet architecture. We termed the modified model as SAHRNet. Second, to compare the efficiency between the proposed SFA unit and

the non-local block architecture, we replaced only the SFA unit with a capsule-based non-local block formulation and kept the other parts of the network architecture unchanged. We termed the modified model as SAHR-CapsNet-N. Third, to compare the performance between the concatenation operation and the addition operation in the ESA module for fusing the features from the CFA and SFA units, we replaced only the concatenation operation with the addition operation in the ESA module and kept the rest parts of the ESA module unchanged. We termed the modified model as SAHR-CapsNet-A. Finally, to compare the efficiency between the proposed SAN and the commonly used RPN architecture that deploys dense anchors at each position to direct region proposal generation, we replaced only the SAN with the rotatable RPN (Li et al., 2019), leaving the remaining parts of the network architecture unchanged, to generate region proposals. We termed the modified model as RPNHR-CapsNet. To provide fair assessments, both of the GOD^{2}18 and DOTA datasets were used to optimize these models with the same parameter configurations and training strategies and used to evaluate the performances of these models.

The quantitative evaluation results and the computational performances of these modified models are reported in Table 1. Apparently, the SAHRNet behaved less effectively than the SAHR-CapsNet on both of the two datasets. Specifically, an accuracy degradation by about 1.05% appeared on the DOTA dataset with respect to the F_1-score. It confirmed that the capsule formulation showed superior performance than the scalar neuron formulation in extracting high-quality feature representations. Note that, the SAHR-CapsNet-N obtained similar performance to the SAHR-CapsNet on both of the two datasets. Thus, it demonstrated that the SFA unit was effective and showed competitive performance with the non-local block formulation. However, the SAHR-CapsNet-N exhibited significantly lower computation performance than that of the SAHR-CapsNet with an average processing speed of about 9 image patches per second on a GPU. The computation performance degradation was mainly caused by the burdensome matrix computations and the large set of parameters. As a result, the SFA unit performed more efficiently than the non-local block architecture. As shown in Table 1, the SAHR-CapsNet-A exhibited a slight performance decline compared with the SAHR-CapsNet. Although the difference was not very significant, it still proved that the concatenation operation performed better than the addition operation. This is because, compared with the concatenation operation, the addition operation might mix up the distinguishing capsule properties from the CFA and SFA units. Consequently, the feature representation quality might be slightly influenced. In addition, the RPNHR-CapsNet showed no significant performance gain on the test datasets compared with the SAHR-CapsNet. However, a significant efficiency decline was obtained by the RPNHR-CapsNet with an average processing speed of about 5 image patches per second on a GPU. This was caused by the extra time cost in regressing the large numbers of dense anchors and the selection of the high-quality region proposals. In conclusion, the SAN performed equally with the RPN in detection accuracy, but behaved superiorly to the RPN in processing efficiency.

4.6. Comparative studies

Aiming at further examining the feasibility and effectiveness of the developed SAHR-CapsNet, a group of intensive experiments were conducted with the recently proposed deep learning based models for performance comparisons. The selected models include the following: geometric transform and global contextual feature fusion network (GTGCF-Net) (Shi et al., 2021), self-adaptive aspect ratio anchor network (SARA-Net) (Hou et al., 2021), center-boundary dual attention network (CBDA-Net) (Liu et al., 2021b), rotation-aware and multiscale CNN (RM-CNN) (Fu et al., 2020), self-adaptive anchor selection network (SAS-Net) (Xiao et al., 2021), and OGA network (OGA-Net) (Yu et al., 2020). Specifically, the GTGCF-Net, SARA-Net, and CBDA-Net are one-stage object detection models and the RM-CNN, SAS-Net, and OGA-Net are two-stage object detection models. In addition, the CBDA-Net and OGA-Net are anchor-free models and the others are anchor-based models. To provide reasonable comparisons, both of the GOD^{2}18 and DOTA datasets were used to optimize the network parameters and evaluate the performances of these models. Specifically, the optimal parameters and training strategies were leveraged to construct these models. Concretely, for the GTGCF-Net, the learning rate of 0.001 was configured for the first 50 epochs. In the rest 350 epochs, the learning rate was initially configured as 0.0001 and decayed by 10 at epochs 120 and 240. For the SARA-Net, a total of 12 epochs were trained with the learning rate, momentum, and weight decay of 0.001, 0.9, and 0.001, respectively. For the CBDA-Net, a total of 140 epochs were trained. The learning rate was initially configured as 0.000125 and decayed by 10 at epochs 90 and 120. For the RM-CNN, the momentum and weight decay were configured as 0.9 and 0.0001, respectively. The learning rate was set as 0.0002 for the first 200,000 iterations and changed as 0.00002 for the last 100,000 iterations. For the SAS-Net, a total of 90,000 iterations were trained with the weight decay and momentum of 0.001 and 0.9. The learning rate was initially configured as 0.01 and decayed by 10 at iterations 60,000 and 82,500. For the OGA-Net, the anchor generation subnetwork and the object detection subnetwork were separately trained for 1000 and 800 epochs with the initial learning rate of 0.01 and a decayed learning rate of 0.001.

Likewise, the object detection results of these models were also evaluated based on the precision, recall, and F_1-score metrics and reported in Table 1, as well as the processing speeds of these models measured by the number of image patches being processed each second on a GPU. As Table 1 reports, the RM-CNN, GTGCF-Net, and OGA-Net showed superior performances than the other models. In contrast, the SARA-Net and the CBDA-Net achieved relatively lower accuracies than the other models. Overall speaking, depending on the pre-generated dense region proposals, the two-stage object detection models (i.e., the RM-CNN, SAS-Net, and OGA-Net) behaved better than the one-stage object detection models (i.e., the SARA-Net and CBDA-Net). However, the one-stage model GTGCF-Net also achieved compatible performance with the two-stage model RM-CNN, even higher performance than the two-stage models SAS-Net and OGA-Net. Therefore, we can conclude that, without the assistance of dense region proposals for pre-locating candidate object regions, the one-stage models can also obtain competitive accuracy by designing powerful and advantageous feature extraction network architectures or highly effective object identification and regression techniques. In addition, we found that the anchor-free models also performed as effectively as the anchor-based models. For example, the one-stage anchor-free model CBDA-Net showed a compatible performance with the one-stage anchor-based model SARA-Net. Even, the two-stage anchor-free model OGA-Net demonstrated higher quality than the two-stage anchor-based model SAS-Net. Thus, it indicated that, without the assistance of anchors for object regression, the anchor-free models can be still on a par with the anchor-based models in providing promising object detection accuracies. However, compared with the above models, our proposed SAHR-CapsNet showed distinctly advantageous performance owing to the novel HR-CapsNet formulation, as well as the effective feature attention mechanism. For qualitative comparisons, Fig. 9 presents some sample object detection results obtained by these models. Specifically, as shown by Fig. 9(c)-(h), some ships of extremely small sizes and some ships parallelly and closely distributed were not successfully detected by the compared models. In contrast, all the ships of different conditions were correctly recognized by the proposed SAHR-CapsNet. Through comparative analyses, we confirmed that the SAHR-CapsNet proposed in this paper offered a reliable and highly effective solution to oriented geospatial object detection tasks.

5. Conclusion

This paper has proposed an effective two-stage anchor-free model, named SAHR-CapsNet, for arbitrarily-oriented geospatial object

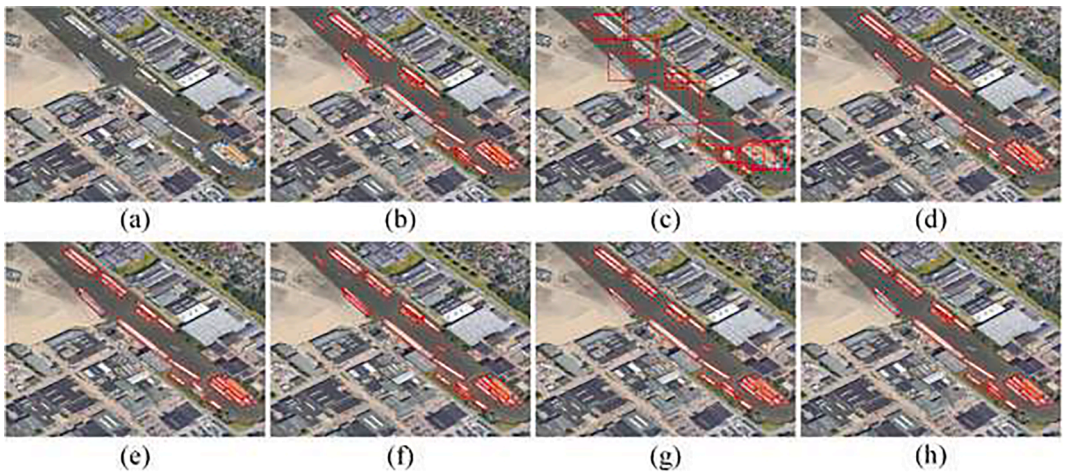

Fig. 9. Sample geospatial object detection results obtained by different models. (a) Test image, (b) the proposed SAHR-CapsNet, (c) GTGCF-Net, (d) SARA-Net, (e) CBDA-Net, (f) RM-CNN, (g) SAS-Net, and (h) OGA-Net.

detection from remote sensing images. The SAHR-CapsNet involved a feature extraction backbone network, a sparse anchoring network, and a multi-category object detection network. Specifically, formulated with a multibranch HR-CapsNet architecture boosted by multiscale feature propagation and fusion, the backbone network can extract semantically strong and spatially accurate feature semantics at multiple scales, thereby favoring the detection of varying-size objects. Designed with the capsule-based ESA module for focusing on the target-specific spatial features and the informative channel features, the feature encoding capability and robustness are significantly promoted, thereby effectively improving the region proposal quality and the object identification accuracy. Constructed with the SAN for producing a fixed quantity of sparse, high-quality region proposals, the SAHR-CapsNet can avoid the predesign and dense deployment of task-oriented anchors and reduce the computation overhead in anchor matching and regression, thereby well accelerating the processing efficiency. Quantitative examinations on two challenging remote sensing image datasets showed that a competitive average accuracy with the precision of 97.59%, the recall of 93.60%, and the F_1-score of 95.55%, respectively, was achieved on the recognition of geospatial objects with varied self-conditions in diverse environmental scenarios. In addition, comparative analyses also demonstrated the promising applicability and advantageous performance of the proposed SAHR-CapsNet for oriented geospatial object detection applications.

Funding

This work was supported by the National Natural Science Foundation of China [grant numbers 62076107, 51975239]; the Six Talent Peaks Project in Jiangsu Province [grant number XYDXX-098]; the Natural Science Research Project of Higher Education Institutions of Jiangsu Province [grant number 18KJB416002]; the Qing Lan Project of Jiangsu Province; and the Opening Project of Henan Engineering Laboratory of Photoelectric Sensor and Intelligent Measurement and Control [grant number HELPSIMC-2020–002].

CRediT authorship contribution statement

Yongtao Yu: Conceptualization, Funding acquisition, Methodology, Writing – original draft. **Jun Wang:** Investigation, Software. **Hao Qiang:** Funding acquisition, Writing – original draft. **Mingxin Jiang:** Funding acquisition, Validation, Visualization. **E Tang:** Investigation, Software. **Changhui Yu:** Methodology, Writing – original draft. **Yongjun Zhang:** Methodology, Writing – review & editing. **Jonathan Li:** Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Chen, X., Lin, J., Xiang, S., Pan, C.-H., 2020a. Detecting maneuvering target accurately based on a two-phase approach from remote sensing imagery. IEEE Geosci. Remote Sens. Lett. 17 (5), 849–853.
Chen, J., Wan, L.i., Zhu, J., Xu, G., Deng, M., 2020b. Multi-scale spatial and channel-wise attention for improving object detection in remote sensing imagery. IEEE Geosci. Remote Sens. Lett. 17 (4), 681–685.
Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. ISPRS J. Photogramm. Remote Sens. 117, 11–28.
Cheng, G., Si, Y., Hong, H., Yao, X., Guo, L., 2021. Cross-scale feature fusion for object detection in optical remote sensing images. IEEE Geosci. Remote Sens. Lett. 18 (3), 431–435.
Courtrai, L., Pham, M.T., Lefèvre, S., 2020. Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks. Remote Sens. 12 (19), 3152.
Fang, Z., Ren, J., Sun, H., Marshall, S., Han, J., Zhao, H., 2020. SAFDet: A semi-anchor-free detector for effective detection of oriented objects in aerial images. Remote Sens. 12 (19), 3225.
Fu, K., Chang, Z., Zhang, Y., Xu, G., Zhang, K., Sun, X., 2020. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. ISPRS J. Photogramm. Remote Sens. 161, 294–308.
Girshick, R., 2015. Fast R-CNN. In: Proc. IEEE Int. Conf. Comput. Vis., Santiago, Chile, pp. 1440-1448.
Hou, J.B., Zhu, X., Yin, X.C., 2021. Self-adaptive aspect ratio anchor for oriented object detection in remote sensing images. Remote Sens. 13 (7), 1318.
Jiang, S., Yao, W., Wong, M.S., Li, G., Hong, Z., Kuc, T.-Y., Tong, X., 2020. An optimized deep neural network detecting small and narrow rectangular objects in google earth images. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 13, 1068–1081.
Jiang, X., Liu, W., Zhang, Y., Liu, J., Li, S., Lin, J., 2021. Spectral-spatial hyperspectral image classification using dual-channel capsule networks. IEEE Geosci. Remote Sens. Lett. 18 (6), 1094–1098.
Jin, R., Lin, D., 2020. Adaptive anchor for fast object detection in aerial image. IEEE Geosci. Remote Sens. Lett. 17 (5), 839–843.
Lei, J., Luo, X., Fang, L., Wang, M., Gu, Y., 2020. Region-enhanced convolutional neural network for object detection in remote sensing images. IEEE Trans. Geosci. Remote Sens. 58 (8), 5693–5702.
Li, Q., Mou, L., Xu, Q., Zhang, Y., Zhu, X.X., 2019. R^3-Net: A deep network for multioriented vehicle detection in aerial images and videos. IEEE Trans. Geosci. Remote Sens. 57 (7), 5028–5042.
Li, K.e., Wan, G., Cheng, G., Meng, L., Han, J., 2020. Object detection in optical remote sensing images: A survey and a new benchmark. ISPRS J. Photogramm. Remote Sens. 159, 296–307.
Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: Proc. IEEE Int. Conf. Comput. Vis., Venice, Italy, pp. 2999-3007.
Liu, Q., Xiang, X., Yang, Z., Hu, Y.u., Hong, Y., 2021a. Arbitrary direction ship detection in remote-sensing images based on multitask learning and multiregion feature fusion. IEEE Trans. Geosci. Remote Sens. 59 (2), 1553–1564.
Liu, S., Zhang, L., Lu, H., He, Y., 2021b. Center-boundary dual attention for oriented object detection in remote sensing images. IEEE Trans. Geosci. Remote Sens., early access, doi: 10.1109/TGRS.2021.3069056.

Lu, X., Zhong, Y., Zheng, Z., Zhang, L., 2021. GAMSNet: Globally aware road detection network with multi-scale residual learning. ISPRS J. Photogramm. Remote Sens. 175, 340–352.
Ma, L., Liu, Y.u., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS J. Photogramm. Remote Sens. 152, 166–177.
Nie, X., Duan, M., Ding, H., Hu, B., Wong, E.K., 2020. Attention mask R-CNN for ship detection and segmentation from remote sensing images. IEEE Access 8, 9325–9334.
Pham, M.T., Courtrai, L., Friguet, C., Lefèvre, S., Baussard, A., 2020. YOLO-fine: One-stage detector of small objects under various backgrounds in remote sensing images. Remote Sens. 12 (15), 2501.
Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., Rodrigo, R., 2019. DeepCaps: Going deeper with capsule networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., Long Beach, USA, pp. 10725-10733.
Ren, Y., Yu, Y., Guan, H., 2020. DA-CapsUNet: A dual-attention capsule U-Net for road extraction from remote sensing imagery. Remote Sens. 12 (18), 2866.
Rezatofighi, H., Tsoi, N., Gwak, J.Y., Sadeghian, A., Reid, I., Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog, Long Beach, USA, pp. 658–666.
Rishikeshan, C.A., Ramesh, H., 2018. An automated mathematical morphology driven algorithm for water body extraction from remotely sensed images. ISPRS J. Photogramm. Remote Sens. 146, 11–21.
Sabour, S., Frosst, N., Hinton, G.E., 2017. Dynamic routing between capsules. In: Proc. Conf. Neural Inform. Process. Syst., Long Beach, USA, pp. 1-11.
Sagan, V., Maimaitijiang, M., Bhadra, S., Maimaitiyiming, M., Brown, D.R., Sidike, P., Fritschi, F.B., 2021. Field-scale crop yield prediction using multi-temporal WorldView-3 and PlanetScope satellite data and deep learning. ISPRS J. Photogramm. Remote Sens. 174, 265–281.
Shi, G., Zhang, J., Liu, J., Zhang, C., Zhou, C., Yang, S., 2021. Global context-augmented objection detection in VHR optical remote sensing images. IEEE Trans. Geosci. Remote Sens., early access, doi: 10.1109/TGRS.2020.3043252.
Shin, S.J., Kim, S., Kim, Y., Kim, S., 2020. Hierarchical multi-label object detection framework for remote sensing images. Remote Sens. 12 (17), 2734.
Sun, X., Wang, P., Wang, C., Liu, Y., Fu, K., 2021. PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 173, 50–65.
Sun, K., Xiao, B., Liu, D., Wang, J., 2019. Deep high-resolution representation learning for human pose estimation. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog., Long Beach, USA, pp. 5693-5703.
Tian, S., Kang, L., Xing, X., Li, Z., Zhao, L., Fan, C., Zhang, Y.e., 2021. Siamese graph embedding network for object detection in remote sensing images. IEEE Geosci. Remote Sens. Lett. 18 (4), 602–606.
Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog., Salt Lake City, USA, pp. 7794-7803.
Wang, X., Li, G., Plaza, A., He, Y., 2021a. Ship detection in SAR images via enhanced nonnegative sparse locality-representation of fisher vectors. IEEE Trans. Geosci. Remote Sens., early access, doi: 10.1109/TGRS.2020.3042506.
Wang, C., Shi, J., Yang, X., Zhou, Y., Wei, S., Li, L., Zhang, X., 2019. Geospatial object detection via deconvolutional region proposal network. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 12 (8), 3014–3027.
Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhang, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B., 2021b. Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell., early access, doi: 10.1109/TPAMI.2020.2983686.
Wang, M., Sun, Z., Xu, G., Ma, H., Yang, S., Wang, W., 2020. Deep hash assisted network for object detection in remote sensing images. IEEE Access 8, 180370–180378.
Wang, J., Wang, Y., Wu, Y., Zhang, K., Wang, Q., 2021c. FRPNet: A feature-reflowing pyramid network for object detection of remote sensing images. IEEE Geosci. Remote Sens. Lett., early access, doi: 10.1109/LGRS.2020.3040308.
Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2018. DOTA: A large-scale dataset for object detection in aerial images. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., Salt Lake City, USA, pp. 3974-3983.
Xiao, Z., Wang, K., Wan, Q., Tan, X., Xu, C., Xia, F., 2021. A^2S-Det: Efficiency anchor matching in aerial image oriented object detection. Remote Sens. 13 (1), 73.
Xu, Q., Chen, K., Sun, X., Zhang, Y., Li, H., Xu, G., 2021a. Pseudo-Siamese capsule network for aerial remote sensing images change detection. IEEE Geosci. Remote Sens. Lett., early access, doi: 10.1109/LGRS.2020.3022512.
Xu, C., Li, C., Cui, Z., Zhang, T., Yang, J., 2020. Hierarchical semantic propagation for object detection in remote sensing imagery. IEEE Trans. Geosci. Remote Sens. 58 (6), 4353–4364.
Xu, F., Somers, B., 2021. Unmixing-based Sentinel-2 downscaling for urban land cover mapping. ISPRS J. Photogramm. Remote Sens. 171, 133–154.
Yang, Y., Tang, X., Cheung, Y.M., Zhang, X., Liu, F., Ma, J., Jiao, L., 2021. AR2Det: An accurate and real-time rotational one-stage ship detector in remote sensing images. IEEE Trans. Geosci. Remote Sens., early access, doi: 10.1109/TGRS.2021.3092433.
Yao, X., Feng, X., Han, J., Cheng, G., Guo, L., 2021a. Automatic weakly supervised object detection from high spatial resolution remote sensing images via dynamic curriculum learning. IEEE Trans. Geosci. Remote Sens. 59 (1), 675–685.
Yao, Q., Hu, X., Lei, H., 2021b. Multiscale convolutional neural networks for geospatial object detection in VHR satellite images. IEEE Geosci. Remote Sens. Lett. 18 (1), 23–27.
Ye, X., Xiong, F., Lu, J., Zhou, J., Qian, Y., 2020. F^3-Net: Feature fusion and filtration network for object detection in optical remote sensing images. Remote Sens. 12 (24), 4027.
Yu, Y., Gu, T., Guan, H., Li, D., Jin, S., 2019. Vehicle detection from high-resolution remote sensing imagery using convolutional capsule networks. IEEE Geosci. Remote Sens. Lett. 16 (12), 1894–1898.
Yu, Y., Guan, H., Li, D., Gu, T., Tang, E., Li, A., 2020. Orientation guided anchoring for geospatial object detection from remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 160, 67–82.
Zhang, S., He, G., Chen, H.-B., Jing, N., Wang, Q., 2019. Scale adaptive proposal network for object detection in remote sensing images. IEEE Geosci. Remote Sens. Lett. 16 (6), 864–868.
Zhang, R., Shao, Z., Huang, X., Wang, J., Li, D., 2020. Object detection in UAV images via global density fused convolutional network. Remote Sens. 12 (19), 3140.
Zheng, Z., Zhong, Y., Ma, A., Han, X., Zhao, J., Liu, Y., Zhang, L., 2020. HyNet: Hyper-scale object detection network for multiple spatial resolution remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 166, 1–14.
Zhong, Y., Han, X., Zhang, L., 2018. Multi-class geospatial object detection based on a position-sensitive balancing framework for high spatial resolution remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 138, 281–294.