TQR-Net: Tighter Quadrangle-Based Convolutional Neural Network for Dense Building Instance Localization in Remote Sensing Imagery

Kaiyu Jiang and Qingpeng Li(✉)

State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China
{kyjiang,liqingpeng}@buaa.edu.cn

Abstract. Building localization in remote sensing imagery (RSI) is widely applied in many geoscience and remote sensing areas. However, many existing methods cannot generate accurate building contours. In this paper, we propose an effective convolutional neural network (CNN) framework, Tighter Quadrangle Network (TQR-Net), to locate buildings with quadrangular contours in RSI. Here, TQR-Net can generate regular contours for each of building targets using a CNN branch which can predict tighter quadrangles in parallel. Then, we train and test TQR-Net on a large building dataset collected from Google Earth, and the experiment results demonstrate that the proposed method can generate high-quality building contours and significantly outperforms other CNN-based detectors.

Keywords: Deep learning · Convolutional neural network · Building instance localization · Remote sensing · Tighter quadrangle

1 Introduction

With the rapid development of spaceborne and airborne imaging technology, the high-resolution remote sensing imagery (RSI) can be more and more accessible to make the spatial structure, texture and other information of geographic objects abundant. Thus, automatic building localization can potentially achieve higher accuracy, which is helpful to many remote sensing applications, such as land planning, environment management and disaster assessment.

Therefore, developing automatic methods of building localization is a significant task. Over the past decades, many approaches have been proposed for automatic building localization. For example, in the early days, low-level handcrafted features were applied for feature extraction to locate buildings. Kim et al. [1] extracted the edge segments and detected possible building structures based on graph search strategy. Jung et al. [2] proposed a Hough transform-based method to extract the rectangular building roofs.

Y. Zhao et al. (Eds.): ICIG 2019, LNCS 11903, pp. 281–291, 2019.
https://doi.org/10.1007/978-3-030-34113-8_24

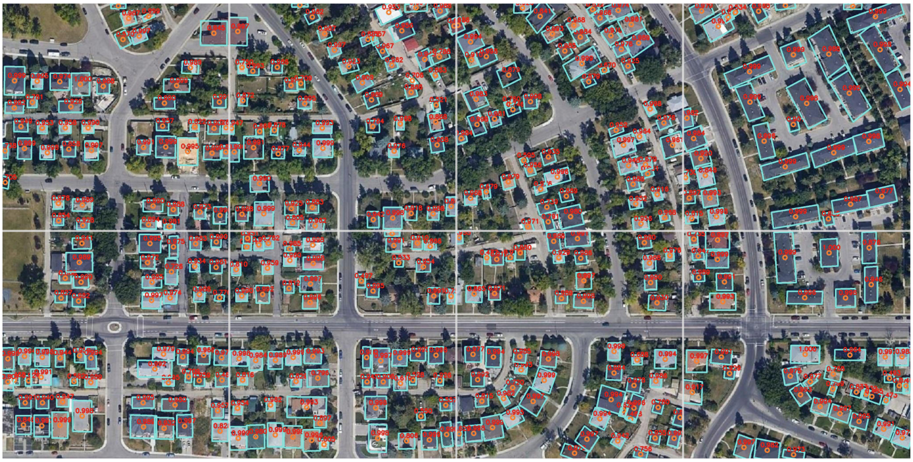
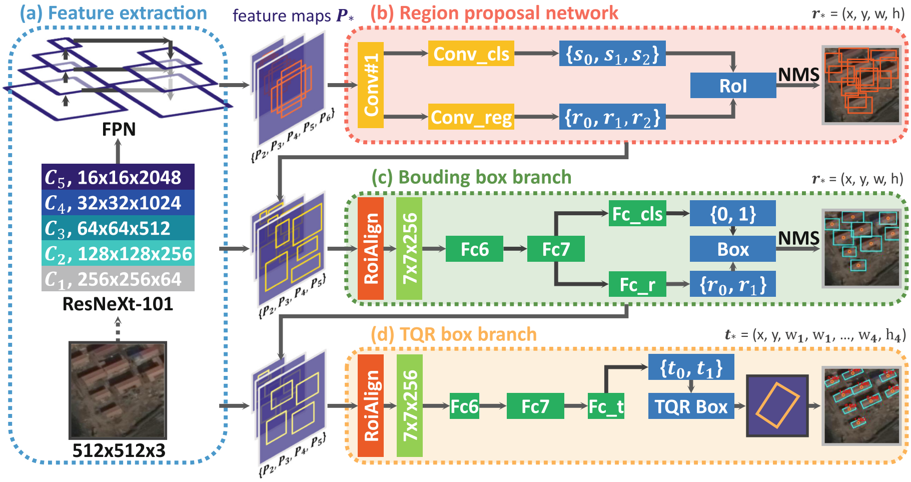


Fig. 1. Example of building localization results from TQR-Net in Google Earth image of Calgary, Alberta, Canada (51.05°N, 114.07°W).

Moreover, in order to obtain building contours, image segmentation can also be utilized to partition RSI into many regions and classify each pixel into a fixed set of categories [3], distinguishing buildings from their surrounding background. For example, Kampffmeyer et al. [4] combined different deep architectures including patch-base and pixel-to-pixel approaches, to achieve good accuracy for small object segmentation in urban remote sensing. Wu et al. [5] proposed a multi-constraint fully convolutional network to improve the performance of the U-Net model in building segmentation from aerial imagery. Troya-Galvis et al. [6] presented two different extensions of a collaborative framework called CoSC which outperform hybrid pixel-object oriented approach as well as a deep learning approach. Insufficiently, such methods can generate roughly building segmentation boundary, however, they are always irregular and can not differentiate building instances.

In recent five years, the CNN-based object detectors [7–10] have made a great improvement for detecting remotely sensed targets [11–17]. Consequently, the CNN-based building detectors have also made a breakthrough. For example, Zhang et al. [18] proposed a CNN-based detector using multi-scale saliency-based sliding window and improved non-maximum suppression (NMS) to detect suburban buildings. Li et al. [19] presented a cascaded CNN architecture utilizing Hough transform to guide CNN to extract mid-level features of the building. Chen et al. [20] proposed a two-stage CNN-based detector for multi-sized building localization, in which a multi-sized fusion region proposal network (RPN) and a novel dynamic weighting algorithm were used to generate and classify multi-sized region proposals, respectively. Although such object detection-based methods can classify individual buildings, they denote detection via rectangular bounding boxes and can not generate building contour. To tackle this problem,



Conv#1	3x3x256, str 1, pad 1	Fc6	(256x7x7)x1024	Fc_r	1024x8
Conv_cls	1x1x3, str 1, pad 1	Fc7	1024x1024	Fc_t	1024x20
Conv_reg	1x1x12, str 1, pad 1	Fc_cls	1024x2		

Fig. 2. The architecture of the proposed multi-stage TQR-Net is as follows: (a) Feature extraction stage generates a rich and multi-scale feature pyramid. (b) Region proposal network outputs a set of object proposals with objectness scores s_i (e.g., $i = 0, 1, 2$ denotes three aspect ratios). (c) Bounding box branch regresses rectangular bounding boxes of each pyramid level. (d) TQR box branch predicts quadrangle bounding boxes and obtains building contours.

some instance segmentation-based methods [21–23] can be adopted to detect buildings in RSI, but the generated contours are still irregular in the instance segmentation-based approaches.

As aforementioned, generally, there are two kinds of bounding boxes to locate building targets. One is rectangular, which cannot generate the contours of buildings. The other is polygonal, based on instance segmentation detectors (e.g., Mask R-CNN [10]), which can locate buildings via predicting their segmentation and polygonal contours. However, such polygonal contours are always inaccurate due to their uncertain nodes and irregular shapes.

In this paper, aiming to make a trade-off between these two kinds of bounding boxes, we propose to use quadrangular bounding boxes, which are generated by a tighter quadrangle-based convolutional neural network (TQR-Net) directly. Considering that most buildings are quadrilateral, we adopt quadrangular bounding boxes with four nodes, which can not only avoid irregular shapes but also keep certain structural restrictions.

Without bells and whistles, the experiment results prove that the proposed TQR-Net can improve the feature extraction domain of corner and contour in building targets with higher precision of building localization. Here, we give an example of localization results acquired by TQR-Net in Google Earth urban area image of Calgary is shown in Fig. 1.

2 Proposed Approach

As shown in Fig. 2, our method is based on a multi-stage region-based object detection framework. In this section, we will elaborate the proposed network in the subsections.

2.1 Multi-stage Region-Based TQR-Net

There are four main stages in TQR-Net, i.e., feature extraction, region proposal network, bounding box branch, and tighter quadrangle box branch, and we will detail each stage as follows.

Feature Extraction. A feature extraction network can extract features from the input image. Here we utilize ResNeXt-101 [24] for feature extraction, and such multi-scale feature maps are extracted on five levels, which can be defined as $\{C_1, C_2, C_3, C_4, C_5\}$. At each level, convolutional layers generate feature maps of the same size. In order to detect buildings in different scales, we use Feature Pyramid Network (FPN) [25] in the convolutional backbone which utilizes top-down lateral connections to build an in-network feature pyramid. The FPN can take $\{C_2, C_3, C_4, C_5\}$ as input and generate the final set of feature maps defined as follows:

$$P_* = \{P_2, P_3, P_4, P_5, P_6\}. \tag{1}$$

Region Proposal Network. A region proposal network (RPN) can generate region of interests (RoIs) on feature maps P_* by the anchors which are pre-defined in five scales and three aspect ratios. In RPN, classification and bounding box regression are performed by a 3×3 convolutional layer, followed by two sibling 1×1 convolutions, subsequently.

Bounding Box Branch. After RPN, feature maps of size 7×7 from RoIs are extracted by using RoIAlign [10] on $\{P_2,\ P_3,\ P_4,\ P_5\}$, and they are fed into bounding box branch which performs classification and rectangular bounding box regression, respectively.

Tighter Quadrangle Box Branch. In the proposed network, a tighter quadrangle (TQR) box branch is applied to generate building contours using quadrangular bounding boxes. Similar to the sequential protocol of coordinates proposed in [26], via ordering the coordinates, we can define the quadrangular bounding box with four nodes uniquely. By default, the four nodes are arranged clockwise, and the node closest to the grid origin is set to be the first. In particular, if there are two nodes at the same distance with the grid origin, we set the node which

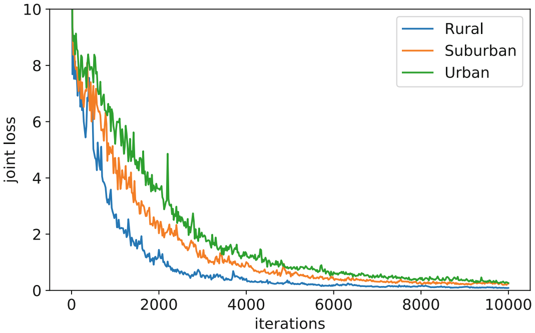


Fig. 3. Joint loss curves of TQR-Net with ResNeXt-101 in three typical areas.

owns smaller value x as the first one. After determining the order of the nodes, inspired by the coordinates of rectangle bounding box as follows:

$$r_* = (x, y, w, h), \tag{2}$$

the 8-coordinate TQR box can be represented as follows:

$$t_* = (x, y, w_1, h_1, w_2, h_2, w_3, h_3, w_4, h_4). \tag{3}$$

Here, variables x, y denote the center coordinates of the TQR box's minimum bounding rectangle, and w_n, h_n represent the n-th $(n = 1, 2, 3, 4)$ relative position to the center coordinates.

As aforementioned, in order to generate the TQR box, $\{P_2,\ P_3,\ P_4,\ P_5\}$ are fed into TQR box branch, which uses RoiAlign to extract 7×7 feature maps from boxes (x_b, y_b, w_b, h_b) output by bounding box branch. Then, three fully-connected layers are utilized to collapse the small feature maps into two 10-d vectors $\{t_0, t_1\}$, where t_0 corresponding to the background class is ignored in the loss computation, and t_1 represents the predicted TQR box. For TQR box regression, we adopt the parameterizations of the 10-coordinate as follows:

$$\begin{aligned}
&d_x = (x - x_b)/w_b,\ d_{w_n} = w_n/w_b,\\
&d_y = (y - y_b)/h_b,\ d_{h_n} = h_n/h_b,\\
&d_x^* = (x^* - x_b)/w_b,\ d_{w_n}^* = w_n^*/w_b,\\
&d_y^* = (y^* - y_b)/h_b,\ d_{h_n}^* = h_n^*/h_b,
\end{aligned} \tag{4}$$

where x^*, y^*, w_n^*, h_n^* $(n = 1, 2, 3, 4)$ stand for the ground-truth TQR box.

2.2 Loss Function

For end-to-end training, we utilize a joint loss to optimize our network. Here, the joint loss is combined of L_{rpn}, L_{bbox} and L_{tqr}, for region proposal network,

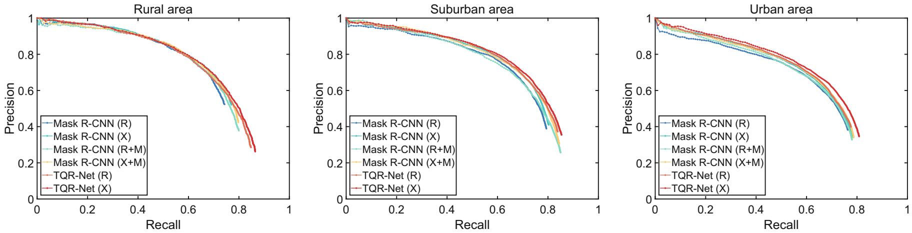


Fig. 4. Precision-recall comparisons of bounding box between TQR-Net and other baseline methods with different backbones on Qinghai Province dataset in three different kinds of areas. (IoU = 0.5). Key: R = ResNet-101-FPN; X = ResNeXt-101-FPN; M = Mask Branch.

bounding box branch and TQR box branch, respectively. Formally, we compute the joint loss function L for each mini-batch as follows:

$$L = \sum_{\theta}^{\Theta} L_{rpn}^{(\theta)} + \sum_{\theta}^{\Theta} L_{bbox}^{(\theta)} + \sum_{\theta}^{\Theta} L_{tqr}^{(\theta)} + \varphi \parallel \mathbf{w} \parallel^2, \tag{5}$$

where φ is a hyper-parameter, $\mathbf{w}$ is a vector of network weights and, the definition of RPN loss $L_{rpn}^{(\theta)}$ and bounding box branch loss $L_{bbox}^{(\theta)}$ can refer to [9,10], for the θ-th image in a mini-batch (e.g., batch size $\Theta = 3$ in our experiments). Moreover, the TQR box branch loss L_{tqr} for one image is defined as follows:

$$\begin{aligned} L_{tqr}(\{d_i\},\{d_i^*\}) &= \lambda \frac{1}{N_{tqr}} \sum_i smooth_{L_1}(d_i - d_i^*), \\ smooth_{L_1}(x) &= \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}. \end{aligned} \tag{6}$$

Here, i and N_{tqr} are the index and number of the TQR boxes, and d_i and d_i^* represent the 10 parameterized coordinates of the predicted and ground-truth TQR boxes, respectively. For the regression loss, we use $smooth_{L_1}$ which is the robust loss function defined in [8].

In this paper, we set the weight decay $\varphi = 0.0001, N_{tqr} = 1000$, and the loss weight $\lambda = 10$. The joint loss curves of TQR-Net with ResNeXt-101 in three typical kinds of areas are shown in Fig. 3.

3 Experiments and Discussion

3.1 Dataset

In order to evaluate our method, we collect a large building dataset from Google Earth, in which all buildings are manually labeled by minimum bounding rectangles. The RGB images in this dataset are from rural, suburban and urban

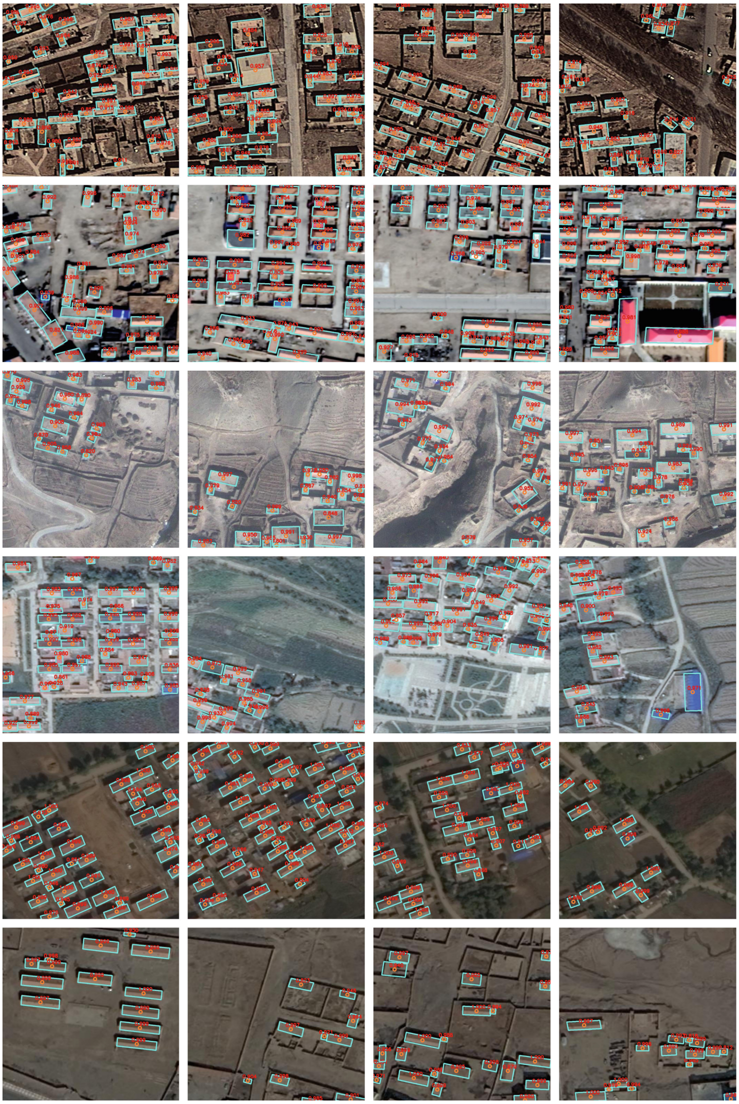


Fig. 5. Building localization results in Qinghai Province, China. First two rows (urban): Tianjun Dist. in Haixi Mongolian T.A.P ($37.30° N$, $99.02° E$) and Xinghai Dist. in Hainan T.A.P ($35.58° N$, $99.99° E$). Key: T.A.P = Tibetan Autonomous Prefecture; E.A = Ethnic Autonomous Dist. Second two rows (suburban): Tu E.A.D in Haidong City ($36.82° N$, $101.99° E$) and Tongde Dist. in Hainan T.A.P ($35.26° N, 100.55° E$). Last two rows (rural): Gonghe Dist. in Hainan T.A.P ($36.40° N, 100.97° E$) and Datong Hui and Tu E.A.D in Xining City ($37.03° N, 101.50° E$).

areas in Qinghai Province, China. Statistically, there are 48222 labeled buildings (7628, 16533 and 24061 in rural, suburban and urban areas) in 1660 images

Table 1. Comparisons of bounding box AP^{bb}(%) and AR^{bb}(%) among the baseline methods and the proposed method on Qinghai Province dataset in three different kinds of areas. Key: M.R. = Mask R-CNN [10]; R = ResNet-101-FPN; X = ResNeXt-101-FPN; M = Mask Branch.

Area	Method	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AR^{bb}
Rural	M.R. (R)	32.5	64.9	29.0	41.4
	M.R. (X)	34.7	66.7	33.3	43.8
	M.R. (R+M)	34.3	67.0	32.7	44.7
	M.R. (X+M)	35.1	67.7	33.2	45.8
	TQR-Net (R)	38.2	68.9	38.7	49.8
	TQR-Net (X)	**38.8**	**70.7**	**39.7**	**51.3**
Suburban	M.R. (R)	33.4	65.3	30.9	44.5
	M.R. (X)	34.9	67.3	32.9	46.2
	M.R. (R+M)	35.4	67.6	34.1	49.7
	M.R. (X+M)	37.0	69.3	36.7	49.3
	TQR-Net (R)	38.7	69.3	39.9	50.8
	TQR-Net (X)	**39.8**	**70.4**	**41.4**	**52.0**
Urban	M.R. (R)	28.8	58.3	25.6	42.0
	M.R. (X)	31.1	61.0	29.1	43.2
	M.R. (R+M)	30.5	59.9	28.5	43.8
	M.R. (X+M)	32.0	61.3	31.1	44.7
	TQR-Net (R)	33.7	61.6	33.5	46.8
	TQR-Net (X)	**35.4**	**64.3**	**36.6**	**48.5**

(296, 631 and 733 in rural, suburban and urban areas). For each area, images are randomly split into 50% for training and 50% for testing.

3.2 Implementation and Results

All models are implemented with PyTorch on 3 NVIDIA GeForce GTX 1080 Ti of 11 GB on board memory. We evaluate ResNet-101 [27] and ResNeXt-101 [24] pre-trained on ImageNet [28] as backbone. As for the parameters in the new layers, we adopt the weight initialization strategy introduced in [29]. In order to train our network, we use stochastic gradient descent (SGD) with a fixed learning rate of 0.002, and the momentum is set to 0.9.

The proposed TQR-Net is compared with Mask R-CNN [10] in three typical areas. We also compare the TQR box branch with the mask branch. Table 1 shows the comparison results of COCO-style bounding box average precision (AP^{bb}) and average recall (AR^{bb}), following the definitions in [30].

In Table 1, we can see that TQR-Net outperforms the baseline methods in both AP^{bb} and AR^{bb} indicators in all three areas. For example, compared to Mask R-CNN with the mask branch, TQR-Net improves 3.7% in AP^{bb} and 5.5%

in AR^{bb} while using ResNeXt-101 as backbone in rural area. Moreover, we show precision-recall curves comparisons of our method and other competitors with different backbones in three different kinds of areas, respectively, in Fig. 4 (for convenience, we draw precision-recall curves according to PASCAL VOC format here). Some localization results generated by TQR-Net with ResNeXt-101 as backbone can be seen in Fig. 5. Thus, our method preserves more geometric information with maintaining certain structural restrictions, which can aid building localization.

4 Conclusion

In this paper, a multi-stage CNN-based method called TQR-Net has been proposed to locate buildings with quadrangle bounding boxes, which can be trained end-to-end by a joint loss function. We make a trade-off between rectangular and polygonal bounding boxes to acquire high-quality building contours in our method. Different from traditional object detection-based and instance segmentation-based methods, TQR-Net can directly generate TQR boxes with more flexibility of freedom than bounding boxes, while avoiding irregular shapes, extra time and resource overheads, associated with predicting masks. Experiments on a large Google Earth dataset of three typical kinds of areas demonstrate its effectiveness for building instance localization task.

References

1. Kim, T., Muller, J.-P.: Development of a graph-based approach for building detection. Image Vis. Comput. **17**(1), 3–14 (1999)
2. Jung, C.R., Schramm, R.: Rectangle detection based on a windowed Hough transform. In: Proceedings, 17th Brazilian Symposium on Computer Graphics and Image Processing, pp. 113–120 (2004)
3. He, L., et al.: A comparative study of deformable contour methods on medical image segmentation. Image Vis. Comput. **26**(2), 141–163 (2008)
4. Kampffmeyer, M., Salberg, A.-B., Jenssen, R.: Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–9 (2016)
5. Wu, G., et al.: Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. Remote Sens. **10**(3), 407 (2018)
6. Troya-Galvis, A., Gançarski, P., Berti-Équille, L.: Remote sensing image analysis by aggregation of segmentation-classification collaborative agents. Pattern Recogn. **73**, 259–274 (2018)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
8. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

9. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
11. Ševo, I., Avramović, A.: Convolutional neural network based automatic object detection on aerial images. IEEE Geosci. Remote Sens. Lett. **13**(5), 740–744 (2016)
12. Cheng, G., Zhou, P., Han, J.: Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans. Geosci. Remote Sens. **54**(12), 7405–7415 (2016)
13. Ren, Y., Zhu, C., Xiao, S.: Small object detection in optical remote sensing images via modified Faster R-CNN. Appl. Sci. **8**(5), 813 (2018)
14. Chen, F., et al.: Fast automatic airport detection in remote sensing images using convolutional neural networks. Remote Sens. **10**(3), 443 (2018)
15. Li, K., Cheng, G., Bu, S., You, X.: Rotation-insensitive and context-augmented object detection in remote sensing images. IEEE Trans. Geosci. Remote Sens. **56**(4), 2337–2348 (2018)
16. Li, Q., Mou, L., Jiang, K., Liu, Q., Wang, Y., Zhu, X.X.: Hierarchical region based convolution neural network for multiscale object detection in remote sensing images. In: IEEE International Geoscience and Remote Sensing Symposium, pp. 4355–4358 (2018)
17. Li, Q., Mou, L., Liu, Q., Wang, Y., Zhu, X.X.: HSF-Net: multiscale deep feature embedding for ship detection in optical remote sensing imagery. IEEE Trans. Geosci. Remote Sens. **56**(12), 7147–7161 (2018)
18. Zhang, Q., Wang, Y., Liu, Q., Liu, X., Wang, W.: CNN based suburban building detection using monocular high resolution Google earth images. In: IEEE International Geoscience and Remote Sensing Symposium, pp. 661–664 (2016)
19. Li, Q., Wang, Y., Liu, Q., Wang, W.: Hough transform guided deep feature extraction for dense building detection in remote sensing images. In: International Conference on Acoustics, Speech and Signal Processing, pp. 1872–1876 (2018)
20. Chen, C., Gong, W., Chen, Y., Li, W.: Learning a two-stage CNN model for multi-sized building detection in remote sensing images. Remote Sens. Lett. **10**(2), 103–110 (2019)
21. Pinheiro, P.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems, pp. 1990–1998 (2015)
22. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3150–3158 (2016)
23. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2359–2367 (2017)
24. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5987–5995 (2017)
25. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, no. 2, p. 4 (2017)
26. Liu, Y., Jin, L.: Deep matching prior network: toward tighter multi-oriented text detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3454–3461 (2017)

27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
28. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
29. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
30. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48