BUILDING OUTLINE DELINEATION: FROM VERY HIGH RESOLUTION REMOTE SENSING IMAGERY TO POLYGONS WITH AN IMPROVED END-TO-END LEARNING FRAMEWORK

Wufan Zhao[1,*], Ivan Ivanov[1], Claudio Persello[1], Alfred Stein[1]

1 Dept. of Earth Observation Science, ITC, University of Twente, Enschede, The Netherlands – wufan.zhao@utwente.nl, i.ivanov@student.utwente.nl, c.persello@utwente.nl, a.stein@utwente.nl

Commission II, WG II/6

KEY WORDS: Building Outline Delineation, Polygon Prediction, Convolutional Neural Networks, Recurrent Neural Networks

ABSTRACT:

Deep learning methods based on Fully convolution networks (FCNs) have shown an impressive progress in building outline delineation from very high resolution (VHR) remote sensing (RS) imagery. Common issues still exist in extracting precise building shapes and outlines, often resulting in irregular edges and over smoothed corners. In this paper, we use PolyMapper, a recently introduced deep-learning framework that is able to predict object outlines in a vector representation directly. We have introduced two main modifications to this baseline method. First, we introduce EffcientNet as backbone feature encoder to our network, which uses compound coefficient to scale up all dimensions of depth/width/resolution uniformly, to improve the processing speed with fewer parameters. Second, we integrate a boundary refinement block (BRB) to strengthen the boundary feature learning and to further improve the accuracy of corner prediction. The results demonstrate that the end-to-end learnable model is capable of delineating polygons of building outlines that closely approximate the structure of reference labels. Experiments on the crowdAI building instance segmentation datasets show that our model outperforms PolyMapper in all COCO metrics, for instance showing a 0.13 higher mean Average Precision (AP) value and a 0.60 higher mean Average Recall value. Also qualitative results show that our method segments building instances of various shapes more accurately.

1. INTRODUCTION

Automatic building extraction from remote sensing images has been a core research topic in the remote sensing area for decades. It has many applications, including cadastral and topographic mapping, cartography, urban planning, and humanitarian aid. Conventional methods, including both pixel-based and object-based methods, conduct building outline delineation by extracting texture, geometric, shadow and more sophisticated, empirically designed spatial features (Turker et al., 2015).

The recent development of deep learning methods, specifically fully convolutional networks (FCNs), together with the emergence of large amounts of earth observation data, has promoted a new round of research studies toward automated mapping of urban areas (Persello et al., 2017; Ji et al., 2019). Most pixel-based classification methods, however, result in irregular building shapes and require further processing such as shape refinement and vectorization. In addition, several challenges, e.g. overhanging vegetation, shadows, and densely constructed buildings, make the task more difficult in separating building objects. The polygons resulting from the vectorization of FCN-based classification usually need substantial manual editing before being included in GIS layers of official topographic or cadastral maps.

Recent research has started integrating deep convolutional neural networks (CNNs) with regularized and structured building outline delineations. Marcos et al. (2018) proposed Deep Structured Active Contours (DSAC) to predict the energy function parameters for an Active Contour Model (ACM) to make its output close to a ground truth set of polygonal outlines. Girard et al. (2018) propose a deep learning method that predicts the vertices of the polygons outlining objects of interest. Zhao et al., (2018) applied the Mask R-CNN for building segmentation and a regularization algorithm to polygonize segmentation results. However, those methods are either not end-to-end trainable or only capable of handling building objects with simple shapes.

Motivated by the success of recent works on automatic object annotation in the computer vision field (Castrejon et al., 2017), Li et al. (2019) developed an end-to-end deep learning architecture, named PolyMapper, which is able to learn and delineate the regularized geometrical shapes of buildings and roads directly in a vector format from a given overhead image. In our study, we follow this research line by introducing two modifications to improve its performance. Specifically, there are two main contributions in this study:

1) Introduction of a state-of-the-art architecture for the backbone network, EfficientNet (Tan et al., 2019), which uses a compound coefficient to scale up CNNs in a more structured manner. This reduces the number of trainable parameters while maintaining high accuracy.

2) The exploitation of Boundary Refinement Block (BRB) to make the bilateral features of building boundary distinguishable with deep semantic boundary supervision. Integrated into skip feature extraction, it strengthens the boundary feature learning and further improves the accuracy of corner prediction.

[*] Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-731-2020 | © Authors 2020. CC BY 4.0 License.

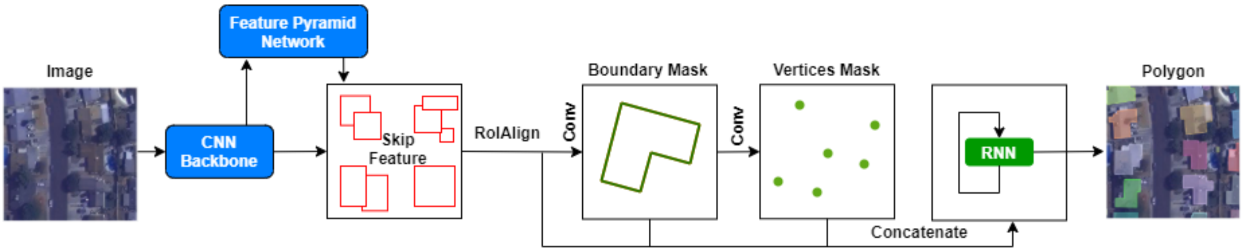

Figure 1. Architecture of the network for building outline polygon delineation. Adapted from Li et al., (2019). In particular, we made modifications in the CNN backbone and skip feature extraction module. See Section 2 for more details.

2. METHODOLOGY

2.1 Overview

The adopted framework integrates multiple network branches trained end-to-end, combining object detection, instance segmentation, and vectorization (Li et al., 2019). In particular, the workflow is able to delineate the building objects and to find vertices and connect them sequentially by using an RNN. As shown in Fig. 1, a CNN backbone takes an image as input to extract multiple levels of spatial features. Then, the Feature Pyramid Network (FPN) generates building bounding boxes, which partitions the image into individual object instances. FPN makes use of the in-network feature hierarchy that produces feature maps with different resolutions to build a feature pyramid. To this end, building objects in different scale can be effectively detected. After acquiring image tiles of individual buildings, RoIAlign is applied to preserve the exact spatial locations for feature extraction. Then additional convolutional layers with skip connections that fuse information from the previous backbone layers extract building boundaries and vertex features. Those are fed sequentially to the multi-layer convolutional long-short term memory (ConvLSTM) module to predict and connect vertices sequentially to obtain the final building polygon.

We observed that the current framework still has room for improvement in terms of backbone encoder and skip feature extraction. We discuss these in details in the following sections.

2.2 Backbone Encoder

A CNN encoder network, also called backbone network, is used to extract features at multiple levels for the tasks of detection and segmentation. From the seminal work of AlexNet (Krizhevsky et al., 2012) and VGG (Simonyan et al., 2014) used in PolyMapper, scaling up CNN in terms of depth, width and image size has been widely developed towards better accuracy. However, arbitrary scaling requires tedious manual tuning and still often yields sub-optimal accuracy and efficiency. To tackle this issue, Tan et al. (2019) proposed a simple yet effective compound scaling method which can uniformly scale network width, depth, and resolution with a set of fixed scaling coefficients:

$$\begin{gathered}\text{depth: } d = \alpha^{\phi} \\ \text{width: } \omega = \beta^{\phi} \\ \text{resolution: } r = \gamma^{\phi} \\ \text{s.t. } \alpha \cdot \beta^{2} \cdot \sigma^{2} \approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1\end{gathered} \tag{1}$$

Based on this approach, they used Neural Architecture Search (NAS) strategy to develop a new baseline network, and scale it up to obtain a family of models, called EfficientNets. Specifically, the authors first fix ϕ and obtain an optimal α, β, γ values using grid search. Then the baseline network is scaled up by fixing the α, β, γ with different compound coefficient ϕ according to *Equation* 1 with less parameters and lower computational cost. More importantly, the compound scaling method allows the scaled model to focus on more relevant regions with more object details.

Stage	Operator	Resolution	#Channels	#Layers
Input	Conv3×3	224×224	32	1
1	MBConv1, k3×3	112×112	16	1
	MBConv6, k3×3	112×112	24	2
2	MBConv6, k3×3	56×56	40	2
3	MBConv6, k3×3	28×28	80	3
4	MBConv6, k3×3	14×14	112	3
	MBConv6, k3×3	14×14	192	4
5	MBConv6, k3×3	7×7	320	1

Table 1. EfficientNet-B0 baseline network

Table 1 shows the basic network structure of EfficientNet-B0. The main building block for EfficientNet is MBConv, an inverted bottleneck convolution, originally known as MobileNetV2 (Sandler et al., 2018). Using shortcuts between bottlenecks by connecting a much smaller number of channels (compared to expansion layers), it was combined with an in-depth separable convolution, which reduced the calculation by almost k^2 compared to traditional layers, where k denotes the kernel size, which specifies the height and width of the 2-dimensional convolution window. Table 2 indicates the bottleneck residual block structure transforming from k to k'channels, with stride s, and expansion factor t.

Input	Operator	Output
$h \times w \times k$	1×1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times (tk)$	3×3 dwise s= s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times (tk)$	linear 1×1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 2. MBConv bottleneck residual block

Considering the balance of model size and its performance, we apply EfficientNet-B5. The scaled width_coefficient, depth_coefficient, and resolution (*α, β, γ*) are set to 1.6, 2.2, 456. In order to extract features of different scales, we fuse different layers into five stages. Moreover, we remove the last pooling stage and fully connected layer from the original structure.

2.3 Discriminative Skip Feature Extraction

As in Polygon RNN (Castrejon et al., 2017), PolyMapper uses additional convolutional layers with skip connections that fuse information from the previous backbone nets and upscale the output by factor of 2. This allows the CNN to extract features that contain low-level information about the building boundaries and corners. This procedure helps the model to follow the object's boundaries and predict more precise vertices.

However, buildings objects in VHR-RS images with high-definition details are always under various complex backgrounds (e.g., shadow, occlusion, and geometric deformation), and often densely distributed. Therefore, we need to amplify the distinction of features. With this motivation, we adopt a semantic boundary to guide the feature learning. To further improve the accuracy of corner prediction of Polymapper, we exploit the BRB (Figure 2), which is the variant of the Refinement Residual Block (RRB) (Yu et al., 2018), trying to differentiate the building objects with background and adjacent feature with similar appearances.

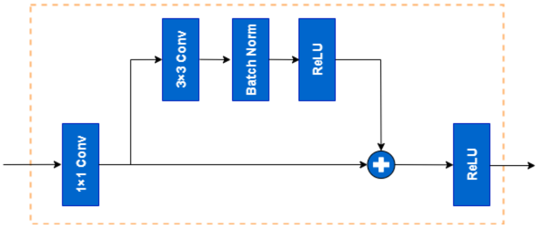

Figure 2. Boundary Refinement Block (BRB). Adapted from Yu et al., (2018)

The first component of the block is a 1 × 1 convolution layer, which is used to unify the channel number and combine the information across the channel. A basic residual block comes as follows to refine the feature map. We remove the 3 × 3 convolutional layer in original RRB to simplify the module. BRBs are then embedded into the network to get discriminative features of the building boundaries from the feature maps of each scale backbone networks simultaneously. The improved CNN for skip feature extraction is illustrated in Figure 3. With the explicit semantic boundary supervision, the network can obtain more distinct boundary features. As an output, we have an output in volume of 28 × 28 × 132 pixels of the encoder features with enhanced boundary information.

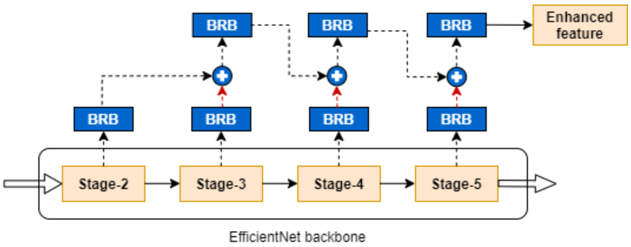

Figure 3. An overview of the multi-scale encoder network integrated with BRB. The red dotted arrows indicate up-sampling.

The combined feature maps and one candidate key point with the highest probability are the input for RNN module to model the sequence of 2D vertices of the polygon outlining the building object. Taking the current and previous vertex as inputs, the ConvLSTM cell generates a conditional probability distribution as an output. The progress ends until the polygon reaching its starting vertex and becoming a closed shape. In the inference phase, beam search procedure is used to select the starting and following vertices. Beam search is a heuristic graph search algorithm, which is used to keep the higher quality nodes at each step of depth expansion in large graph space.

2.4 Loss Function

The total loss of the network is a combined loss from the FPN, CNN and RNN parts. Specifically, the FPN loss consists of a cross-entropy loss for anchor classification and a smooth L1 loss for anchor regression which are described below (for a single anchor).

$$L_{cls}^{(i)} = -(y_i\, log(p_i) + (1 - y_i)\, log(1 - p_i)) \qquad (2)$$

$$f(x) = \begin{cases} \frac{1}{2}x^2, |x| < 1 \\ |x| - \frac{1}{2}, otherwise \end{cases} \qquad (3.1)$$

$$\begin{aligned} L_{box}^{(i)} = f(x_g^* - x_g) + f(y_g^* - y_g) \\ + f(w_g^* - w_g) + f(h_g^* - h_g) \end{aligned} \qquad (3.2)$$

$$L_{FPN} = \sum_i L_{cls}^{(i)} + \lambda \sum_i L_{box}^{(i)} \qquad (4)$$

where L_{cls} is the classification loss, y_i (either 0 or 1) is the polarity of the anchor, p_i is the predicted probability of the class. L_{box} is the regression loss of the bounding box, where$(x_g^*, y_g^*, w_g^*, h_g^*)$ denote the predicted coordinates of the box. The superscript i denotes the i-th anchor, λ denotes a self-defined parameter when training.

In addition, weighted logarithmic loss is used to remedy the imbalance of the positive and negative samples for the mask of boundary and vertices in CNN part separately. The cross-entropy loss is adopt in RNN part for the multi-class classification at each time step.

3. EXPERIMENTAL RESULTS

3.1 Datasets and Evaluation Matrics

3.1.1 Datasets

We performed experiments on the challenging crowdAI dataset to validate our method (Mohanty 2018). The training and testing set consisted of 280,741 and 60,317 tiles, respectively, of 300×300 pixels extracted from the RGB channels of satellite imagery. Typical instance annotations were used to supervise box and mask branches, and the semantic branch is supervised by the Common Objects in Context (COCO) format annotations (Lin et al., 2014). In contrast with standard vector labels and georegistered image files, each object instance annotation in COCO .json contains a series of fields, including the category id, segmentation mask, and enclosing bounding box coordinates of the object. Example images and corresponding label is shown in figure 4.

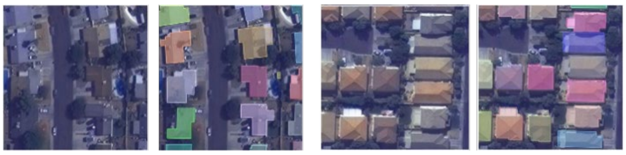

Figure 4. Example tiles of crowdAI dataset. Note that multiple colors are used to differentiate building instances.

3.1.2 Evaluation Metrics

We applied the standard MS COCO measures, including the mean Average Precision (AP) and mean Average Recall (AR) over multiple Intersection over Union (IoU) values. IoU defined as the area of the intersection divided by the area of the union of a predicted bounding box (B_p) and a ground-truth box (B_{gt}).

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \qquad (5)$$

Specifically, AP and AR were averaged over 10 Intersection over Union (IoU) values with thresholds of from .50 to 0.95 with steps of 0.05. Averaging over IoUs rewards detectors with better localization. Thus, an AP was calculated as *Equation* 6.

$$AP = \frac{AP_{0.50} + AP_{0.55} + \ldots + AP_{0.95}}{10} \qquad (6)$$

In addition, $AP_{(S, M, L)}$ and $AR_{(S, M, L)}$ were used to further measure the performance of the algorithm on detecting objects of different sizes. Specifically, small, medium and large represent an area < 32^2, an area between 32^2 and 96^2 and an area > 96^2 respectively, where the area is measured as the number of pixels in the segmentation mask. Both AP and AR were evaluated using mask IoU.

3.2 Implementation Details

We trained our model using the Adam optimizer with a batch size b = 4 and an initial learning rate of 0.0001. Weight decay and momentum were set to 0.9, respectively. The total iteration number was set as 1,600,000. The network was implemented using Tensorflow 1.15. We performed all the training and testing on a single TITAN X GPU.

3.3 Results and Discussion

We compared our model to the state-of-the-art instance segmentation method Mask R-CNN (He et al., 2017) and the original PolyMapper. It is worth noting that we did not reproduce the same results as in Li et al. (2019), which is probably due to a different training procedure (which is not fully detailed in the paper).

3.3.1 Quantitative analysis

Table 3 shows a quantitative comparison of our method with three methods. Our method outperforms PolyMapper in all AP and AR metrics, especially for the later ones. It demonstrates that there is a higher proportion of buildings detected by our approach with respect to the ground truth. In addition, comparing with Mask R-CNN, our method only show slightly lower in AR_L, which refers to large size object. But our method works significantly better in delineating small and medium size buildings and achieves higher precision in all levels.

At the same time, it can also be seen that instance segmentation is relatively weak for small object recognition and is an aspect that can be further optimized in the future. Possible reasons are that small objects have features that are harder to learn and are easily confused with ground objects such as cars.

Method	Mask R-CNN	PolyMapper	Our Method
AP	0.419	0.432	0.445
AP_S	0.124	0.197	0.211
AP_M	0.581	0.568	0.582
AP_L	0.519	0.550	0.558
AR	0.476	0.439	0.499
AR_S	0.181	0.220	0.280
AR_M	0.652	0.566	0.653
AR_L	0.633	0.594	0.606

Table 3. Extraction results on crowdAI dataset

3.3.2 Qualitative analysis

Figure 5 provides a qualitative comparison of the prediction of image annotated and segmented by the three methods. Compared to Mask R-CNN, PolyMapper and our method can generate output in polygon representations automatically instead of pixel-wise output masks.

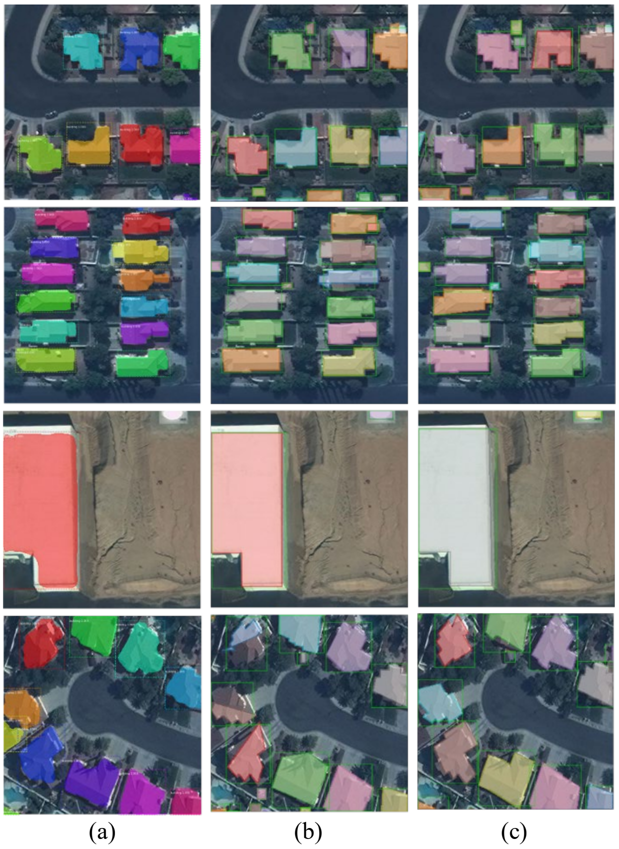

Figure 5. Qualitative results on crowdAI dataset. From left to right: (a) Mask R-CNN, (b) PolyMapper, (c) Our method

As compared with the Mask R-CNN, the other two methods are able to correctly segment building instances with a variety of shapes and sizes. The output shapes are more compact and regularized. In addition, our model is able to segment building instances of various shapes more accurately compared with PolyMapper. Moreover, our method yields more accurate results for vertices detection. Because BRB enhances the learning ability for boundary features and further guides the detection of corner points. This also avoids incomplete detection of the geometry. In summary, our model can well outline the buildings in a given VHR-RS image and provide accurate geometrical details.

4. CONCLUSIONS

In this study, we investigated an end-to-end learnable model for building outline polygon extraction from VHR-RS image. Our method is built on top of PolyMapper, and introduces several improvements that allow us to increase its performance in terms of accuracy and regularity. The comparisons against other state-of-the-art methods on a benchmark dataset, demonstrate our method's ability in regularized building outlines are better aligned with ground truth building boundaries. To extend, we are currently working on further improving from the following points of view: 1) testing our method on larger image scene with dense building objects; 2) refining the training strategy.

ACKNOWLEDGEMENTS

Authors declare that they have no conflict of interest. The first author is funded by the China Scholarship Council (CSC) from the Ministry of Education of P.R. China.

REFERENCES

Castrejon, L., Kundu, K., Urtasun, R. and Fidler, S., 2017. Annotating object instances with a polygon-rnn. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 5230-5238.

Girard, N. and Tarabalka, Y., 2018. End-to-end learning of polygons for remote sensing image classification. In IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium pp. 2083-2086.

He, K., Gkioxari G., Dollar P., and Girshick R., "Mask R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2017, vol. 2017-Oct, pp. 2980–2988.

Ji, S., Wei, S. and Lu, M., 2018. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Transactions on Geoscience and Remote Sensing, 57(1), pp.574-586.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

Li, Z., Wegner, J.D. and Lucchi, A., 2019. Topological map extraction from overhead images. In Proceedings of the IEEE International Conference on Computer Vision pp. 1715-1724.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014, September. Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R. and Urtasun, R., 2018. Learning deep structured active contours end-to-end. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 8877-8885.

Mohanty S. 2018. Crowdai Mapping Challenge 2018: Baseline with Maskrcnn. https://github.com/crowdai/crowdai-mapping-challenge-mask-rcnn.

Persello, C., Stein, A., 2017. Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images. IEEE Geoscience and Remote Sensing Letters 14, 2325–2329.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Tan, M. and Le, Q.V., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946.

Turker, M. and Koc-San, D., 2015. Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (SVM) classification, Hough transformation and perceptual grouping. International Journal of Applied Earth Observation and Geoinformation, 34, pp.58-69.

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G. and Sang, N., 2018. Learning a discriminative feature network for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 1857-1866.

Zhao, K., Kang, J., Jung, J. and Sohn, G., 2018, June. Building Extraction From Satellite Images Using Mask R-CNN With Building Boundary Regularization. In CVPR Workshops pp. 247-251.

https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-731-2020 |