

Learning Orientation-Estimation Convolutional Neural Network for Building Detection in Optical Remote Sensing Image

Yongliang Chen

Key Lab of Optoelectronic Technology and System of Education Ministry
Chongqing University
 Chongqing, China
 ylchen@cqu.edu.cn

Abstract

Benefiting from the great success of deep learning in computer vision, CNN-based object detection methods have drawn significant attentions. Various frameworks have been proposed which show awesome and robust performance for a large range of datasets. However, for building detection in remote sensing images, buildings always pose a diversity of orientations which makes it a challenge for the application of off-the-shelf methods to building detection. In this work, we aim to integrate orientation regression into the popular axis-aligned bounding-box detection method to tackle this problem. To adapt the axis-aligned bounding boxes to arbitrarily orientated ones, we also develop an algorithm to estimate the Intersection over Union (IoU) overlap between any two arbitrarily oriented boxes which is convenient to implement in Graphics Processing Unit (GPU) for accelerating computation. The proposed method utilizes CNN for both robust feature extraction and rotated bounding box regression. We present our model in an end-to-end fashion making it easy to train. The model is formulated and trained to predict orientation, location and extent simultaneously obtaining tighter bounding box and hence, higher mean average precision (mAP). Experiments on remote sensing images of different scales shows a promising performance over the conventional one.

Index Terms

building detection, remote sensing image, convolutional neural network, orientation regression, intersection over union estimation

I. INTRODUCTION

With building being a key factor in region management and planning, building detection in very high resolution (VHR) optical remote sensing images, which has numerous applications such as damage assessments by comparison of detection results [1] and map updating [2], becomes one of the inevitable challenge in aerial and satellite image analysis. Extensive methods have been exploited by researchers in recent years. Generally, those methods can be classified into four categories [3]: template matching-based methods [4], knowledge-based methods [5], OBIA (object-based image analysis)-based methods [6], and machine learning-based methods [7].

Thanks to the advancing in feature representations and classifier design, machine learning-based methods has drawn significant attention of researchers, which casts detection work into a classification task. In [8], lower-level features containing spatial and structural information is extracted to construct higher features, which is followed by a Bayesian framework to learn object detector iteratively. [9] build HOG (Histogram of Oriented)-based feature pyramids to train a support vector machine (SVM). [10] exploit the complex patterns of contrast features provided by the training data to model buildings, in which detection problem is also reduced to a machine learning classification task by introducing candidate region search.

More recently, deep learning [11] has made a significant breakthrough in the field of computer vision. Convolutional neural networks (CNNs) with deep structure, which directly processes the raw pixels of input image to generate multiple level feature representations with semantic abstracting properties, has shown an impressively strong power in feature representation and obtained a series of success in a wide range of applications [12]. As deep CNN becomes a powerful tool in feature extraction, more sophisticated detection methodologies have been developed [13]–[18]. In these work, feature maps gathered from convolutional layers play a key role. [14] use CNNs to extract features from each candidate region for the classifier. [13] does classification directly on the feature maps by introducing ROI (region of interest) layer, which significant reduces computation cost, and applies a regression layer for finer locating objects.

To deal with heavily computational budget and fully utilize computation capability of graphic processing unit (GPU), [18] integrate generating candidate regions into the networks by designing proposes region proposal network (RPN) which coarsely locates objects via predicting class and offset for each anchor box, which is configured by a set of hyperparameters. Having achieved attractive performance in both speed and accuracy, the idea of detecting on feature map becomes popular. [17] frame



Fig. 1. An intuitive comparison between axis-aligned detection (a) and our proposed method (b). Our method leads to tighter and more precise bounding box for each object, especially when processing image shot with an arbitrary rotation which is commonly observed in optical remote sensing.

detection as a regression problem and simplifies the pipeline by replacing the whole system with a single neural network. [16] utilize multiple feature layers to improve accuracy while keeps time cost low.

CNN is naturally applicable to VHR optical remote sensing image, particularly the RGB aerial image. [19] combine both deep CNN and finite state machine to extract road networks from VHR aerial and satellite imagery. [20] use CNN for per-pixel classification to improve high-level segmentation. [21] work towards to a CNN-based method for vehicle detection. Similarly, [22] employ also exploits deep CNNs to effectively extract features to improve building detection.

However, different from natural scene images, where objects are typically in an upright orientation and thus could be reasonably described by axis-aligned bounding boxes, objects in optical remote sensing images usually pose a diversity of orientations which hampers the use of the off-the-shelf methods. To deal with this problem, [23] propose a rotation-invariant CNN (RICNN) model to advance the performance of object detection in these images where objects with rotation variation dominate the scope. [24] propose ORCNN (Oriented R-CNN) to tackles this problem by applying an additional classifier on the detected regions to get orientations from the six predefined angle classes. [25] exploit RPN by adding multiangle anchor boxes besides the conventional ones to address rotation variations and appearance ambiguity.

In this work, we present a novel method derived from ORCNN and RPN, which predicts buildings orientation in a regression fashion. The model detects buildings in optical remote sensing images with not only locating their pixel coordinates and sizes but also providing their orientation information in the format of oriented bounding box as Fig.1 shows.

We summarize our contributions as follows:

- We propose a novel method derived from RPN which not only locates buildings in optical remote sensing image with bounding boxes but also provides orientation information by regressing their orientation angles.
- A numerical algorithm is also developed for fast estimating Intersection over Union (IoU) between two arbitrarily given rectangles, which is capable of being implemented in GPU allowing accelerated computation.

In Section II, we review some related works and, by comparing then with ours we shed light on innovations in this work. Section III describes the proposed model in detail. Experiments are conducted in Section IV, followed by conclusions in Section V.

II. RELATED WORK

There are some methods involving more sophisticated data to get more accurate estimation which have the ability to handle more complex building shapes. As [26] shows, the author takes point cloud data as the input to extract precise edges. However, point cloud data is dense and requires more efforts to obtain which also poses challenges for processing. In most cases, we use RGB-channel image as input.

In [18], the system firstly choose a number of candidate regions from a series of pre-configured bounding boxes, namely the anchors, by coarsely identifying whether an object is captured by any anchor. This stage is performed within RPN. And then, further classification and regression for each cropped region are simultaneously carried out. In this stage, object is classified into a certain class and a precise relative location is also regressed. For effectively leveraging the extracted features, all predictions are performed on the feature maps.

To estimate the orientation angles, [24] exploit the detection result by partitioning the radius space into 6 bins (i.e., 30, 60, 90, 120, 150 deg) and selecting one to describe the orientation for each finally detected region. In our works, we propose angle anchor box to take orientation into account and by extending the framework of [18] we cast the task into a regression problem. To achieve this goal, a numerical algorithm is developed allowing an estimation of IoU between two arbitrarily oriented rectangles. Besides this, the algorithm is capable of being implemented in GPU for rapid computation.

Recently, [25] design multiangle anchor boxes to handle the problem of rotation variation which is similar to ours. However, the anchor boxes are limited to counteract orientation variation, and thus orientation information is not predicted in the final output. [27] take different framework (SSD) to obtain similar output for vehicle detection. In their work, calculation of IoU overlap between any two arbitrarily oriented rectangles is technically avoided, which comes as our second contribution.

Beyond remote sensing, there is a similar problem in text detection where text region in the given images are not always horizontal which encourages researchers to work for more specific detection methods. Concurrently with our work, [28] and [29] propose to use oriented box for text detection based on CNN.

[28] present Rotation Region Proposal Networks (RRPN) to generate inclined proposals to include angle information. Following the two-stage fashion (e.g., Faster R-CNN [18]), they develop Ration Region of Interest (RRoI) pooling layer to handle the arbitrary-oriented proposals generated by RRPN. In their work, an algorithm for computing IoU between arbitrary-oriented rectangles is also developed. Different from Faster R-CNN, box re-regression has been cut off in their work leaving classifiers for binary classification.

[29] propose Rotational Region CNN (R2CNN) for detecting text on natural images. In their work, anchor keeps axis-aligned when produced by RPN and orientation is taken into account by imposing several pooling operation of different sizes to get concatenated feature used for inclined box regression.

Both of the two works are based on Faster R-CNN and run a binary classification. Similarly, our work also follows this fashion but takes a more light and unified framework. Our model inherits the anchor's generation but we truncate the pipeline to get one-stage detection model. The model extends the predecessor's 4-dimension regression vector to 5-dimension to incorporate orientation regression. In our work, all the 5 elements are simultaneously predicted when the convolution kernel are sliding on the feature map as we emphasize the ability of convolutional layers to locate object on the feature map which coincides with the research of [30]. As a result, no further pooling is needed.

III. PROPOSED METHOD

Our proposed model is illustrated in Fig.2. For feature extraction, the pretrained VGG-16 model [31] is loaded after being cut off its fully connected layers and softmax layer. The convolutional layers take an image of 448×448 pixel size as input to produce a 512-channel 28×28 pixel size feature map. The following two extra convolutional layer branches keeping outputs' dimensionality same to that of the input are added to adapt the network to this task. Network-based detection model usually consists of classification and regression components. The feature map is to be shared by classification (*Conv6_cls*) and regression (*Conv6_reg* and *Conv7_reg*) branches. Each branch has its own layer configurations for its specific task. In classification branch, the last convolutional layer's output is split into two volumes, after which each volume is reshaped to a vector. A softmax is applied on the two to obtain the vector of object score where each element is considered to measure the membership of the content in the corresponding bounding box to the class of building object. The regression branch takes one convolutional layer to assign a 5-dimension vector to each anchor box to get the bounding box. Each vector represents the predicted offset of the target building from the already generated multiangle anchor which will be introduced later.

A. Multiangle Anchor Box Proposal

To incorporate angle information, we further develop RPN to generate oriented anchor boxes in addition to the traditional ones. As Fig.3 illustrates, on the shared feature map we generate our proposed multiangle anchor boxes for each location.

The proposed boxes consist of axis-aligned and oriented anchor boxes. They share a uniform format of (x, y, α, h, w) , where x, y are measured by location of the center on the raw input image; α is the angle between the shorter side and the x axis of the feature map; we denote the longer side as h , subsequently the shorter one, w . To take account of various sizes of buildings, we set different scales to configure anchor boxes' shapes. To avoid ambiguity, we limit the value of orientation angle in $[0, \pi)$.

B. IoU Estimation

The value of IoU is frequently referenced in detection task. Contrary to the axis-aligned boxes, calculating IoU for arbitrarily oriented boxes is to take longer time to be done and requires more complex design in algorithm which may hamper the training and inference process.

To tackle this problem, we develop a numerical algorithm to estimate IoU between two arbitrarily oriented rectangles and the precision could be controlled via a single parameter, named *Fast-IoU*. The algorithm is illustrated in Algorithm1 and Fig.4. The ideas behind the algorithm are using point number to approximate areas and simplifying counting inner points by rotation transformation. As showed in Fig.4, in the first stage, a group of uniformly distributed gird points are generated in a unit

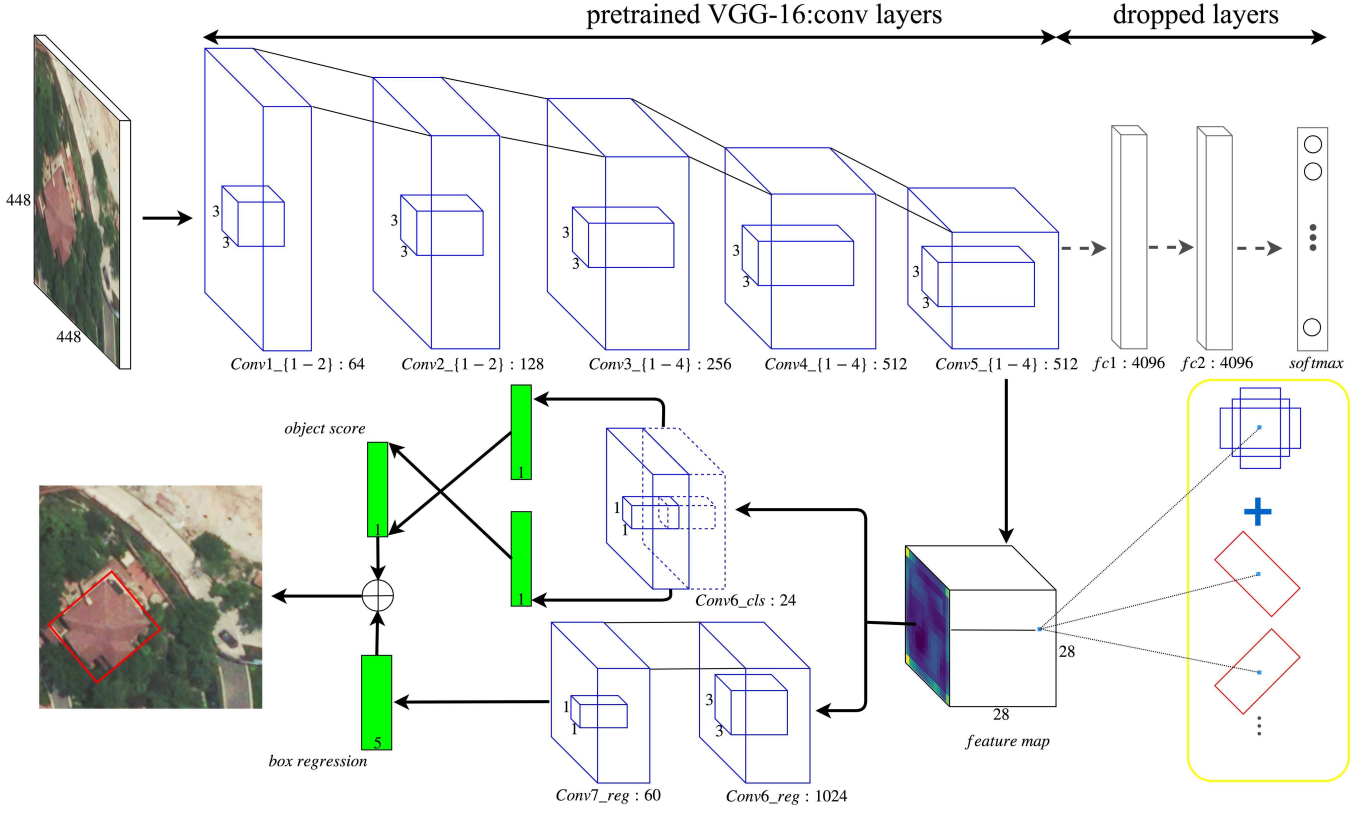


Fig. 2. Architecture of the proposed model. The pretrained VGG-16 model's parameters are used for feature extraction after being cut off its classification layers. The extracted feature map goes through the two extra convolutional layer branches (i.e., *Conv6_cls* for classification branch, *Conv6_reg* and *Conv7_reg* for regression branch). The feature map is shared between the both branches and our proposed multiangle anchors are generated at each pixel in the feature map.

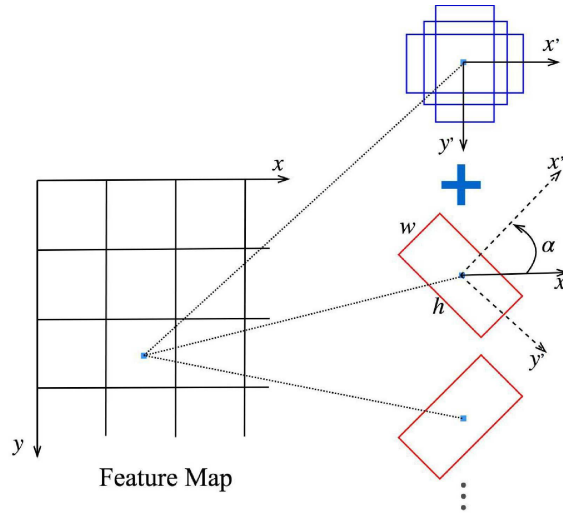


Fig. 3. Illustration of our proposed multiangle anchors. For each location in the feature map we generate a fixed number of anchors that consist of traditional ones (blue boxes) plus oriented anchors (red boxes). The orientation angle is denoted by α .

Algorithm 1: IoU Estimation

Input: $R_1(x_1^r, y_1^r, \alpha_1^r, h_1^r, w_1^r), R_2(x_2^r, y_2^r, \alpha_2^r, h_2^r, w_2^r)$
Output: IoU value

Parameter: N //controls preciseness

- 1: Generate point vector $P(P_1, P_1, \dots, P_{N \times N})$ in which $P_i(x_i, y_i)$ is evenly distributed in $[0.5, 0.5]^2$
 - 2: $P \leftarrow (x_1^r, y_1^r)^T + \text{diag}(w_1^r, h_1^r)M(\alpha_1^r)P$ // rotate, scale and shift P , M denotes Rotation Matrix
 - 3: $(u, v) := M(-\alpha_2^r)(x_2^r, y_2^r)^T$ // rotate $-\alpha_2^r$ to get axis-aligned R_3
 - 4: $P \leftarrow M(-\alpha_2^r)P$ // rotate same angle as R_2
 - 5: $R_3 := (u, v, 0, h_2^r, w_2^r)$ //after this operation R_3 is axis-aligned
 - 6: n : the number of points in P which fall into R_3
 - 7: $I := h_1^r w_1^r n / N^2$ // calculate the intersection area
 - 8: $IoU := I / (h_1^r w_1^r + h_2^r w_2^r - I)$
 - 9: **return** IoU
-

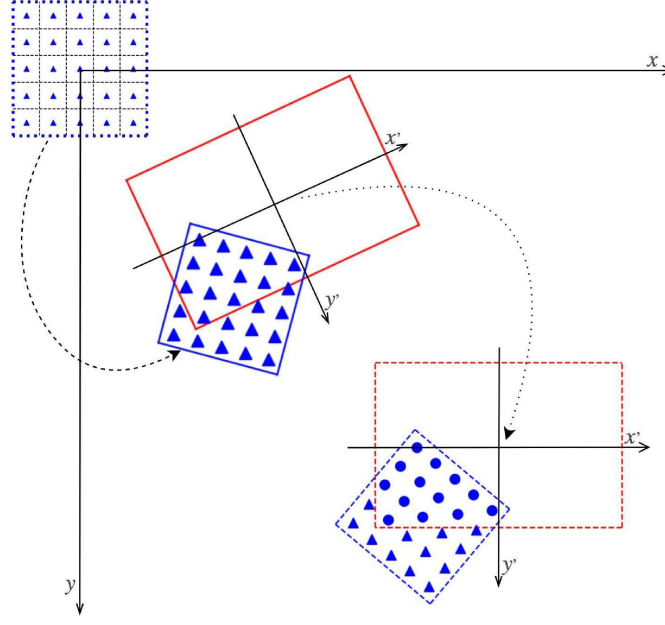


Fig. 4. Illustration of our fast estimation algorithm. To estimate two arbitrarily imposed rectangles (denoted by red and blue solid boxes respectively, center of the figure) we firstly generate grid points (blue triangles, up-left) and transfer them to fit one (triangles, center). After another transformation (down-right) we count the intersection points (circles).

square which are subsequently transferred to fit the inner space of one of the two rectangles by an affine matrix. By applying another affine transformation on the inner points and the other box, we make the box axis-aligned while preserving the whole geometry property, after which counting the inner points of the box becomes straightforward.

With the multiangle anchor mechanism we further illustrate our method in Fig.5. The method generates multiangle anchors on the feature map and uses the *Fast-IoU* to implement *NMS* operation.

C. Training and Inference

To train the proposed model we use loss function defined in [13]:

$$\begin{aligned}
 L(\{p_i\}, \{t_i\}) &= \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, l_i) \\
 &+ \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)
 \end{aligned} \tag{1}$$

where i denotes the index of anchor in training batch and, p_i , t_i represent predicted probability of the anchor being a building object and 5-dimension vector parameterizing geometry properties of the predicted box respectively. l_i and t_i^* are the corresponding ground truths.

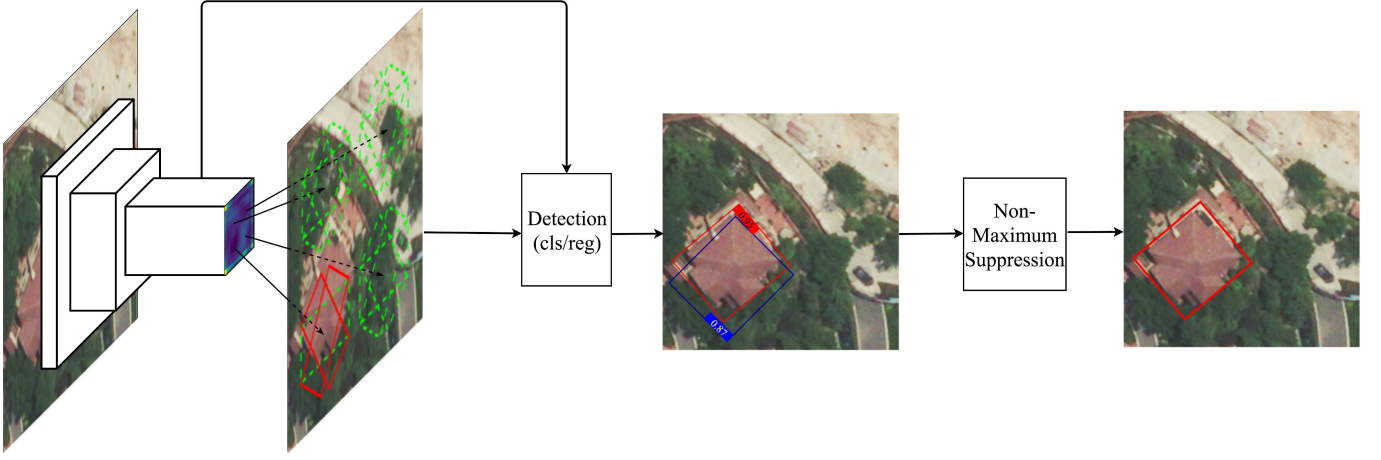


Fig. 5. Pipeline of the proposed method. For a given image, the feature map is obtained through a stack of convolutional layers and oriented anchors are generated for each location of the feature map. Anchors with IOU below the threshold are marked as negative samples (green-dashed boxes). The detection component predicts both class and offset for each anchor. For post processing *non-maximum suppression* is applied.

Cross-entropy is used to evaluate classification loss $L_{cls}(p_i, l_i)$:

$$L_{cls}(p_i, l_i) = -\log p_i^{(l_i)} \quad (2)$$

and for regression loss, we use *smooth* L_1 :

$$L_{reg}(t_i, t_i^*) = \sum_{k \in \{x, y, \alpha, h, w\}} w_k L_1(t_i^{(k)} - t_i^{*(k)}) \quad (3)$$

$$L_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{else} \end{cases} \quad (4)$$

where, w_k is the weight of the k th dimension. Each dimension is defined as:

$$\begin{aligned} t_x &= \frac{x - x_a}{w_a}, & t_y &= \frac{y - y_a}{h_a} \\ t_h &= \log \frac{h}{h_a}, & t_w &= \log \frac{w}{w_a}, \\ t_\alpha &= \alpha - \alpha_a, & t_x^* &= \frac{x^* - x_a}{w_a}, \\ t_y^* &= \frac{y^* - y_a}{h_a}, & t_h^* &= \log \frac{h^*}{h_a}, \\ t_w^* &= \log \frac{w^*}{w_a}, & t_\alpha^* &= \alpha^* - \alpha_a \end{aligned} \quad (5)$$

where subscript of a and superscript of $*$ indicate anchor and ground-truth respectively.

We notice that each dimension is parameterized only by local information, independent from image size, which is essential for adapting the trained model to larger input.

We exploit the strategy in [18] to assign labels to anchors. That is, anchors with highest IoU overlap with any ground-truth box or with IoU exceeding specific threshold are marked as targets. Also, since buildings are sparse in an input image we take sampling strategy from [13] to generate training batch. With the network architecture depicted in Fig.2, our training processing makes the model directly learn to regress oriented bounding box.

Contrary to the huge amount of training samples required for training *VGG-16*, in this task, we are only able to obtain a limited number of manually labeled images. As a result, we take the fine tuning strategy to facilitate the training process by loading the convolutional layers' parameters of *VGG-16*.

In inference stage, we map the outputs (values of the offsets) back to the original image to get rotated bounding boxes and apply NMS for eliminating redundant boxes. The *Fast-IOU*, again, is invoked to calculate IoU among the boxes. The process greedily filters out boxes whose IoUs with their buddies excess the predefined threshold while being associated with lower predicted scores.



Fig. 6. Part of labeled samples for training and evaluating. Each image has 448×448 pixels, and target buildings are marked by red bounding-boxes. Beyond the popular four-dimension, rotation angle is taken into account to locate target building.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we train our proposed model and run the prediction on the test images. We compare the popular axis-aligned method with ours in both subjective and objective ways. Though the experiments are conducted on a relatively small dataset, the results explicitly show that the proposed method outperforms its counterpart in handling optical remote sensing images, more specifically, we use faster R-CNN [18] as the baseline method. We also conduct experiments on 896×896 images and the results show that our model is able to learn to detect building object by local context information and is capable of scaling to variable sizes of input images.

We also conduct experiment for R2CNN [29] which is proposed to address arbitrary-orientated text detection on our building dataset. Results show an approximately equal performance of them two.

A. Dataset Setup

There is a scarcity of off-the-shelf datasets for training and evaluating for this task, which requires orientation labels. To handle this problem, we manually labeled 364 images of 448×448 size, target buildings in each image are enclosed within a rotated bounding box. Some labeled samples are as shown in Fig.6. Each target building is described by a 5-dimension vector: (x, y, α, h, w) where, x, y denote the coordinate of the center, and α denotes the angle from image's x -axis to bonding-box's x -axis.

B. Implementation Details and Parameter Optimization

Within *back-propagation* our model can be trained in an end-to-end fashion and we take stochastic gradient descent (SGD) for optimization. In each updating step, the model takes one image as the input, all anchors are marked as positive, negative or ignored (anchors with ignored label make no effect in model's updating) and we randomly choose 128 anchors to evaluate the loss function. Since targets are sparse in real images we restrict the ratio of the positive and negative samples to 1:1 to avoid overfitting.

For leveraging the CNN's power in feature extraction, we use convolutional layers and the corresponding parameters of the VGG-16 model pre-trained on ImageNet. The regression and classification branches are shallow and randomly initialized for training. However, for the borrowed convolutional layers, there are some noticeable challenges in exploiting the pre-trained parameters. Firstly, there is a considerable difference between ordinary image and remote sensing image: objects in the two have totally different orientation styles which makes feature extraction less effective and thus hurts the performance in both classification and regression; then, as we know, CNN is usually trained for sight rotation invariance which is achieved by installing pooling layers and mirroring training images. However, in this task we aim to output a rotation angle for each target which moves beyond that property. Those encourage us to keep these layers learnable with slowing their learning rates to 1/100 of their following branches' (i.e., classification and regression).

C. Evaluation Metrics

To evaluate the performance of the proposed method, two widely-used measures are used, namely the precision-recall curve (PRC) and average precision (AP).

For *Precision-Recall Curve*, we use TP , FP and FN to denote the number of true-positives, false-positives and false-negatives respectively. Precision measures the fraction of true-positives (TPs) over all samples that are predicted as positives: $Precision = TP / (TP + FP)$. And Recall measures the fraction of TPs over all sample that are labeled as positives: $Recall = TP / (TP + FN)$. The AP measures the average value of precision over the interval ranging from $Recall=0$ to $Recall=1$. Hence, higher curve for PRC and higher value for AP are desired.

Generally, under the topic of object detection, a detection box whose IoU overlap ratio with any ground truth is greater than 0.5 is considered to be a true positive, and otherwise false positives. For several boxes overlapping with the same ground truth we only consider the one of the highest overlap ratio as a true positive.



Fig. 7. Detection results comparison between the axis-aligned (upper row) and our proposed method (bottom row).



Fig. 8. Detection result on 896×896 images. Contrary to the axis-aligned method (left column), our method (right column) keeps better visualization performance for large images.

D. Experiments and Discussions

In this subsection, we aim to demonstrate that with the help of rotated anchor mechanism, our proposed method tends to be more efficient in locating object over its axis-aligned counterpart. The IOU value is calculated by the algorithm proposed in subsection III-B.

Intuitive comparisons of the two methods are showed in Fig.1 and Fig.7. Because of the introduction of the angle regression, our method theoretically outputs tighter and precise bounding boxes. The proposed method also acquires ability to better estimate information of objects near the boundary; and notably, as Fig.1 shows, while there is only one building target in the left-bottom corner being detected by the axis-aligned, the proposed method detects two and separates them precisely as the model is able to learn a joint distribution which allows getting optimal shape of bounding box according to the predicted orientation.

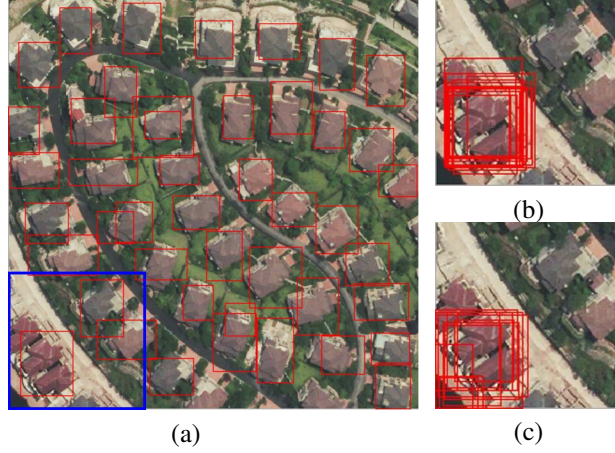


Fig. 9. Detection result in different stages of the axis-aligned method. The final result (a) reports only one building object in the left-bottom region (denoted by blue box), and the related output without NMS (b) clearly shows that the model coarsely encloses the two buildings as one unit, similar to the region proposals (c) produced by the RPN.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT METHOD/HYPERPARAMETER IN TERM OF mAP VALUES.

Method	Scales of Anchor	Angles of Anchor (deg)	mAP
AA	$64^2, 128^2, 256^2$	-	0.68
AG	$60^2, 90^2, 130^2$	-60,0,60	0.93
AG-1	$64^2, 128^2, 256^2$	-60,0,60	0.81
AG-2	$64^2, 128^2, 256^2$	0	0.74
AG-3	$60^2, 90^2, 130^2$	0	0.77
R2CNN	-	-	0.94

We extend our experiment to images containing more dense regions to get a more comprehensive comparison. The results are showed in Fig.1 and Fig.8. Although the images tend to contain more objects and present a more complex environment for detection task, our method still obtains a better visualization performance over its counterpart. Compared to the axis-aligned method, the proposed method shows more advanced detection in handling dense regions where buildings, usually, with orientation, are closer to each other increasing the difficulty of detecting. In the axis-aligned method, the model is limited to predict coarse location for target with orientation due to the lack of orientation regression component. Consequently, the method is more likely to suffer from dense building regions by roughly clustering them together. As Fig.9 shows, the axis-aligned method inappropriately treats the two building objects as one in the region marked by blue box. To study this case, we turn to the output of the network, as (Fig.9(b)) shows, which is free from *NMS* and observe that the two buildings have already been mixed up. We then, further investigate the region proposals of the *RPN*. As Fig.9(c) shows, the *RPN* produces similar response to that region. In both of the two stages, the model puts more detection boxes around the center of the two building objects spanning exactly to the boundary of them two indicating the model takes the two as a single object.

Meanwhile, as Fig.8 and Fig.9 show, we notice that the axis-aligned method is more likely to produce redundant detections. This is not only caused by inappropriate threshold of the *NMS* but the lack of orientation regression as we also observe there is a considerable amount of overlaps between object boxes in dense building regions which, if lower *IOU* threshold in *NMS*, tends to result in miss detection.

As the detection result shown in Fig.1(b), the convolutional layers have the ability to preserve orientation information. And, by introducing orientation regression loss we force the model to learn to acquire it.

We train our model with different anchors settings and evaluate the performance using the two measures. As TABLE I and Fig.10 show, our method achieves significant improvement compared with the axis-aligned method. We own this to the additional orientation regression, because of which the model is able to simultaneously predict the orientation, location and size. And the higher utilization of bounding box with respect to *IOU* makes it more likely to obtain better performance in terms of mAP value.

E. Comparison with Other Similar Method

In parallel with our work, there are some researches which also target the problem of orientation not for building in remote sensing but text in natural scene. To have a comparison with the similar work, we transfer R2CNN model in [29] from text detection to our dataset. and evaluate the two methods on the data.

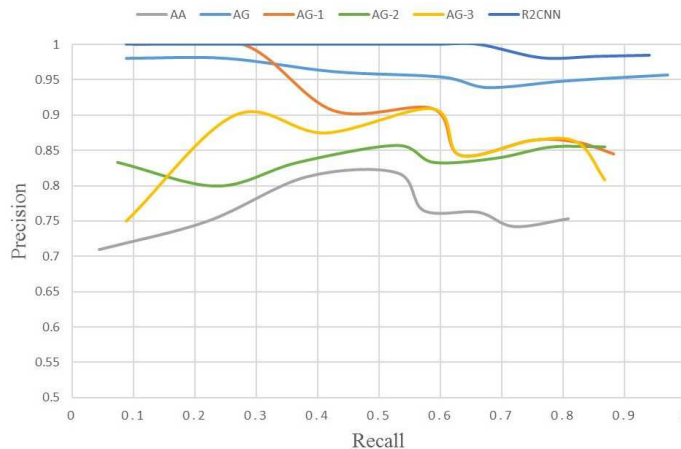


Fig. 10. Precision-recall curves (PRCs). We compare our method under different settings with the axis-aligned method.

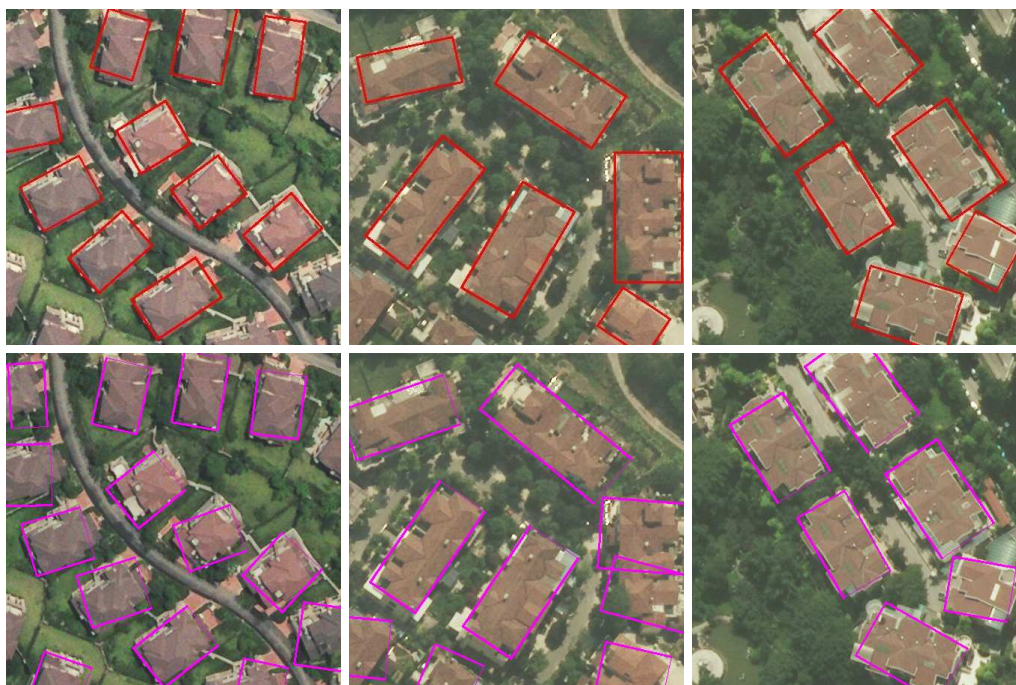


Fig. 11. Detection results of R2CNN method (upper row) and ours (bottom).

Fig.11 shows detection results for the methods. Both of the methods show similar performances in locating buildings and estimating orientation. In detail, R2CNN tends to obtain more precise detection as it takes two-stage framework for refining prediction. However, our method is more relying on local information since the model uses sliding kernel to regress the boxes, and as a result our method tends to discover more buildings in image margins which can be considered as partial occlusion.

The performance of R2CNN is measured in the last row of TABLE I and the PR curve is drawn in Fig.10. As the results show the two methods are almost neck and neck in terms of mAP. Although R2CNN obtains a slightly better score, our model takes a more unified framework and is more compact.

In TABLE I, we use different settings to explore the validity of the proposed detection model as well as multiangle anchor. Comparison between AA and AG-2 directly demonstrates the value of detecting with orientation as the settings of $0\ deg$ for the angle of the anchors amounts to generating identical anchors for the two methods.

AG with AG-3 aims to test the validity of the proposed multiangle anchors. The higher score achieved by the former suggests that with properly adding angles of the proposed orientation anchors detection would have a significant improvement.

V. CONCLUSION

In this work, we aim to bridge the gap between the popular CNN-based axis-aligned object detection method and the orientation variation of buildings that generally exists in remote sensing images. To tackle this, we introduce orientated anchors

and additional dimensionality to regress orientation for each building object. To address the problem of calculating IoU overlap between any two arbitrarily oriented boxes, we develop an algorithm for estimating which is feasible to implement in GPU for fast computing. To train and test our model, we construct a dataset of remote sensing images and manually label building objects. Our method is implemented in an end-to-end fashion and tends to produce tighter bounding boxes of higher IoU overlap with building objects which leads to better performance over the axis-aligned one in terms of mAP. Also, we compare our method with the related work proposed for rotational text region detection. They achieve closed scores but differ at structure. Our model presents an unified and compact framework, however it errors of orientation estimation and shows sensitive to hyperparameter settings. Hence, we will take more efforts on obtaining more precise orientation estimation in our future study.

REFERENCES

- [1] D. Brunner, G. Lemoine, and L. Bruzzone, "Earthquake Damage Assessment of Buildings Using VHR Optical and SAR Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2403–2420, May 2010.
- [2] R. Bonnefon, P. Dhrt, and J. Desachy, "Geographic information system updating using remote sensing images," *Pattern Recognition Letters*, vol. 23, no. 9, pp. 1073 – 1083, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167865502000545>
- [3] 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, Jul. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271616300144>
- [4] K. Stankov and D. C. He, "Detection of Buildings in Multispectral Very High Spatial Resolution Images Using the Percentage Occupancy Hit-or-Miss Transform," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 10, pp. 4069–4080, Oct. 2014.
- [5] S. Ahmadi, M. V. Zoej, H. Ebadi, H. A. Moghaddam, and A. Mohammadzadeh, "Automatic urban building boundary extraction from high resolution aerial images using an innovative model of active contours," *International Journal of Applied Earth Observation and Geoinformation*, vol. 12, no. 3, pp. 150–157, Jun. 2010. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0303243410000115>
- [6] H. Shi, L. Chen, F. k. Bi, H. Chen, and Y. Yu, "Accurate Urban Area Detection in Remote Sensing Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1948–1952, Sep. 2015.
- [7] H. G. Akay and S. Aksoy, "Automatic Detection of Compound Structures by Joint Selection of Region Groups From a Hierarchical Segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3485–3501, Jun. 2016.
- [8] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object Detection in Optical Remote Sensing Images Based on Weakly Supervised Learning and High-Level Feature Learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/6991537/>
- [9] G. Cheng, J. Han, L. Guo, X. Qian, P. Zhou, X. Yao, and X. Hu, "Object detection in remote sensing imagery using a discriminatively trained mixture model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 85, pp. 32–43, Nov. 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0924271613001809>
- [10] J. P. Cohen, W. Ding, C. Kuhlman, A. Chen, and L. Di, "Rapid building detection using machine learning," *Applied Intelligence*, vol. 45, no. 2, pp. 443–457, Sep. 2016. [Online]. Available: <https://link.springer.com/article/10.1007/s10489-016-0762-6>
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: <http://www.nature.com/articles/nature14539>
- [13] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*, 2014.
- [15] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2980–2988.
- [16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision ECCV 2016*, ser. Lecture Notes in Computer Science. Springer, Cham, Oct. 2016, pp. 21–37.
- [17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [18] 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 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>
- [19] J. Wang, J. Song, M. Chen, and Z. Yang, "Road network extraction: a neural-dynamic framework based on deep learning and a finite state machine," *International Journal of Remote Sensing*, vol. 36, no. 12, pp. 3144–3169, Jun. 2015. [Online]. Available: <https://doi.org/10.1080/01431161.2015.1054049>
- [20] M. Lngkvist, A. Kiselev, M. Alirezaie, and A. Loutfi, "Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks," *Remote Sensing*, vol. 8, no. 4, p. 329, Apr. 2016. [Online]. Available: <http://www.mdpi.com/2072-4292/8/4/329>
- [21] B. Li, T. Zhang, and T. Xia, "Vehicle Detection from 3d Lidar Using Fully Convolutional Network," *arXiv:1608.07916 [cs]*, Aug. 2016, arXiv: 1608.07916. [Online]. Available: <http://arxiv.org/abs/1608.07916>
- [22] K. Nemoto, R. Hamaguchi, M. Sato, A. Fujita, T. Imaizumi, and S. Hikosaka, "Building change detection via a combination of CNNs using only RGB aerial imageries," vol. 10431. International Society for Optics and Photonics, Oct. 2017, p. 104310J. [Online]. Available: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10431/104310J/Building-change-detection-via-a-combination-of-CNNs-using-only/10.1117/12.2277912.short>
- [23] G. Cheng, P. Zhou, and J. Han, "Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
- [24] C. Chen, W. Gong, Y. Hu, Y. Chen, and Y. Ding, "Learning Oriented Region-based Convolutional Neural Networks for Building Detection in Satellite Remote Sensing Images," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-1/W1, pp. 461–464, May 2017. [Online]. Available: <http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-1-W1/461/2017/>
- [25] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-Insensitive and Context-Augmented Object Detection in Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, no. 99, pp. 1–12, 2017.
- [26] M. Awrangjeb, "Using point cloud data to identify, trace, and regularize the outlines of buildings," *International Journal of Remote Sensing*, vol. 37, no. 3, pp. 551–579, Feb. 2016. [Online]. Available: <http://www.tandfonline.com/doi/full/10.1080/01431161.2015.1131868>
- [27] Tianyu Tang, Shilin Zhou, Zhipeng Deng, Lin Lei, and Huanxin Zou, "Arbitrary-Oriented Vehicle Detection in Aerial Imagery with Single Convolutional Neural Networks," *Remote Sensing*, vol. 9, no. 11, p. 1170, Nov. 2017. [Online]. Available: <http://www.mdpi.com/2072-4292/9/11/1170>
- [28] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov 2018.

- [29] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2cnn: Rotational Region CNN for Orientation Robust Scene Text Detection," *arXiv:1706.09579 [cs]*, Jun. 2017, arXiv: 1706.09579. [Online]. Available: <http://arxiv.org/abs/1706.09579>
- [30] K. Lenc and A. Vedaldi, "R-cnn minus r," in *Proceedings of the British Machine Vision Conference (BMVC)*, M. W. J. Xianghua Xie and G. K. L. Tam, Eds. BMVA Press, September 2015, pp. 5.1–5.12. [Online]. Available: <https://dx.doi.org/10.5244/C.29.5>
- [31] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Sep. 2014, arXiv: 1409.1556. [Online]. Available: <http://arxiv.org/abs/1409.1556>