Accurate Object Detection & Instance Segmentation of Remote Sensing, Imagery Using Cascade Mask R-CNN With HRNet Backbone

SUBMISSION DATE / POSTED DATE

17-10-2022 / 24-10-2022

CITATION

Kumar, Durga (2022): Accurate Object Detection & Instance Segmentation of Remote Sensing, Imagery Using Cascade Mask R-CNN With HRNet Backbone. TechRxiv. Preprint.
https://doi.org/10.36227/techrxiv.21345639.v1

DOI

10.36227/techrxiv.21345639.v1

Accurate Object Detection & Instance Segmentation of Remote Sensing, Imagery Using Cascade Mask R-CNN With HRNet Backbone.

Durga Kumar, *Xiaoling Zhang*

***Abstract*—This letter presents a Cascade Mask R-CNN framework with an HRNet backbone for geospatial object detection and instance segmentation from high-resolution remote sensing imagery. A multi-stage object detection & instance segmentation architecture, the Cascade Mask R-CNN, framed of a chronological sequence of detectors prepared with enhancing IoU thresholds, is proposed to address these problems. The detectors are trained sequentially, using the output of a detector as a training set for the next. The HRNet (Backbone) acquires a fresh high-resolution feature pyramid network (HRFPN) to take the total reward of the feature maps of high-resolution and low-resolution convolutions for remote sensing images. In this scheme, the HRFPN connects high-to-low resolution sub-networks in parallel and can maintain high resolution and repel switch over the entropy throughout resolutions. Then, we enclose the Microsoft Common Objects in Context (COCO) evaluation metrics, which allows not only the more prominent caliber evaluation metrics average precision (AP) for more precise bounding box regression, but also the evaluation metrics for small, medium, and large targets, to exactly evaluate the detection & instance segmentation performance of our method. The proposed method is compared with other backbones (ResNet, ResNext) based Cascade Mask R-CNN methods and with different depths of backbone & epochs. Experiments are conducted on a series of remote sensing images (NWPU VHR-10). Detection & instance segmentation results demonstrate that our method provides better performance (significant improvement in accuracy) in terms of average precision (AP).**

***Index Terms*—HRNet, HRFPN, Object Detection, Instance Segmentation, Cascade, Bounding Box Regression.**

I. Introduction

THE development of high-resolution remote sensing imagery now provides images of up to a few centimeters and submeter spatial resolution, respectively, by airborne sensors and satellite sensors. The automatic extraction of information from high-resolution images has in recent years been a very active topic. The object detection & instance segmentation method based on deep learning has been greatly improved compared with the traditional method, however, the accuracy will be significantly reduced when the universal object detection & instance segmentation method is used for remote sensing images [1]. The dispute in this job comes from the varying orientations and scales of objects in VHR remote sensing images because they are taken from either airplanes or satellites. Furthermore, while high-resolution images bring in fine details of ground objects, they also produce complex and cluttered backgrounds [2], [3].

In the field of computer vision, object detection is one of the most central and ambitious problems. In 2013, a discovery was made by Girshick, who suggested a region-based CNN (R-CNN) detector that betters mean average precision (mAP) by more than 50% relative to the former best result. Since then, considerable efforts have been made to improve the detector along the R-CNN based pipeline [4]–[6]. The best-improved detector is Faster R-CNN [7], which comprises of a region proposal network (RPN) for foreseeing applicant areas, an object detection network for classifying object proposals and refining their spatial areas, and bounding box regression, to solve localization. After that, a new kind of detector architecture denoted as Cascade R-CNN [8] was discovered. This architecture is a multi-stage extension of the R-CNN, where the detector stages deeper into the cascade is sequentially more selective against close false positives. The cascade of R-CNN [8] stages is trained sequentially, using the output of one stage to train the next. After the original publication, the Cascade R-CNN [8] has been successfully reproduced within many different codebases, including the popular MMDetection [9]. After that, the extended version of Cascade R-CNN to instance segmentation, by adding a mask head to the cascade, denoted as Cascade Mask R-CNN [8], [9]. This shows the significant improvements over the popular Mask R-CNN [10]. Cascade R-CNN is simpatico with lots of complementary sweetenings proposed in the detection and instance segmentation literature.

In this work, we define thresholding the intersection over union (IoU) score between candidate and ground truth bounding boxes. While the threshold is typically set at the value of $u = 0.5$. The hypotheses that nearly the public would take close false positives frequently pass the $IoU \geq 0.5$ test. While training with $IoU \leq 0.5$ criteria, they make it difficult to train detectors that can effectively reject close false positives. The principle distinction is that the resampling performed by the Cascade R-CNN doesn't mean to mining hard negatives. Rather, by modifying bounding boxes, each stage means to locate a decent arrangement of close false positives for preparing for the next stage [8].

II. Object Detection & Instance Segmentation In 'NWPU VHR-10' Data Set

In this section, we discuss the challenges of high quality object detection & instance segmentation in remote sensing images.

*1) **Object Detection**:* While the estimation proposed in this work can be applied to different detector architectures,

we center around the mainstream two-stage architectures of the Faster R-CNN [7]. The first stage is a proposition subnetwork, in which the whole image is handled by a backbone network (HRNet) [11], [12], and a proposition head $\left(''\mathrm{H}_0''\right)$ is applied to deliver primer detection hypotheses, known as object proposition. In the subsequent stage, these hypotheses are prepared by a region-of-interest $(''ROI'')$ detection subnetwork $\left(''\mathrm{H}_1''\right)$, indicated as a detection head. A last classification score $\left(''\mathrm{CS}''\right)$, and a bounding box $\left(''\mathrm{BB}''\right)$, are attributed per hypothesis. The whole detector is found to start to finish, utilizing a perform multi-task loss with the bounding box regression and classification components [8].

*a) **Bounding Box Regression**:* A bounding box is denoted by $\left(''\mathbf{bb}''\right)$, which

$$\mathbf{bb} = (bb_x, bb_y, bb_z, bb_w) \tag{1}$$

contains the four directions X, Y, Z, and W of an image patch P. Bounding box regression plans to return a candidate bounding box $\left(''\mathbf{bb}''\right)$ into a ground truth bounding box $(''\mathbf{gt}'')$, utilizing a regressor $f(\mathbf{P}, \mathbf{bb})$. This is found out from a training set $(\mathbf{gt}_i, \mathbf{bb}_i)$, by limiting the hazard

$$\mathcal{H}_{loc}[f] = \sum_i L_{loc}\left(f\left(\mathbf{P}_i, \mathbf{bb}_i\right), \mathbf{gt}_i\right) \tag{2}$$

As described in Fast R-CNN [13],

$$L_{loc}(\mathbf{a}, \mathbf{bb}) = \sum_{i \in \{x,y,z,w\}} \mathit{smooth}_{L_1}\left(a_i - bb_i\right) \tag{3}$$

where

$$\mathit{smooth}_{L_1}(y) = \begin{cases} 0.5y^2, & |y| < 1 \\ |y| - 0.5, & \mathit{otherwise,} \end{cases} \tag{4}$$

is the smooth L_1 loss function. To urge invariance to scale and area, smooth $_{L_1}$ works on the separation vector

$$\Delta = (\delta_x, \delta_y, \delta_z, \delta_w)$$

characterized by

$$\begin{array}{ll} \delta_x = \left(gt_x - bb_x\right)/bb_z, & \delta_y = \left(gt_y - bb_y\right)/bb_w \\ \delta_z = \log\left(gt_z/bb_z\right), & \delta_w = \log\left(gt_w/bb_w\right). \end{array} \tag{5}$$

Since bounding box regression usually performs minor adjustments on b, the numerical values of (5) may be very small. In general, this makes the loss of regression much less than the loss of classification. To improve the efficiency of multi-task learning Δ is standardized by their mean and variance, just like δ_x is substituted by

$$\delta_x' = \frac{\delta_x - \mu_x}{\sigma_x} \tag{6}$$

This is used extensively in literature [7], [8], [10], [14]–[16].

*b) **Classification**:* The classifier is a function $h(P)$ that allocates a picture patch P to one of the $1 + M$ classes (In NWPU VHR-10 data set total number of object classes are 10.), where class 0 contains the backcloth what's more, the rest of the classes the object to identify. $h(P)$ is a $1 + M$ dimensional estimate of the back appropriation over classes, for example

$$h_k(P) = p(y = k|P) \tag{7}$$

Where y is the class name. Given a preparation set (P_i, yi), it is found out by limiting the order hazard

$$\mathcal{H}_{cls}[h] = \sum_i L_{cls}\left(h\left(P_i\right), y_i\right) \tag{8}$$

Where

$$L_{cls}(h(P), y) = -\log h_y(P) \tag{9}$$

is the cross-entropy loss.

*c) **Quality of Object Detection**:* Consider a ground truth object of the bounding box $(''\mathrm{gt}'')$ related with class name y and a detection hypothesis P of bounding box $\left(''\mathrm{bb}''\right)$. Since a b usually includes an object and some amount of background, it very well may be hard to decide whether detection is right or not. This is normally addressed by the intersection over union (IoU) metric

$$\mathit{IoU}(\mathbf{bb}, \mathbf{gt}) = \frac{\mathbf{bb} \cap \mathbf{gt}}{\mathbf{bb} \cup \mathbf{gt}} \tag{10}$$

On the off chance that the IoU is over a limit u, the patch is viewed as a case of the class of the object of bounding box $(''\mathrm{gt}'')$ and meant "positive". In this way, the class label of a hypothesis P is a component of u,

$$y_u = \begin{cases} y, & \mathit{IoU}(\mathbf{bb}, \mathbf{gt}) \geq u \\ 0, & \mathit{otherwise.} \end{cases} \tag{11}$$

On the off chance that the IoU doesn't surpass the limit for any object, P is allocated to the background and indicated $\left(''\mathrm{negative}''\right)$. Although there is no compelling reason to characterize $\left(''\mathrm{positive/negative}''\right)$ models for the bounding box relapse task, an IoU limit u is additionally required to choose the set of samples used to train the regressor.

$$\mathcal{G} = \left\{\left(\mathbf{gt}_i, \mathbf{bb}_i\right) | \mathit{IoU}\left(\mathbf{bb}_i, \mathbf{gt}_i\right) \geq u\right\} \tag{12}$$

At the point when u is high, positives contain less background be that as it may, it is hard to gather huge positive preparing sets. At the point when u is low, more extravagant and progressively different positive preparing sets are conceivable, however, the prepared detector has a minimal motivating force to dismiss close false positives.

*2) **Instance segmentation**:* Recently, end-to-end trainable instance segmentation approaches based on CNNs have been emerging. There are mainly two approaches; proposal-based and clustering-based approaches [17]. It intends to portend pixel-level segmentation for each instance, in addition to deciding its object class. This is more troublesome than object recognition, which as it were predicts a bounding box (in addition to class) per instance. The model is prepared on the training set and evaluated on the validation set during the training procedure. After the training, instance segmentation is referenced on the test set, and assessed by utilizing the AP metric. Then, we introduce the Microsoft Common Objects in Context (COCO) evaluation metrics, which provide not only higher quality evaluation metrics average precision (AP) for more accurate bounding box regression, but also the evaluation metrics for small, medium, and large targets, to precisely evaluate the detection and segmentation performance of our method [18], [19].

III. Proposed Method

In this section, the proposed approach (Cascade Mask R-CNN With HRNet) will be elaborated on in detail.

*1) **The Background of HRNet**:* Visual recognition generally consists of three major research problems: image-level (image classification), region-level (object detection), and pixel-level (including image segmentation, and human pose estimation). Therefore, to compensate for the loss of spatial precision, there are two main lines for computing high-resolution. One is to recover high-resolution representations from low-resolution representations as shown in Figure 1, by greenline and the other one is to maintain high-resolution representations through high-resolution convolutions and strengthen the representations with parallel low-resolution convolutions, e.g., high-resolution network (HRNet) [11], [12], [18], as shown by the black line in Figure 1.

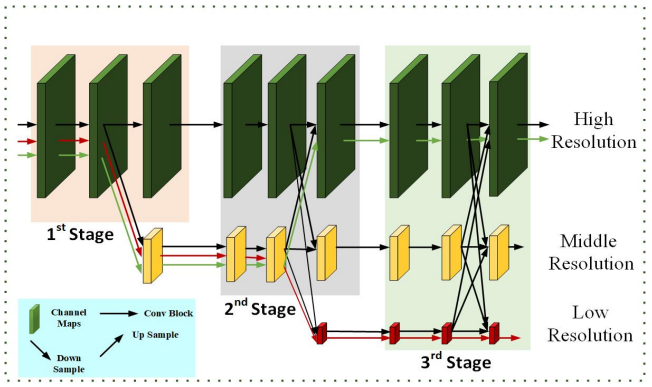

Fig. 1. The architecture of representation learning. The red line path indicates the low-resolution representation learning network, and the black line and the green line paths indicate the high-resolution representation recovering network.

*2) **Detailed Description of the Network Architecture (Cascade Mask R-CNN)**:* In the Mask R-CNN, as appeared in Fig. 2 (a), the segmentation branch is embedded in parallel to the detection branch. Be that as it may, the Cascade R-CNN has different detection branches. This raises two questions:

a) Where to include the segmentation branch.

b) What number of segmentation branches to include.

We think about three systems for mask prediction in the Cascade R-CNN. The initial two systems address the main inquiry, including a solitary mask prediction head at either the first or the last phase of the Cascade R-CNN, as appeared in Fig. 2 (b) and (c), individually. The third procedure addresses the subsequent inquiry, adding a division branch to each cascade stage, as appeared in Fig. 2 (d) maximizes the diversity of samples used to learn the mask prediction task.

IV. Experimental Result

For the sake of fair comparison, the experiments and comparisons are implemented on mmdetection [9], which is a well-known open-source deep learning framework, and executed on a personal computer (PC) with an Intel(R) i7-8700 CPU @3.20GHz, NVIDIA GTX-1080 GPU (8 GB memory), and 64 GB RAM. The PC operating system is a 64-bit Ubuntu 18.04 with Python 3.7, PyTorch 1.1, CUDA 10.1, CUDNN 7.0.4, and NCCL 2.1.15 Software environment.

A. Dataset Description (NWPU VHR-10)

NWPU VHR-10 geospatial object detection data sets [3], [20], [21] are used in the experiments and were randomly divided into 70% for training, and 30% for testing. NWPU VHR-10 data set contains ten man-made objects like airplanes, ships, storage tanks, baseball diamonds, tennis courts, basketball courts, ground track fields, harbors, bridges, and vehicles. This dataset contains a total of 800 VHR optical remote sensing images, where 715 color images were acquired from Google Earth with spatial resolution ranging from 0.5 to 2 m and 85 pan-sharpened color infrared images were acquired from Vaihingen data with a spatial resolution of 0.08 m.

B. Evaluation Metrics

To quantitatively evaluate the performance and robustness of the proposed frameworks, the following metrics are widely used: intersection over union (IoU), precision, recall, and mean average precision (mAP). As shown in formula (10), IoU is the overlap rate of the predicted bounding box and ground-truth generated by the model. The calculation formulas of precision and recall are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

where TP (True Positives) indicates the number of correctly detected objects; FN (False Negatives) denotes the number of non-detected or missed objects; and FP (False Positives or false alarms) represents the number of incorrectly detected objects. For single-class object detection, the mean average precision (mAP) is defined by [22]:

$$mAP = \int_0^1 P(r)dr \tag{15}$$

where r represents recall and P(r) denotes the precision value that recall = r corresponds to. For object detection, the larger the value of mAP gives the better detection performance of the object. Be that as it may, the mean average precision (mAP) doesn't completely mirror the presentation of an object detection framework. Contrasted with the mAP of PascalVOC [22], the Microsoft Common Objects in Setting (COCO) [19] incorporates not just the better assessment measurements, for example, AP, AP_{50}, and AP_{75} for progressively exact bounding box relapse, yet besides the assessment measurements AP_L, AP_M, and AP_S for large, medium, and small objects. Along these lines, COCO measurements are increasingly goal and exhaustive for object detection tasks. In general, mAP is a default metric of precision in the PascalVOC competition [22], which is the same as the AP_{50} metric in the MS COCO competition. Besides, COCO metrics are standard and widely used evaluation metrics in object detection tasks, As shown in formula (10). For NWPU VHR-10 object detection, we leverage the standard COCO [19] metrics to quantitatively evaluate

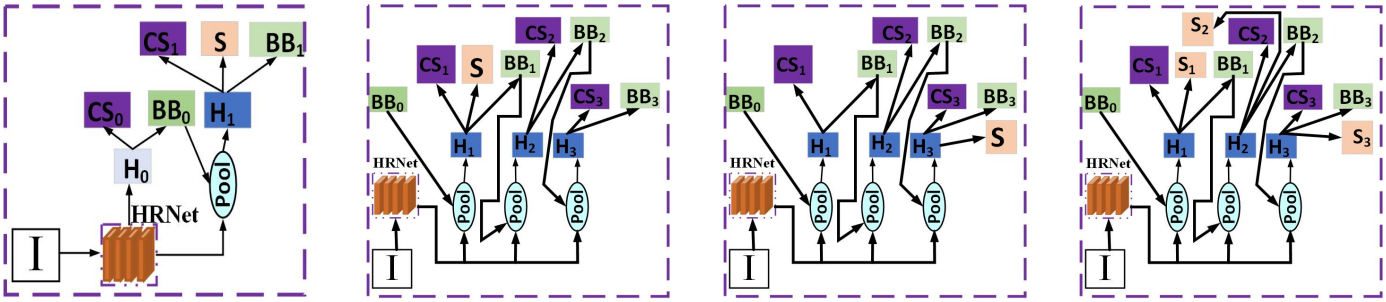

Fig. 2. (a) Architectures of the Mask R-CNN. (b)-(c) Cascade Mask R-CNN strategies for instance segmentation with single head. (d) Cascade Mask R-CNN strategies for instance segmentation with triple head. "S" denotes a segmentation branch. "BB" denotes a bounding box. "CS" denotes a classification score. Note that segmentations branches do not necessarily share heads with the detection branch.

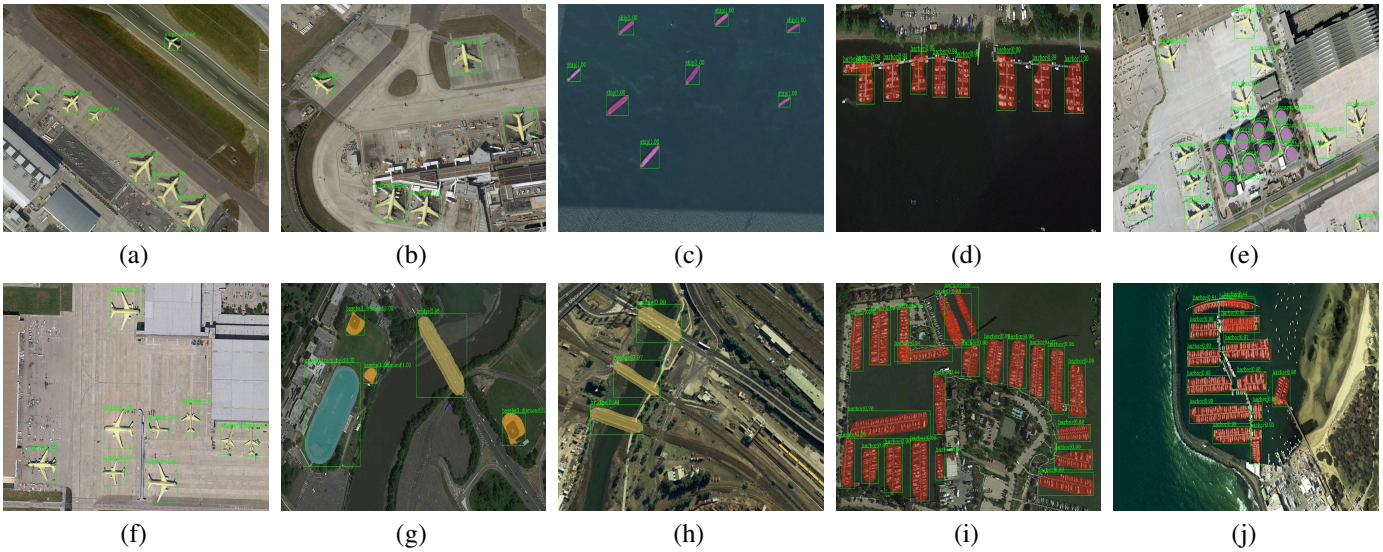

Fig. 3. Results of different scene object detection and instance segmentation in NWPU VHR-10 dataset (a)-(b) airplane (c) ship (d) harbor (e) airplane and harbor (f) airplane (g) ground track field, baseball diamond and bridge (h) three bridges (i)-(j) harbor.

TABLE I
DETECTION & SEGMENTATION PERFORMANCE OF DIFFERENT FRAMEWORK & COMPARISON WITH PROPOSED METHOD

		Detection Performance								Segmentation Performance						
Backbone	**Epochs**	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR	**Task/Sec.**	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR
ResNet-50-FPN	12e	64.6	89.4	76.3	59.2	65.8	58.2	25.6	8.6	59.2	88.7	64.4	45.9	59.3	58.6	24.7
	20e	67.8	90.3	80.1	60.4	68.5	65.6	26.5	9.7	62.3	89.8	69.6	47.4	62.2	65.5	25.6
ResNet-101-FPN	12e	65.5	90.4	77.2	57.8	66.5	63.3	26.1	7.6	59.3	89.6	63.9	44.7	58.7	65.3	25.1
	20e	69.2	91.9	80.8	58.7	70.0	69.4	27.1	7.8	62.7	91.6	67.3	45.8	62.6	68.8	25.8
ResNeXt-101-32x4d-FPN	12e	67.9	90.9	80.7	59.7	69.5	64.7	26.7	6.3	62.0	91.4	66.8	46.6	61.5	68.1	26.0
	20e	69.9	92.1	81.3	59.9	71.2	68.4	27.3	6.3	64.3	91.9	70.1	47.2	64.7	68.5	26.3
ResNeXt-101-64x4d-FPN	12e	68.4	91.9	80.7	61.0	69.2	67.0	27.2	4.6	62.0	91.4	66.8	46.6	61.5	68.1	26.0
	20e	69.9	91.6	82.8	60.4	70.4	70.3	27.6	4.7	64.3	91.3	68.9	48.6	63.8	70.8	26.5
HRNetV2p-W18	20e	40.1	62.1	46.4	44.2	39.9	31.6	16.5	6.8	35.2	58.7	36.9	29.9	34.4	32.9	16.4
HRNetV2p-W32	**20e**	**70.9**	**92.5**	**80.8**	**62.2**	**71.7**	**69.2**	**27.7**	**5.6**	**65.1**	**91.9**	**71.5**	**49.5**	**64.7**	**69.8**	**26.6**

the performance of the proposed framework for detection & segmentation performance, including AP, AP_{50}, AP_{75}, AP_L, AP_M, AP_S as described in **Table I**. AP_{50} denotes the set threshold of IoU as 0.50; AP_{75} denotes that the threshold is set as 0.75; AP indicates that the threshold of IoU is set from 0.50 to 0.95, where the step size is 0.05; AP_S is set for small objects in which the area is smaller than 32^2; AP_M is set for medium objects in which the area is between 32^2 and 96^2; AP_L is set for large objects in which the area is bigger than 96^2. The larger the value of AP is, the more accurate the prediction results and the better the detection performance of the objects. For AP_{50}, when the IoU of the ground truth and the predicted box is greater than 0.5, the test case is predicted as an object. Therefore, with a higher IoU threshold, the bounding box regression will be better and the object is well covered by the predicted bounding box. So AP_{75} evaluates the

TABLE II
COCO DATASET OBJECT DETECTION EVALUATION METRICS [19]

Metrics	Metrics Meaning
AP	AP at IoU = 0.50: 0.05: 0.95
AP_{50}	AP at IoU = 0.50
AP_{75}	AP at IoU = 0.75
AP_S	AP for small objects: area $<32^2$
AP_M	AP for medium objects: 32^2 <area $<96^2$
AP_L	AP for large objects: area $>96^2$

accuracy of the bounding box regression better than AP_{50}.

C. Result Analysis

With the help of **Table II** we can easily understand the detection & segmentation performance is better with HRNetV2p-W32 (20e) backbone as compare to all other backbones like ResNet-50-FPN (12e, 20e), ResNet-101-FPN (12e, 20e), ResNeXt-101-32x4d-FPN (12e, 20e), ResNeXt-101-64x4d-FPN (12e, 20e), HRNetV2p-W18 (20e) etc. Detection parameter in terms of AP, AP_{50}, AP_{75}, AP_L, AP_M, AP_S with HRNetV2p-W32 backbone having 20 epochs are respectively 70.9%, 92.5%, 80.8%, 62.2%, 71.7%, 69.2%, 27.7% and the same parameters for segmentation performance are respectively 65.1%, 91.9%, 71.5%, 49.5%, 64.7%, 69.8%, 26.6%. The execution time is 5.6 task/second for proposed backbone. Results of different scene for object detection and instance segmentation in NWPU VHR-10 dataset is shown clearly in Fig. 3.

V. CONCLUSION

In this paper, we propose object detection & instance segmentation based on the Cascade Mask R-CNN framework with HRNet backbone in high-resolution NWPU VHR-10 images. The HRNet adopts a novel HRFPN to make full use of the feature maps of high-resolution and low-resolution convolutions for remote sensing imagery. In this way, the HRFPN connects high-to-low resolution subnetworks in parallel and can maintain the high-resolution.

Future work: our future work will focus to speed up object detection & instance segmentation algorithm for remote sensing imagery.

ACKNOWLEDGMENT

We would like to especially thank Hao Su, Shunjun Wei & Yuanyuan Zhou for helpful discussions and an effective understanding of Pytorch and different kinds of images.

REFERENCES

[1] X. Ying, Q. Wang, X. Li, M. Yu, H. Jiang, J. Gao, Z. Liu, and R. Yu, "Multi-attention object detection model in remote sensing images based on multi-scale," *IEEE Access*, vol. 7, pp. 94 508–94 519, 2019.

[2] C. Wang, X. Bai, S. Wang, J. Zhou, and P. Ren, "Multiscale visual attention networks for object detection in vhr remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 2, pp. 310–314, 2018.

[3] H. Su, S. Wei, M. Yan, C. Wang, J. Shi, and X. Zhang, "Object detection and instance segmentation in remote sensing imagery based on precise mask r-cnn," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 1454–1457.

[4] D. Kumar and X. Zhang, "Improving more instance segmentation and better object detection in remote sensing imagery based on cascade mask r-cnn," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 4672–4675.

[5] ——, "Ship detection based on faster r-cnn in sar imagery by anchor box optimization," in *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2019, pp. 1–6.

[6] D. Kumar, X. Zhang, H. Su, and S. Wei, "Accurate object detection based on faster r-cnn in remote sensing imagery," in *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*. IEEE, 2019, pp. 1–6.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] Z. Cai and N. Vasconcelos, "Cascade r-cnn: high quality object detection and instance segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1483–1498, 2019.

[9] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[11] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[12] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[13] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[14] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European conference on computer vision*. Springer, 2016, pp. 354–370.

[15] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[17] N. Inoue and T. Yamasaki, "Fast instance segmentation for line drawing vectorization," in *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2019, pp. 262–265.

[18] S. Wei, H. Su, J. Ming, C. Wang, M. Yan, D. Kumar, J. Shi, and X. Zhang, "Precise and robust ship detection for high-resolution sar imagery based on hr-sdnet," *Remote Sensing*, vol. 12, no. 1, p. 167, 2020.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016.

[21] A. Farooq, J. Hu, and X. Jia, "Efficient object proposals extraction for target detection in vhr remote sensing images," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017, pp. 3337–3340.

[22] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *arXiv preprint arXiv:1904.04514*, 2019.