HOUGH TRANSFORM GUIDED DEEP FEATURE EXTRACTION FOR DENSE BUILDING DETECTION IN REMOTE SENSING IMAGES

Qingpeng Li[1], *Yunhong Wang*[1*], *Qingjie Liu*[1], *Wei Wang*[2]

School of Computer Science and Engineering, Beihang University, Beijng, China[1]
National Disaster Reduction Center of China Ministry of Civil Affairs, Beijing, China[2]

ABSTRACT

Detecting dense buildings without elevation information is an important and challenging task in remote sensing applications. In this paper, we present a novel cascaded deep neural network architecture, incorporating multi-stage region proposal detection and Hough transform to obtain better mid-level semantic information for man-made objects. This proposed network can be trained end-to-end by multi-loss jointly. We train and test it on a large building dataset collected from Google Earth, including buildings from urban, suburban and rural areas. Experiments demonstrate great robustness and superiority of our method to various buildings over other convolutional neural network (CNN) based detection methods.

***Index Terms*—** Building detection, Hough transform, deep learning, CNN, remote sensing

1. INTRODUCTION

With the rapid development of remote sensing technologies, automatic building detection has become a practical interest for many applications, such as land management, change detection and disaster assessment. Normally, there are two kinds of remotely sensed data available for building detection task. One is with elevation, such as LiDAR [1], DSM [2] and SAR data [3]. The other is non-elevation information based data, which can been found considerably prevalent in optical remote sensing imagery. Although detecting buildings from elevation data is relatively easy, it is still expensive to acquire such data. Thus, non-elevation based data such as optical imagery, has become the main data source for building detection. However, most of the conventional approaches to processing these data always adopt low-level or handcrafted features, which suffers from a lack of robustness and generalization, still leaving it hard to achieve satisfactory results.

Building detection tasks. Early studies on building detection or reconstruction mainly involved low-level geometry features, such as edges and corners [4, 5]. Building detection from elevation data such as LiDAR, focuses more on reconstructing buildings from 3D point data [6], and detection from monocular optical images spends more effort on discovering useful cues related to the identification of buildings, or designing machine learning algorithms to recognize buildings. For example, many methods use shadow information as an important cue to identify buildings [7], and some methods utilize either unsupervised [8, 9] or supervised learning [10] algorithms to detect buildings. Most of these methods adopt low-level or handcrafted features to distinguish buildings from their surroundings, which is unable to achieve accepted performance due to clutter backgrounds, diversity shapes and various building types. These years, inspired by the achievement of CNN methods in computer vision [11], more and more man-made objects detection methods are developed based on CNNs and the performance has been improved a lot. For instance, Zhang et al. [12] proposed a CNN building detection method with multi-scale saliency based sliding window to detect buildings in suburban area.

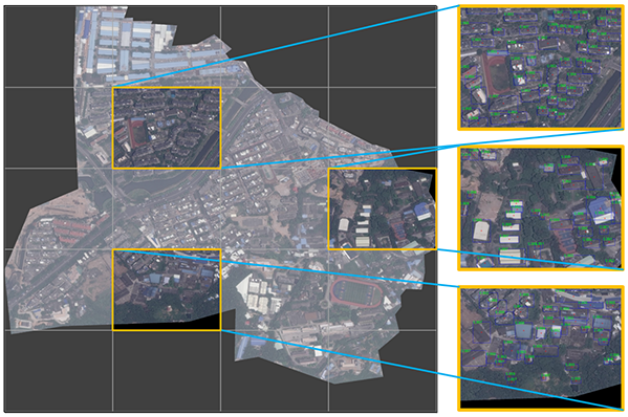


Fig. 1: Building detection result in Google Earth image of Cangshan District, Fuzhou City, Fujian Province.

CNN based object detection. In recent years, CNN based object detection methods have made great advances. In general, they are classified into two categories. One is one-stage approaches which always show higher speed but lower precision on some public datasets (e.g., PASCL VOC and COCO benchmarks), such as SSD [13] and YOLO [14]. The other is two-stage approaches which are cascaded with

Part of this work has been supported by National Disaster Reduction Center of China Ministry of Civil Affairs.

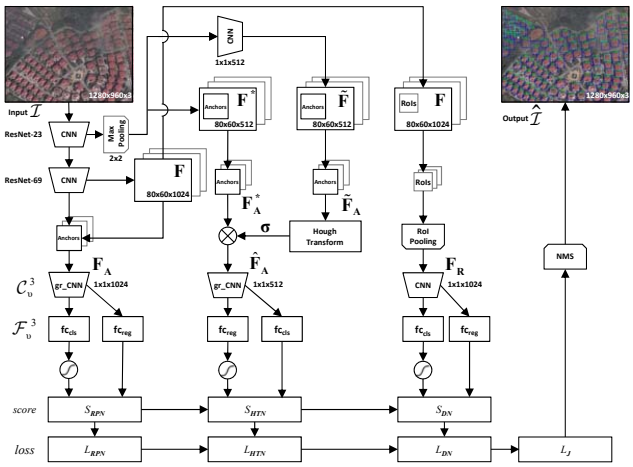


Fig. 2: Illustration of our 3-branch cascaded network for building detection.

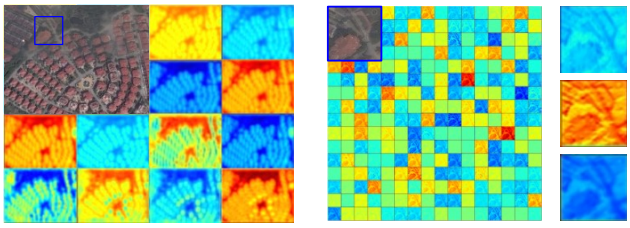
multi-detectors, such as Faster R-CNN [15], R-FCN [16], showing state-of-art precision and better flexibility in multi-scale features.

In this paper, aiming to detect buildings from dense building regions, we design a cascaded CNN architecture. Fig. 1 shows an example result of our method. In this model, we combine mid-level handcrafted features in Hough transform space and deep learning based high-level features together, so that prior knowledge can be adopted to guide CNN training. For building detection tasks, we also improved a region proposal based framework [15, 16] into a cascaded model. Besides, a joint training strategy is also used to achieve faster training speed and better performance.

2. CASCADED CNNS FOR BUILDING DETECTION

As shown in Fig. 2, the proposed cascaded model for building detection consists of following parts. Take an image $\mathcal{I}$ of size $1,280 \times 960$ as input. First, a 23-layer ResNet [17] (ResNet-23) is applied for mid-level feature extraction, which is followed by a max pooling layer and generates 512-d feature maps $\mathbf{F}^*$ of size 80×60. Then, $\mathbf{F}^*$ is sent into two mini-branches. One is to keep the features, the other is to get features $\widetilde{\mathbf{F}}$ after an $1 \times 1 \times 512$ convolutional layer. These mini-branches are called Hough transform network (HTN).

Different from Hough transform network, the region proposal network (RPN) branch and detection network (DN) branch are designed to generate high-level features $\mathbf{F}$ [15]. Thus, after being extracted by ResNet-23, the features are continuously sent into the rest of ResNet (ResNet-69) and we can get the feature maps of size $80 \times 60 \times 1024$. Then, all these three branches can be jointly trained by multi-loss.



(a) (b)

Fig. 3: The capture of feature maps from different levels. (a) represents the high-level feature maps extracted by ResNet-23 and ResNet-69 of the whole $1,280 \times 960$ input image. (b) represents the mid-level feature maps extracted by ResNet-23 of the anchored 256×256 area in (a), which includes more line and edge information.

2.1. Hough transform network

It has been proved that, line and edge features are useful to detect man-made objects such as buildings and roads [18, 19]. Compared with other objects, buildings can be detected with higher probability by more lines and edges[1]. The challenge is that these features are hard to be extracted from complex remote sensing scenes in original image space by conventional methods. Nevertheless, CNN model can filter out most of backgrounds and keep high response for building regions, which is more convenient to extract lines and edges from feature maps instead of raw image domain.

Extracting region proposals by anchors. We get feature maps $\{\mathbf{F}, \mathbf{F}^*, \widetilde{\mathbf{F}}\}$ (80×60 blocks) after feature extraction. In RPN and HTN branch, we slide 9 anchors of 3 scales (128^2, 256^2, 512^2 mapped pixels and $1:1$, $1:2$, $2:1$ ratios for each) per block on each feature maps synchronously. We define feature maps of each branch for the a-th anchor (there are $N_A = 80 \times 60 \times 9 = 43,200$ anchors for input $\mathcal{I}$) as $\{\mathbf{F}_A, \mathbf{F}_A^*, \widetilde{\mathbf{F}}_A\}$ and each anchor maybe propose building objects. In our derivation process of this paper, for simplification, suppose that we have one anchored region size of 256^2, and its matching area on feature map is 16^2, as is shown in Fig.3(b).

Hough transform on feature maps. On HTN stage, we use Canny Operator to extract edges on feature maps $\widetilde{\mathbf{F}}_A$. Inspired by classical Hough transform to extract line features, for simplified calculation, we just consider standard Hough transform (SHT) and set the thresholds, including the maximum of extreme points $\mathcal{N}$ ($\mathcal{N} = 1, 2, ..., 16^2$), the minimum distance between two line segments distance of the same Hough transform bin $\mathcal{D}$ ($\mathcal{D} = 1, 2, ..., 16$) and the minimum line length $\mathcal{L}$ ($\mathcal{L} = 1, 2, ..., 16$). Empirically, we randomly set $\mathcal{N} = 10 \sim 30$, $\mathcal{L} = 3 \sim 5$ and $\mathcal{D} = 3 \sim 5$ at each time.

Furthermore, considering one slice feature map $\widetilde{F}_A^{(k)} \in \widetilde{\mathbf{F}}_A$ ($k = 1, 2, ..., 512$) as input, we can get a series of lines $\boldsymbol{l}$ by SHT in its Canny-operated binary image. After selecting

[1]Note that in this paper, our proposed HTN branch may take effect on both true buildings and some other false man-made objects such as roads, so we expect to weaken the effects of the false detection in other branches.

the longest line in each averagely separated angle region to reduce repeated calculation, the selected lines $\tilde{\boldsymbol{l}}$ are then utilized for loss analysis. We set κ to represent the numbers of the selected lines.

Here, a 7-d vector $\boldsymbol{q}^{(i)} = (\rho^{(i)}, \theta^{(i)}, \boldsymbol{p_1}^{(i)}, \boldsymbol{p_2}^{(i)}, n^{(i)})$ can be given to represent the i-th selected line $\tilde{l}^{(i)}$ $(i = 1, 2, ..., \kappa)$ in $\tilde{\boldsymbol{l}}$, where $(\rho^{(i)}, \theta^{(i)})$ $(-90° \leq \theta^{(i)} \leq 90°)$ represents the Hough parameters of line $\tilde{l}^{(i)}$ in Hough space, and points $(\boldsymbol{p_1}^{(i)}, \boldsymbol{p_2}^{(i)})$ are used to represent the 2 endpoints of line $\tilde{l}^{(i)}$ in the k-th feature map $\widetilde{F}_A^{(k)}$ $(k = 1, 2, ..., 512)$ of size 16×16. Notably, we define the number of extreme points $n^{(i)}$ for each line. For building detection task, we hope that more and longer lines in feature maps can be extracted, hence, the normalized weight σ_k $(0 < \sigma \leq 1)$ for feature map $\widetilde{F}_A^{(k)}$ can be defined as

$$\sigma_k = \frac{\sum_{i=1}^{\kappa} \parallel n_k^{(i)}(\boldsymbol{p_1}^{(i)} - \boldsymbol{p_2}^{(i)})_k \parallel_2}{\max\limits_k \sum_{i=1}^{\kappa} \parallel n^{(i)_k}(\boldsymbol{p_1}^{(i)} - \boldsymbol{p_2}^{(i)})_k \parallel_2} \ (k = 1, 2, ..., 512).$$

And the feature maps $\widehat{\mathbf{F}}_A$ of an anchor in HTN branch can be defined as $\widehat{\mathbf{F}}_A = \boldsymbol{\sigma}\mathbf{F}_A^*$, where $\boldsymbol{\sigma}$ is the set of σ_k.

2.2. Loss function definition

The multi-loss is combined of L_{RPN}, L_{HTN} and L_{DN}. Here, L_{RPN}, L_{HTN} and L_{DN} are defined based on ground-truth of classification (foreground or background) and bounding box regression. In testing process, we serially compute the scores in RPN, HTN and DN, to orderly give candidates.

In order to compute the loss functions of the υ-th $(\upsilon = 1, 2, 3)$ branch, we define the predicted probability of being a building as the score $S^{(\upsilon)} \in \{S_{RPN}, S_{HTN}, S_{DN}\}$. While $S^{*(\upsilon)} \in \{0, 1\}$ is defined as the ground-truth label, where 1 indicates the anchor is a true building, and is 0 if the anchor is not. We define $f(\langle\langle \mathcal{C}_\upsilon^3, \mathbf{F}_A^{(\upsilon)}\rangle, \mathcal{F}_\upsilon^3\rangle)$, where $f(\cdot)$ represents the element-wise nonlinear mapping function $f(\alpha) = 1/(1 + e^{-\alpha})$, and $\mathcal{C}_\upsilon^3 \in \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$, $\mathbf{F}_A^{(\upsilon)} \in \{\mathbf{F}_A^{(1)}, \mathbf{F}_A^{(2)}, \mathbf{F}_A^{(3)}\}$ and $\mathcal{F}_\upsilon^3 \in \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$ represent convolution layers, feature maps and fully-connected layers. Here, $\mathbf{F}_A^{(1)} = \mathbf{F}_A, \mathbf{F}_A^{(2)} = \widehat{\mathbf{F}}_A, \mathbf{F}_A^{(3)} = \mathbf{F}_R$, and $\mathbf{F}_R$ is the feature maps of the RoI area after RoI pooling layer. In particular, $\mathcal{C}_1$ and $\mathcal{C}_2$ are both grouped convolutions [20], by which we expect to speed up the training process. $\mathcal{F}_\upsilon^3$ is of three fully-connected layers with 256-d neurons.

We also define $L^{(\upsilon)} \in \{L_{RPN}, L_{HTN}, L_{DN}\}$ $(\upsilon = 1, 2, 3)$. Then we can compute the multi-task loss $L^{(\upsilon)}$ by

$$L^{(\upsilon)} = \sum_{a=1}^{N_A} (L_{cls}^{(\upsilon)} + \eta S^{*(\upsilon)} L_{reg}^{(\upsilon)})_a, \tag{1}$$

where, hyper-parameter $\eta = 1$, and the classification loss function $L_{cls}^{(\upsilon)}$ is defined as log loss for true class over two classes (building or not) and the regression loss $L_{reg}^{(\upsilon)}$ is defined as smooth L_1 loss. They are defined as follows

$$\begin{aligned} L_{cls}^{(\upsilon)}(S^{(\upsilon)}, S^{*(\upsilon)}) &= -S^{(\upsilon)} \log S^{(\upsilon)} \\ L_{reg}^{(\upsilon)}(\boldsymbol{t}, \boldsymbol{t^*}) &= \sum_{\mu \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_\mu, t_\mu^*). \end{aligned} \tag{2}$$

Here, smooth L_1 loss is as follows

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \tag{3}$$

which is proved to be less sensitive to outliers. In (2), t_μ and t_μ^* represent the 4 parameters of 4-d vectors $\boldsymbol{t}$ and $\boldsymbol{t^*}$, where $\boldsymbol{t}$ and $\boldsymbol{t^*}$ are parameters of the bounding box and its ground-truth, which can be parameterized by coordinates (t_x, t_y, t_w, t_h) and $(t_x^*, t_y^*, t_w^*, t_h^*)$ [15].

For the loss matrix $L^{(\upsilon)}$ $(\upsilon = 1, 2, 3)$, we can add them with loss weights $\{\lambda_1, \lambda_2, \lambda_3\}$ and the joint loss L_J is defined as follows

$$\begin{aligned} L_J =& \lambda_1 L_{RPN} + \lambda_2 L_{HTN} + \lambda_3 L_{DN} + \phi \parallel \boldsymbol{w} \parallel^2 \\ =& \sum_{\upsilon}^{3} \lambda_\upsilon L^{(\upsilon)} + \phi \parallel \boldsymbol{w} \parallel^2 \\ =& \sum_{\upsilon}^{3} \sum_{a}^{N_A} \lambda_\upsilon (L_{cls}^{(\upsilon)} + \eta S^{*(\upsilon)} L_{reg}^{(\upsilon)})_a + \phi \parallel \boldsymbol{w} \parallel^2, \end{aligned} \tag{4}$$

where L_{RPN}, L_{HTN} and L_{DN} calculated by (2) denote losses of three branches. In this paper, we set loss weights $\lambda_1 = \lambda_2 = 1$ and $\lambda_3 = 10$. In (4), ϕ is a hyper-parameter and w is a vector of learnable model parameters. We learn $\boldsymbol{w}$ by optimizing the regularized least squares objective.

2.3. Training and testing procedure

Training procedure. Considering an image of size $1,280 \times 960$ as input, after extracting features, we can get feature maps with 80×60 blocks. In forward propagation step, we generate anchors and feed the relative features into different branches (i.e., RPN, HTN and DN). and we can compute the loss on each branch and get the final joint loss of the whole model in (4). Then we update the parameters for each branch by the same joint loss in backward propagation (BP) step. The whole model is end-to-end trained.

Fig. 4: Diverse building samples in our dataset.

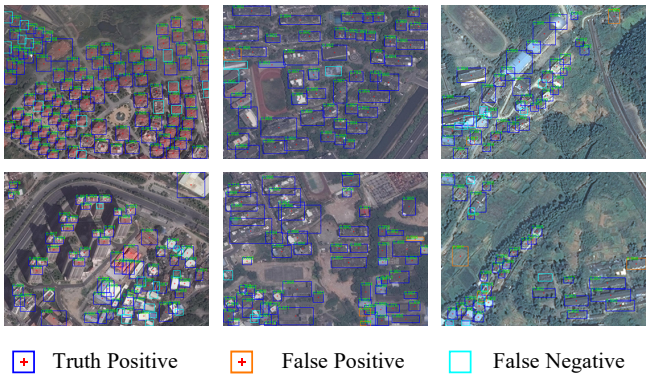


Fig. 5: Captures of the experiment results in three dense detection cases in Fujian Province, China. The first column is Fengze District in Quanzhou City (urban area). The second is Cangshan District in Fuzhou City (suburban area). The third is Jianyang District in Nanping City (rural area).

Table 1: **Experiment results of three chosen places.**

	Case-1	Case-2	Case-3
	Urban	Suburban	Rural
Area	Fengze	Cangshan	Jianyang
Lat.	24.89°N	26.04°N	27.32°N
Lon.	118.62°E	119.29°E	118.13°E
True Positive	658	506	594
False Positive	156	139	214
False Negative	212	194	314
$\mathcal{P}_0$	0.3	0.3	0.3
IoU	0.7	0.6	0.5
AP(%)	80.8	78.4	73.5
Recall(%)	75.6	72.3	65.4

Testing procedure. Different from the parallel step in training process, our test pipline is serial. As a typical cascaded model, after feature extraction stage, the first two branches of proposing RoIs can generate scores (S_{RPN} and S_{HTN}) together. The fused score is to generate RoIs. In detection network, the score of results (S_{DN}) can be computed to reserve positive RoIs. At last, we use the traditional non-maximum suppression (NMS) strategy to get final results and the NMS defined by threshold $\mathcal{P}_0$ ($0 < \mathcal{P}_0 < 1$) is also enabled in this step.

3. EXPERIMENTAL RESULTS

Dataset. To facilitate the research, our method is tested on manually labeled dataset collected from Google Earth, containing RGB-band. The images in our dataset are collected from different places in Fujian Province, China, varying from typical urban and suburban area to rural area. Ignoring sample augmentation, in total, there are more than 100,000 labelled buildings labeled in 2,180 images (1,090 for training and validation set and 1,090 for test set) in this dataset. Some of the building samples captured from the images are shown in Fig. 4.

Table 2: **Comparison of AP(%) between the proposed method and other baseline methods. (Recall=0.7)**

Method	Urban	Suburban	Rural
Faster R-CNN [15]	75.8	70.2	58.5
R-FCN [16]	76.6	72.4	61.2
Zhang's [12]	65.8	60.6	50.3
Ours	**78.7**	**75.7**	**70.2**

Experiment and comparison. We test our method on three places, including urban area, suburban area and rural area. Some result captures are shown in Fig. 5. According to the results in Table 1, we find that for most rural areas, the probability of true detection is generally below the suburban and rural area. It is possible because the rural buildings are more irregular than others, which are similar to background with spectral appearance, and sometimes it is hard to identify them even by human vision.

We also compare our method with some baseline methods on our test set, such as Faster R-CNN [15], R-FCN [16] and Zhang's method [12]. All of their feature extraction networks use ResNet-101 [17]. The results show that, in different detection cases, our method achieves the best average precision (AP) results that $\mathrm{AP}_{urban} = 78.7\%$, $\mathrm{AP}_{suburban} = 75.7\%$ and $\mathrm{AP}_{urban} = 70.2\%$ when the recall is set to 0.7. The results prove the high robustness of our cascaded method (Table 2).

Implementation details. Prior to the training step, all new layers are randomly initialized by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. The weights of part of primary layers are initialized by pre-trained model for ImageNet classification [21]. In SGD step, we use a learning rate of 0.001 for 3,000 epochs, and 0.0001 for the rest 1,200 epochs on our dataset. The momentum and the weight decay are set to 0.9 and 0.0005. Our total epochs are about 4,200 and the batch size is set to 64. Our implementation is based on Caffe [22]. All experiments are carried on one NVIDIA GeForce GTX 1080Ti GPU with 12G onboard memory.

4. CONCLUSION

In this paper, a cascaded model for dense building detection has been proposed. In our method, Hough transform can guide one CNN branch to extract mid-level features of the building, which is cascaded by other high-level feature extraction branches. To get faster training speed and better performance, our method is trained by joint loss function. Besiges, a large dataset of buildings in different cases is also collected from Google Earth. Experiment results show high performance of our proposed method compared with other baseline methods.

5. REFERENCES

[1] F. Rottensteiner, "Advanced methods for automated object extraction from LiDAR in urban areas," in *IGARSS*, Jul. 2012, pp. 5402–5405.

[2] D. Chai, "A probabilistic framework for building extraction from airborne color image and DSM," *JSTAR*, vol. 10, no. 3, pp. 948–959, Mar. 2017.

[3] B. Liu, K. Tang, and J. Liang, "A bottom-up/top-down hybrid algorithm for model-based building detection in single very high resolution SAR image," *GRSL*, vol. 14, no. 6, pp. 926–930, Jun. 2017.

[4] A. L. Reno, "Detecting buildings in aerial images using shape descriptors," in *ICIP*, Jul. 1997, vol. 2, pp. 468–472 vol.2.

[5] Yanfeng Wei, Zhongming Zhao, and Jianghong Song, "Urban building extraction from high-resolution satellite panchromatic image using clustering and edge detection," in *IGARSS*, Sep. 2004, vol. 3, pp. 2008–2010 vol.3.

[6] Franz Rottensteiner, Gunho Sohn, Markus Gerke, Jan Dirk Wegner, Uwe Breitkopf, and Jaewook Jung, "Results of the ISPRS benchmark on urban object detection and 3D building reconstruction," *P&RS*, vol. 93, no. Supplement C, pp. 256 – 271, 2014.

[7] T. T. Ngo, V. Mazet, C. Collet, and P. de Fraipont, "Shape-based building detection in visible band images using shadow information," *JSTAR*, vol. 10, no. 3, pp. 920–932, Mar. 2017.

[8] D. Chaudhuri, N. K. Kushwaha, A. Samal, and R. C. Agarwal, "Automatic building detection from high-resolution satellite images based on morphology and internal gray variance," *JSTAR*, vol. 9, no. 5, pp. 1767–1779, May 2016.

[9] D. Konstantinidis, T. Stathaki, V. Argyriou, and N. Grammalidis, "Building detection using enhanced HOG-LBP features and region refinement processes," *JSTAR*, vol. 10, no. 3, pp. 888–905, Mar. 2017.

[10] C. Senaras, M. Ozay, and F. T. Yarman Vural, "Building detection with decision fusion," *JSTAR*, vol. 6, no. 3, pp. 1295–1304, Jun. 2013.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[12] Q. Zhang, Y. Wang, Q. Liu, X. Liu, and W. Wang, "CNN based suburban building detection using monocular high resolution Google Earth images," in *IGARSS*, Jul. 2016, pp. 661–664.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, Jun. 2016, pp. 779–788.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "SSD: Single shot multibox detector," in *ECCV*, 2016, pp. 21–37.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[16] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *NIPS*, 2016, pp. 379–387.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, Jun. 2016, pp. 770–778.

[18] M. Wang, S. Yuan, and J. Pan, "Building detection in high resolution satellite urban image using segmentation, corner detection combined with adaptive windowed Hough transform," in *IGARSS*, Jul. 2013, pp. 508–511.

[19] X. Yang and G. Wen, "Road extraction from high-resolution remote sensing images using wavelet transform and Hough transform," in *ICISP*, Oct. 2012, pp. 1095–1099.

[20] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, Jul. 2017, pp. 1800–1807.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "ImageNet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[22] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*, 2014, pp. 675–678.