Arbitrary-angle bounding box based location for object detection in remote sensing image

Fei Sun, Huanyi Li, zhiyang Liu, Xinyue Li and Zhize Wu

ABSTRACT
Object location is a fundamental yet challenging problem in object detection. In the remote sensing image, different imaging projection directions make the same object have various rotation angles, and in some scenes, the object distribution is relatively dense. Most of the existing deep learning-based object detection algorithms utilize horizontal bounding box to locate objects, which causes inaccurate location of the objects with dense distribution or arbitrary direction, thus leading to the detection misses. In this paper, we propose an arbitrary-angle bounding box based location and embed it into the Faster R-CNN, developing a new framework called Rotated Faster R-CNN (R-FRCNN) for object detection in remote sensing image. In R-FRCNN, we specially improve anchor ratios to adapt to the objects like ship with large aspect ratio and increase the weights of the horizontal bounding box regression to reduce the interference of the arbitrary-angle bounding box on the horizontal bounding box prediction. Comprehensive experiments on a public dataset and a self-assembled dataset (which we make publically available) show the superior performance of our method compared to standalone state-of-the-art object detectors.

Introduction
With the continuous development of modern remote sensing technology, a large number of remote sensing images with higher spatial resolution and richer content have emerged, providing important analysis condition and resources for the research on various fields (Ye et al., 2020). Object recognition from images means to automatically find the object(s) of interest and returning their category and location information. Due to the continuous improvement of the spatial resolution of satellite images, the information contained in single images is growing. The currently existing automated image interpretation approaches are unable to meet the needs of many real-world applications in the remote sensing community (Wu et al., 2020). Therefore, the question of how to accurately and quickly extract the position and type of objects from satellite images has come into the focus of research.

Object detection from satellite images is widely used in civil (Chen et al., 2015) and military fields (Zhong et al., 2018). Among the many possible objects to detect, airports, ports, oil depots, ships, and airplane are usually key concerns. The recognition of such objects can be used for airport flight analysis and adjustment, for assistance to rescue ships in distress, for port and maritime traffic monitoring, and it can also provide data for urban planning (Zhong et al., 2018, Hua et al., 2020, Khan et al., 2019, Orengo et al., 2017, Hu et al., 2019). It also plays an important role in modern military systems, providing information for combat deployment, for monitoring the dynamics of airports, ports, and sea areas, and for analyzing the enemy combat strength (Wu et al., 2020, Bachagha et al., 2020, Melillos et al., 2018). In summary, object recognition from satellite images is an important research topic and has broad application prospects.

Unlike close-range imaging, each satellite image covers a wide area and has a complex background as well as different projection directions. Hence, object detection faces two challenges: (1) Satellite images are shot downward from high altitude. Different projection directions lead to different rotation angles of the same object, resulting in morphological differences. Large object rotation angles can cause loss and misjudgment in object detection. (2) Some scenarios exhibit high object densities, e.g., aircraft in airports, oil depots, and ships in ports. Closely located objects often interfere with each other in detection algorithms, making it difficult to accurately locate and easy to lose them.

Due to the ability to learn high-level semantic features, deep learning and especially Convolutional Neural Networks (CNNs) (LeCun et al., 2015) have been widely and successfully used for object detection. Existing deep learning-based object detection algorithms mostly adopt horizontal bounding boxes to locate the objects, and perform non-maximum
suppression processing on all bounding boxes output by the network. The purpose of non-maximum suppression is to remove the redundant bounding box of a certain object, so as to retain the optimal bounding box as the detection result of the object. However, when there are some densely distributed objects in an inclined state, the non-maximum suppression operation not only filters the redundant bounding box of a certain object itself, but also may filter the bounding boxes of other objects, thereby removing many useful bounding boxes, which results in a poor object detection performance on satellite images.

Especially in the dense distribution of objects, such detection misses occur frequently. As shown in Figure 1: (a) The ships show a nearly horizontal direction with a slight inclination, the overlap between most object bounding boxes is relatively small, but there are still a few of the bounding boxes with a large amount of overlap, resulting in one ship being missed; (b) The ships show a large inclination angle, the overlap between some object bounding boxes is more, which is caused by non-maximum suppression filtering, so two ships are missed.

In recent years, due to the complexity of text distribution in natural scenes, text detection algorithms are increasingly using arbitrary-angle bounding boxes for detection, and a large number of excellent text detection algorithms, such as SegLink (Shi et al., 2017), R2CNN (Jiang et al., 2017), RRPN (Ma et al., 2018), etc., have been produced. Among them, R2CNN (Jiang et al., 2017) is a milestone for text detection. Its basic framework is based on the Faster R-CNN. By adding a layer of arbitrary-angle bounding box prediction layer at the end of the Faster R-CNN network, R2CNN can jointly predict the coordinate information of arbitrary angle bounding box. As there is no need to change the main framework of the Faster R-CNN, R2CNN is simple and efficient, and it is widely used in the field of text detection.

With the idea of the R2CNN, we propose an arbitrary-angle bounding box based object location algorithm and embed it into the Faster R-CNN for object detection in remote sensing image. We call this new framework Rotated Faster R-CNN (R-FRCNN). Based on the Faster R-CNN, R-FRCNN introduces the arbitrary-angle bounding box to locate objects without redundancy in the remote sensing image by two improvements. On the one hand, in the anchor generation stage, adding a larger proportion of anchor points to generate candidate boxes with a larger aspect to adapt to some objects with a larger aspect ratio in the remote sensing image. On the other hand, an arbitrary angle bounding box prediction layer is added to the last layer of the object detection network, and different weight information is assigned to the prediction of the horizontal bounding box and the arbitrary angle bounding box. We additionally build a remote-sensing image datasets, namely, Tiny Object Detection for Remote Sensing Images (TODRS) (Sun & Wu, 2020). It consists of 4615 images with aircraft, ships, and oil depots, which are labeled with arbitrary angle bounding boxes.

The contributions of this paper are:

1. We propose an arbitrary-angle bounding box based object location method and add anchor ratios to fit objects like ships with large aspect ratio. With this, we improve the location accuracy of densely distributed and arbitrarily directed objects.

2. We embed the new object location into the Faster R-CNN and obtain the Rotated Faster R-CNN (R-FRCNN) for object detection in remote sensing images.

3. We construct, present, and publish the dataset TODRS, to aid the development of robust methods for object detection from remote sensing images.

Figure 1. Type your title here. Obtain permission and include the acknowledgment required by the copyright holder if a figure is being reproduced from another source. Dense ship detection results using the Faster R-CNN (Ren et al., 2017).
The rest of this paper is organized as follows. Section 2 provides a brief overview of the related work on object detection from remote sensing image. In Section 3, we introduce our novel object location algorithm along with a thorough theoretical analysis. Then, Section 4 details the performance evaluation and discussion on two object detection dataset of remote sensing image. Finally, the conclusions and an outlook on future work are discussed in Section 5.

Related work
In recent years, with the development of feature extraction and machine learning-based classification, object detection is commonly regarded as a classification problem. Figure 2 shows the basic flow chart of object detection in remote sensing image, which is mainly divided into the following three steps:

(1) Region proposal generation: A strategy is used to generate several object candidate regions in the remote sensing image. Commonly used candidate regions generation methods include sliding window search (Felzenswalb et al., 2009; Zhang et al., 2013; Zhu et al., 2010) and region of interest (ROI) selection (Harel et al., 2007; Hou & Zhang, 2007; Itti et al., 1998).

(2) Feature extraction for candidate region: Extract corresponding features from the generated candidate region of the object. Commonly used features are Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), Scale Invariant Feature Transform (SIFT) (Lowe, 2004), and Bag of Words (BOW) (Csurka et al., 2004).

(3) Object classification: The feature vectors of each candidate region are input into a classifier for classification. Commonly used classifiers are K-Nearest Neighbor (KNN) (Cover & Hart, 1967), Support Vector Machine (SVM) (Vapnik, 2013), Adaboost (Freund, 1995; Freund & Schapire, 1996), Conditional Random Field (CRF) (Lafferty et al., 2001), Sparse Representation (Wright et al., 2008) etc.

In recent years, deep learning is a new technology emerging in the field of artificial intelligence. Among them, CNNs (LeCun et al., 2015) because of their powerful learning ability and feature expression ability have achieved better results in many applications of computer vision compared with traditional methods. In the field of object detection in natural scene, the representative of breakthrough is Region-based Convolutional Neural Network (R-CNN) (Girshick et al., 2015), which uses Selective Search (Uijlings et al., 2013) to obtain a group of region proposals, that is, to generate a group of candidate regions. Then, use the CNN to extract the features of these region proposals, and finally use SVM to classify them. R-CNN is the first successful attempt of deep learning in object detection, and its detection accuracy far exceeds the traditional detection algorithms. Since then, a lot of deep learning-based object detection algorithms have emerged, which are mainly divided into two categories: two-stage object detection and one-stage object detection.

In response to the problem of repeated feature extraction and inaccurate object location of candidate regions, Fast R-CNN (Girshick et al., 2015) takes the entire image as the CNN input to obtain the corresponding feature map, and then maps the generated candidate regions to feature maps using RoI pooling operation. This way, a large number of repeated feature extraction calculations are avoided, and the time for object detection is reduced. At the same time, the Fast R-CNN uses the softmax layer for classification and bounding box regression, which improves the accuracy of object detection. However, the Fast R-CNN still use Selective Search to generate regional proposals, which is still very time-consuming. In order to solve this problem, Faster R-CNN (Ren et al., 2017) introduced a region proposal network (Region Proposal Network, RPN) based on Fast R-CNN to automatically generate a small number of highly efficient candidate regions, greatly reducing the time consumption and obtaining a higher detection accuracy than the R-CNN and the Fast R-CNN. So far, the two-stage target detection algorithm is carried out in the way of “RPN+CNN”. Faster-RCNN has also become the most effective representative of the two-stage object detection algorithm. Subsequent two-stage object detection algorithms are basically based on the improvement of the Faster-RCNN, such as FPN (Lin et al., 2017), mask R-CNN (He et al., 2017) and so on.

Figure 2. Object detection based on machine learning in remote sensing image.
In addition, the representative approaches for one-stage object detection are YOLO (Redmon et al., 2016) and SSD (Liu et al., 2016). Such approaches regard the object detection as a regression problem, and the most classic approach is YOLO. The YOLO approach uses a single end-to-end network to directly divide the entire image into a fixed grid. The YOLO predicts candidate bonding boxes, location confidence, and all categories’ probabilities for all grids at once and completes the end-to-end process of the input from the original image to the output of the object category and location. Since there is no need to generate candidate regions, the one-stage object detection algorithm is faster than the two-stage object detection algorithm, but the blind meshing also makes the one-stage object detection algorithm detection accuracy lower.

Encouraged by the successful application of deep learning methods on natural scene images, several researchers have tried to use them for detecting objects from remote sensing images. (Cao et al., 2016) and (Yao et al., 2017) used the R-CNN and the Faster R-CNN, respectively, and found that this greatly improved the performance of object detection compared with traditional algorithms in this field. (Deng et al., 2017) proposed the Enhanced Faster R-CNN (E-FRCNN), which uses multi-layer network feature information for feature extraction and further improves the performance. In (Wu et al., 2020), Wu et al. provide a performance analysis of state-of-the-art aircraft type recognition and deep learning approaches on remote sensing images. A large-scale and publicly available object detection benchmark dataset is proposed in (Li et al., 2020). This new dataset can help the earth observation community to further explore and validate deep learning-based methods.

These success stories proved the feasibility of deep learning-based methods in this field. At the same time, due to the different imaging principles behind remote sensing and natural scene images, there are still many problems when directly applying deep learning to remote sensing images.

The one challenging problem is how to locate objects in satellite images, as they can appear directed in any angle and are often not aligned with the image axes. Also, in some scenes, the object distribution is relatively dense.

Existing deep learning-based algorithms mostly use horizontal bounding boxes to locate objects. As a result, the overlapping bounding boxes of dense objects lead to interference: In the subsequent non-maximum suppression processing, many useful bounding boxes will be removed and the objects they contain are overlooked.

In (Chen et al., 2019), expect for minimizing the classification error, Cheng et al. impose a rotation-invariant regularizer and a Fisher discrimination regularizer on the FC7 layer of VGGNet-16 (Simonyan & Zisserman, 2014) to enforce the CNN features to be rotation-invariant and have powerful discriminative capability. In (Li et al., 2020), an unified object detection framework is proposed for combining the RPN and the contextual feature fusion network to extract the proposals and to simultaneously locate the geospatial objects. By adding anchors of multiangle in RPN, the proposed framework can effectively handle the problem of geospatial object rotation variations.

Differing from the above approaches, this article introduces an arbitrary-angle bounding box based object location method, and embeds it into the Faster R-CNN detection framework. By adding the corresponding anchor point ratio setting to adapt to the target with relatively large length and width, and by increasing the weight of the horizontal bounding box regression, the interference of the bounding box at any angle to the prediction of the horizontal bounding box is reduced, so that the network can better locate remote sensing image targets.

Proposed method

We illustrate the framework of the proposed R-FRCNN in Figure 3. The compositions of FEN and RPN are the same as that of the Faster R-CNN. Differing from the Faster R-CNN, we introduce a new object location based on arbitrary-angle boxes for the characteristics of objects in remote sensing image. Especially, we change the setting of the anchor point in the RPN, and add an arbitrary-angle bounding box prediction layer at the last layer of the ODN to perform coordinate regression of the arbitrary-angle bounding box. We will describe the main idea and specific process of the R-FRCNN in detail below.

Arbitrary-angle bounding box location

As shown in Figure 4, (a) is the commonly used object location based on horizontal boundary box. The coordinates of the object boundary box are, where represent the coordinates of the center point of the rectangular boundary box, respectively, the width and height of the rectangular boundary box; (b) represents the object location based on arbitrary-angle bounding box. The coordinates of the object bounding box, where represents the center point coordinate of the rectangular box, represent the long side and short side of the rectangular bounding box, and represent the counterclockwise rotation angle of the long side and horizontal line of the rectangular bounding box.

Obviously, once the object is tilted relative to the horizontal direction, the horizontal bounding box based object location contains more redundant regions in addition to the object itself and at the same time leads to more overlap between bounding boxes of dense objects. However, arbitrary-angle bounding box
based object location happens to include the objects, there are not too many redundant areas, and the boundary boxes of dense objects do not interfere with each other, and there is less overlap. Therefore, the arbitrary-angle bounding box based object location is more suitable for densely distributed object with arbitrary directions in remote sensing images.

**Proportional of anchor points**

In the R-FRCNN, the arbitrary-angle bounding box is used to locate objects. The intuitive idea is to generate multiple anchor points with different rotation angles in the RPN. However, this method needs to set several different angles, which not only greatly increases the number of anchor points generated, but also can not include all angles. When the angle is not proper, it can not adapt to the object of any angle.

Therefore, this method is not suitable for arbitrary-angle object in remote sensing image. In fact, for the object, its arbitrary-angle bounding box and horizontal bounding box are not independent of each other. They are actually a relationship of inclusion and inclusion. As shown in Figure 5, the object locations in the horizontal bounding box are as follows: (a) The object is located in the horizontal direction of the horizontal bounding box; (b) The object is located in the vertical direction of the horizontal bounding box; (c) The object is located in the diagonal direction of the horizontal bounding box. In (a) (b) two cases, the arbitrary angle bounding box and the horizontal bounding box of the object are overlapped, while in (3) case, the arbitrary angle bounding box of the object is included in the diagonal of the horizontal bounding box. Therefore, considering the connection between the object’s arbitrary angle bounding box and horizontal bounding box, the algorithm still generates a horizontal bounding box at the RPN stage.

In the original Faster R-CNN, there are three different aspect ratios (1:1, 1:2, 2:1) and three different scales (64*64, 128*128, 256*256). According to this setting, for each pixel on the feature map, nine anchor

---

Figure 3. Framework of the R-FRCN.

![Framework of the R-FRCN](image)

Figure 4. Comparison of locations between horizontal bounding box and arbitrary-angle bounding box.

(a) Horizontal Bounding Box  (b) Arbitrary-angle Bounding Box

![Comparison of locations between horizontal bounding box and arbitrary-angle bounding box](image)
points centered on the current pixel are generated. In the R2CNN, different ROI pool sizes are used to pool the generated horizontal bounding box for capturing the features with large aspect ratio. The reason why this method is adopted is that the text object in the natural scene often does not have a fixed aspect ratio. Therefore, it is difficult to design a more appropriate anchor point ratio for the text object. However, the object in remote sensing images (such as aircraft, ships, oil depots, etc.) has a relatively fixed aspect ratio. With this, we consider designing anchor point ratios of different sizes to suit the object aspect ratio of remote sensing images.

For most objects in remote sensing images, such as airplanes, oil depots, baseball fields, tennis courts, basketball courts, track and field fields, and vehicles, the aspect ratio is usually 1:1, 2:1 or close to 1:1, 2:1 proportion. Ships usually have a large aspect ratio and are densely distributed, so this approach focuses on the aspect ratio of ships. Ships usually include aircraft carriers and warships. Table 1 counts the size of some aircraft carriers, and Table 2 counts the size of some warships. According to Tables 1 and Tables 2, the ratio of ships is between 3:1, 10:1. Obviously, we cannot set all the ratios, otherwise, the excessive number of anchor points will be generated and it will be very time-consuming. Therefore, in order to adapt to the

Table 1. Size of some aircraft carriers.

| Aircraft Carrier Name | Length (m) | Width (m) | Aspect Ratio |
|------------------------|------------|-----------|--------------|
| USS John F. Kennedy    | 323.6      | 93.6      | 3.46         |
| USS Kitty Hawk         | 323.6      | 93.6      | 3.46         |
| USS Harry S. Truman    | 332.8      | 76.8      | 4.33         |
| USS Abraham Lincoln    | 333        | 76.8      | 4.34         |
| USS Harry S. Truman    | 332.9      | 76.8      | 4.33         |
| USS Enterprise         | 342.3      | 76.8      | 4.46         |
| USS Yorktown           | 246.7      | 33.4      | 7.39         |
| USS Nimitz             | 332.8      | 76.4      | 4.36         |
| USS Ronald Reagan      | 332.9      | 78.3      | 4.25         |
| USS Carl Vinson        | 335.6      | 77.1      | 4.35         |
| USS Dwight D. Eisenhower| 332.9     | 76.8      | 4.33         |
| USS George Washington  | 333        | 76.8      | 4.34         |
| Liaoning               | 304.5      | 75        | 4.06         |
| Minsk                  | 273.1      | 49.2      | 5.55         |
| Kiev                   | 274        | 51        | 5.37         |
| Charles de Gaulle      | 261.5      | 64.4      | 4.06         |
| Clemenceau-class       | 265        | 51.2      | 5.18         |
| Admiral Kuznetsov      | 305        | 70        | 4.36         |
| Cavour                 | 235.6      | 39        | 6.04         |
| INS Viraat             | 226.9      | 39.6      | 5.73         |
| Príncipe de Asturias   | 195.2      | 29        | 6.74         |

(c) Diagonal Direction

Figure 5. Example of object locations within the horizontal bounding box.
Table 2. Size of some warships.

| Warship Category | Ship Class    | Length (m) | Width (m) | Aspect Ratio |
|------------------|--------------|------------|-----------|--------------|
| Frigates         | Abukuma      | 109        | 13.4      | 8.13         |
|                  | Oliver Hazard| 138        | 13.7      | 10.07        |
|                  | Perry        | 142.9      | 17.5      | 8.68         |
| Cruiser          | Ticonderoga  | 172.8      | 16.8      | 10.29        |
|                  | Whidbey Island| 185.6    | 25.6      | 7.25         |
|                  | Austin       | 173.8      | 30.5      | 5.70         |
|                  | San Antonio  | 208.5      | 31.9      | 6.54         |
|                  | Tarawa       | 254.2      | 40.2      | 6.32         |
|                  | Wasp         | 253.2      | 32        | 7.91         |
| Destroyer        | Shirane      | 159        | 17.5      | 9.09         |
|                  | Hatakaze     | 150        | 16.4      | 9.15         |
|                  | Haruna       | 153        | 17.5      | 8.74         |
|                  | Murasame     | 151        | 17.4      | 8.68         |
|                  | Tachikaze    | 142.9      | 14.3      | 9.99         |
|                  | Kongō        | 161        | 21        | 7.67         |
|                  | Hatsuyuki    | 130        | 13.6      | 9.56         |
|                  | Takanami     | 151        | 17.4      | 8.68         |
|                  | Asagiri      | 137        | 14.6      | 9.38         |
|                  | Arleigh Burke| 153.8      | 20.4      | 7.54         |
|                  | Atago        | 165        | 21        | 8.68         |

The aspect ratio of ships and take into account the time performance, we add four additional proportions: (1:4, 1:8, 1:1, 4:1). With this, each pixel on the feature map generates 21 anchors centered on the current pixel.

Network training

The loss function of the RPN consists of the classification loss \( L_{cls}(p_i, p_i^*) \) and the bounding box regression loss \( L_{reg}(t_i, t_i^*) \), as defined in Equation (1).

\[
L_{RPN}(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N_{reg}} \sum_i p_i L_{reg}(t_i, t_i^*)
\]  

(1)

The classification loss is a binary cross-entropy loss, which is defined as Equation (2), and the position regression loss is a Smooth_L1 loss, which is defined as Equation (3).

\[
L_{cls}(p_i, p_i^*) = -\log(p_i) 
\]  

(2)

\[
L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2, & |t_i - t_i^*| \leq 1 \\ |t_i - t_i^*| - 0.5, & \text{otherwise.} \end{cases}
\]  

(3)

In Equation (1), \( i \) represents the \( i \)-th sample in a mini-batch, \( p_i \) represents the classification prediction value of the sample, and \( p_i^* \) represents the true category corresponding to the sample. If the sample is positive, then \( p_i^* = 1 \), otherwise \( p_i^* = 0 \). For location prediction, a total of four values need to be predicted, i.e. \((x, y, w, h)\), where \((x, y)\) is the coordinate of the center point of the bounding box, and \((w, h)\) is the width and height of the bounding box. \( t_i \) is the offset of the position of the bounding box of the sample, and \( t_i^* \) is the offset of the position of the object bounding box corresponding to the sample. The definitions of \( t_i \) and \( t_i^* \) are shown in Equation (4). Only when the sample is a positive sample, there will be a position regression loss. For a negative sample, only the classification loss needs to be calculated.

\[
t_x = (x - x_a)/w, \quad t_y = (y - y_a)/h,
\]

\[
t_w = \log(w/w_a), \quad t_h = \log(h/h_a),
\]

\[
t_x^* = (x^* - x_a)/w, \quad t_y^* = (y^* - y_a)/h,
\]

\[
t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a),
\]  

(4)

The R-FRCN adds a layer of arbitrary-angle bounding box regression layer in the ODN. According to the object location model of the arbitrary-angle bounding box defined above, this layer predicts five-coordinate values. In summary, the loss function of the ODN is defined as shown in Equation (5). The third term in Equation (5) is the regression loss of the arbitrary-angle bounding box. It is the same as the definition of the regression loss of the horizontal bounding box, i.e. the second term, in the equation. We also use the Smooth_L1 for the regression loss.

\[
L_{ODN}(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N_{reg}} \sum_i [p_i^* \geq 1] L_{reg}(t_i, t_i^*) + \beta_1 \frac{1}{N_{reg}} \sum_i [p_i^* \geq 1] L_{reg}(t_i, t_i^*)
\]  

(5)

Here, \( \beta_1 \) and \( \beta_2 \) are the respective weights of the horizontal bounding box regression loss and arbitrary angle bounding box loss. The R2CNN has proved that the horizontal bounding box prediction and the arbitrary angle bounding box prediction are indispensable to the detection results. However, it neglects the importance of each of the two to the final detection, and finally give the same weight to horizontal bounding box and arbitrary angle bounding box regression. In the experiment, it is found that when the prediction of the arbitrary-angle bounding box is added, the prediction of the original horizontal bounding box is affected.

As shown in Figure 6: (a) is the detection result of the original Faster R-CNN. This time, only the horizontal bounding box is predicted, and all ships are correctly detected; (b) is the detection result of the R2CNN with the arbitrary angle bounding box; (c) is the detection result of the R2CNN with the horizontal bounding box. We can see that when adding the prediction of the arbitrary-angle bounding box, the prediction and positioning of the horizontal bounding box is inaccurate, resulting in missed and false alarms. The reason is that during the network training process, it is necessary to pay attention to the regression loss of the horizontal bounding box and arbitrary angle bounding box at the same time, so it has a certain influence on the prediction of the horizontal bounding box. To solve this
Problem, we propose to redistribute the weights for the loss of horizontal and arbitrary-angle bounding boxes to obtain the best prediction results. The specific weight allocation will be discussed in the experimental part.

After defining the loss function, we carry out the network training for outputting the predicted values close to the true values by minimizing the loss function. The actual training process is shown in Algorithm 1.

**Algorithm 1. Details of the R-FRCN Training Procedure**

**Stage 1**
Use the large-scale classification dataset ImageNet (Deng et al., 2009) to pre-train the VGG16 (Simonyan & Zisserman, 2014) model. After the pre-training is completed, load the trained network parameters to initialize the improved VGG16 model. At the same time, the network parameters of the RPN and ODN are initialized randomly;

**Stage 2**
Keep the ODN network parameters unchanged, train the RPN according to the defined RPN loss function, i.e., Equation (1), generate a series of candidate regions, and fine-tune the FEN's parameters;

**Stage 3**
Keep the RPN's parameters unchanged, receive the candidate regions generated by the RPN, train the ODN network according to the defined ODN loss function, i.e., Equation (5), fine-tune the parameters of the FEN;

**Stage 4**
Keep the parameters of the FEN and ODN unchanged, and train the RPN for the second time. At this time, the RPN's parameters are fine-tuned, while still producing a series of candidate regions;

**Stage 5**
Keep the parameters of the FEN and RPN unchanged, and train the ODN for the second time. At this time, the ODN's parameters are fine-tuned;

Through the above stages, we can complete the training of the R-FRCNN. The above training steps are actually an alternating training method. The RPN and ODN are trained twice, respectively. The first alternating training requires fine-tuning the parameters of the original FEN. The second alternating training the RPN and ODN have completely shared the FEN.

**Experiment and analysis**

The experimental environment we use is Intel Core i7-7700 processor, Kingston 32GB RAM, NVidia GTX 1080Ti graphics card and ubuntu16.04 operating system. Next, we will conduct an experiment study around data sets, evaluation indicators, experimental results and analysis.

**Experiment dataset**

We conducted experiments on two datasets, namely, NWPU VHR-10 (Cheng et al., 2016) and TODRS-3, of which NWPU VHR-10 is a publicly available remote sensing dataset for object detection. We relabel the NWPU VHR-10 with the arbitrary angle bounding boxes. In order to further verify the validity of the
proposed approach, we constructed and labeled the dataset TODRS-3, which is a new remote sensing dataset for small object detection.

The NWPU VHR-10 contains a total of 800 remote sensing images, of which 715 images are derived from Google Earth, with a spatial resolution of 0.5 to 2 m. The rest 85 images are from the Vaihingen dataset (Rottensteiner et al., 2012) and have a spatial resolution of 0.08 m. The NWPU VHR-10 totally contains 10 types of objects, including aircraft, ships, oil depots, baseball fields, tennis courts, basketball courts, track and field fields, ports, bridges, vehicles. The dataset is divided into two sub-datasets: positive dataset and negative dataset. There are 650 images in the positive dataset, and at least one of the above object categories exists in each image. The negative dataset contains the remaining 150 images, and none of the above object categories exist in all images. Therefore, only the positive images are used in the actual detection, and some positive examples of the NWPU VHR-10 are shown in Figure 7. For each object in the image, the coordinate information of the upper left corner and the lower right corner of the bounding box and the corresponding object category are labeled.

The TODRS-3 is a dataset of remote sensing images with 4,615 small objects. The spatial resolution is 0.6 m to 3 m. And there are three categories of objects, including aircraft, ships, and oil depots. All images in the above two datasets are derived from the Google Earth. Each image has no fixed size but no more than 1000*1000 pixels. We manually mark the images with bounding box. Some examples are illustrated in Figure 8.

We divide the NWPU VHR-10 dataset and the TODRS-3 dataset into a training set and a test set at 4:1 ratio. For the NWPU VHR-10, 520 images are for training and 130 images are for testing. For the TODRS-3, 3692 images are for training and 923 images are for testing. The learning rate of each layer during the fine-tuning was set to 0.001, and it would be set to the previous 10% after 5000 iterations, and the momentum unit was set to 0.9. The weight decay is set to 0.0005, the batch size is set to 44, and the number of iterations depends on the convergence of the network model.

**Evaluation index**

Recall and precision are the basic performance evaluation indexes of object detection, normally defined as:

\[
\text{recall} = \frac{TP}{TP + FN} \quad (6)
\]

\[
\text{precision} = \frac{TP}{TP + FP} \quad (7)
\]
Here, TP is the number of correctly detected objects of this class; FP is the number of misjudged objects of this class in the detected results; FN is the number of undetected objects of this class; TN is the number of correctly detected objects of other classes.

Based on the Recall and Precision, we use four commonly used object detection indicators such as Missing Alarm Rate (MAR), False Alarm Rate (FAR), F1 index, and test time to evaluate the performance. The specific definitions of MAR, FAR, and F1 index are shown in Formula (8), (9), and (10), respectively. The test time refers to the average test time of a single image.

\[
MAR = 1 - \text{recall} \quad (8)
\]

\[
FAR = 1 - \text{precision} \quad (9)
\]

\[
F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (10)
\]

The MAR reflects the proportion of objects that are missed among all objects. The FAR reflects the proportion of objects that are misjudged among the detected objects. There is no absolute relationship between the FAR and MAR. F1 index is used for comprehensive consideration of FAR and MAR. The lower the FAR and the MAR, the better the detection effect, and the higher the F1 index, the better the detection effect.

**Experimental results and analysis**

**Anchor scale**

We give the detection results of R-FRCNN using different aspect ratios of anchors in Tables 3 and Tables 4, respectively. On the NWPU VHR-10, when adding four ratios (1:4, 4:1, 1:8, 8:1), the MAR is slightly reduced, the FAR is also slightly increased, and the F1 index is slightly reduced. This is because the increased aspect ratio of anchors in the R-FRCNN is

| Aspect Ratio of Anchor | MAR (%) | FAR (%) | F1 (%) | Test Time (ms) |
|-----------------------|---------|---------|--------|---------------|
| (1:1), (1:2), (2:1)   | 9.66    | 16.29   | 86.90  | 81.56         |
| (1:1), (1:2), (2:1), (1:4), (4:1), (1:8), (8:1) | 9.53 | 16.46 | 86.87 | 90.56 |

| Aspect Ratio of Anchor | MAR (%) | FAR (%) | F1 (%) | Test Time (ms) |
|-----------------------|---------|---------|--------|---------------|
| (1:1), (1:2), (2:1)   | 25.74   | 32.36   | 70.80  | 81.56         |
| (1:1), (1:2), (2:1), (1:4), (4:1), (1:8), (8:1) | 25.20 | 32.21 | 71.12 | 75.10 |

**Figure 8.** Samples of the TODRS-3.
mainly for ships with large aspect ratio, but for the NWPU VHR-10 data set, the number of ship objects with large aspect ratio is less, so the improvement is not obvious for ship, while for other objects, the increased anchor points may cause interference, so the F1 index is reduced. On the TODRS-3 data set, there are more ship objects with larger aspect ratio. After adding four ratios (1:4, 4:1, 1:8, 8:1) to the R-FRCN, the detection effect of ships is improved: the FAR and the MAR are both reduced; and the

| β-Value   | MAR (%) | FAR (%) | F1 (%) | Test Time (ms) |
|-----------|---------|---------|--------|----------------|
| β₁=1, β₂=1 | 9.66    | 16.28   | 86.90  | 81.56          |
| β₁=2, β₂=1 | 9.16    | 16.12   | 87.22  | 82.78          |
| β₁=1, β₂=2 | 9.91    | 16.61   | 86.61  | 81.25          |

Table 6. Detection results on the TODRS-3 with different β-values.

| β-Value   | MAR (%) | FAR (%) | F1 (%) | Test Time (ms) |
|-----------|---------|---------|--------|----------------|
| β₁=1, β₂=1 | 25.74   | 32.36   | 70.80  | 63.50          |
| β₁=2, β₂=1 | 24.50   | 31.81   | 71.66  | 67.49          |
| β₁=1, β₂=2 | 25.88   | 32.40   | 70.71  | 68.41          |

Table 7. Results on NWPU VHR-10 using different detection algorithms.

| Algorithm     | MAR (%) | FAR (%) | F1 (%) | Test Time (ms) |
|---------------|---------|---------|--------|----------------|
| Faster R-CNN  | 9.53    | 17.03   | 86.56  | 78.98          |
| R2CNN         | 9.66    | 16.38   | 86.85  | 83.73          |
| RIFD-CNN      | 9.50    | 17.42   | 86.36  | 79.28          |
| YOLOv4        | 9.48    | 16.76   | 86.73  | 63.29          |
| R-FRCNN       | 9.29    | 16.14   | 87.15  | 90.90          |

Table 8. Results on TODRS-3 using different detection algorithms.

| Algorithm     | MAR (%) | FAR (%) | F1 (%) | Test Time (ms) |
|---------------|---------|---------|--------|----------------|
| Faster R-CNN  | 27.85   | 33.61   | 69.15  | 57.67          |
| R2CNN         | 25.65   | 32.35   | 70.84  | 70.04          |
| RIFD-CNN      | 25.67   | 32.79   | 70.59  | 57.83          |
| YOLOv4        | 26.78   | 33.07   | 69.93  | 50.13          |
| R-FRCNN       | 24.22   | 31.73   | 71.83  | 77.46          |

Figure 9. Examples of detection results on the NWPU VHR-10.
final F1 index is increased. In addition, when the proportion of anchors is increased, i.e. the number of anchors is increased, the time consumption is increased accordingly.

**Weight distribution**
In the R-FRCN, we perform both horizontal bounding box regression and arbitrary-angle bounding box regression. In order to explore the influence of the two on the final detection result, we set up a set of comparative experiments. The experimental results are shown in Tables 5 and Tables 6.

It can be seen from the table that when $\beta_1=2$ and $\beta_2=1$, that is, when the regression of the horizontal bounding box is given more weight, the lowest MAR, FAR, and the highest F1 index are obtained. Note that when $\beta_1=1$ and $\beta_2=2$, the MAR and FAR are slightly higher than those when $\beta_1=2$ and $\beta_2=1$. This shows that paying too much attention to the arbitrary-angle bounding box will result in the inaccurate positioning of the horizontal bounding box, and the prediction of the arbitrary-angle bounding box depends on the prediction result of the horizontal bounding box, so the final F1 index is reduced.

**Algorithm comparison**
We now compare the performance of our method with the four state-of-the-art approaches Faster R-CNN (Ren et al., 2017), RIFD-CNN (Chen et al., 2019), YOLOv4 (Bochkovskiy et al., 2020), and R2CNN (Jiang et al., 2017). We present the detection results of different object detection algorithms on the two datasets in Tables 7 and Tables 8.

Generally, we can see from the table that compared to other comparison algorithms, our R-FRCNN algorithm has the lowest MAR and FAR on both data sets, and its F1 index is the highest. We can also find that the detection results of the R2CNN and the R-FRCNN are better than those of the original Faster R-CNN, which shows the introduction of the arbitrary-angle bounding boxes can effectively reduce the MAR and FAR, and improve the performance of object recognition of remote sensing image. In terms of the test time, YOLOv4 (Bochkovskiy et al., 2020) performs the best on both datasets.

Compared with R2CNN, due to the addition of the corresponding anchor point ratio setting for the ship with a relatively large aspect ratio, and the weight of the horizontal bounding box regression is increased, the proposed R-FRCNN achieves a better detection result. Therefore, our proposal has better applicability in object detection of remote sensing image. In addition, since the number of anchor points is increased and the prediction of bounding boxes at any angle is increased, the test time of the R-FRCNN is a little longer than that of Faster R-CNN and R2CNN, but it does not
significantly increase the time consumption and is within an acceptable range.

**Detection display**

Figures 9 and Figures 10 respectively show some examples of the detection results of the R-FRCNN on the two data sets. From the figure, it can be seen that whether the object is densely distributed or non-densely distributed, this algorithm can obtain more satisfactory detection results.

**Conclusion**

In this paper, we focus on object location and detection of remote sensing images. For horizontal boundary box location in most current deep learning based object detection algorithms, the inaccurate location of densely distributed and arbitrary directions of remote sensing images leads to missing detections. In order to solve this problem, we propose an object location method based on the arbitrary-angle bounding boxes and embed it into the Faster R-CNN, forming the new detection framework, i.e. R-FRCNN. In the R-FRCNN, we specially add the anchor ratio for ships with a relatively large aspect ratio and increase the weights of the horizontal bounding box regression to reduce the interference of the arbitrary-angle bounding box on the horizontal bounding box prediction. Finally, we conduct experiments on two object detection datasets of remote sensing image, NWPU VHR-10 and TODRS-3. We study the impact of different proportions of anchors on the detection performance, analysis the degree of impact of horizontal and arbitrary angle bounding box regression on the detection results. When compared with other deep learning based methods, the R-FRCNN can effectively reduce the MAR and FAR of object detection of remote sensing images.

Our R-FRCNN cannot completely mitigate the influence of the arbitrary-angle bounding box on the prediction of the horizontal bounding box. Therefore, the R-FRCNN will continue to be improved in the future, so as to more accurately locate the object in any direction.

**Acknowledgments**

This work was supported by the Youth Project of the Provincial Natural Science Foundation of Anhui Province 1908085QF285, the Scientific Research and Development Fund of Hefei University 19ZR05ZDA, the Talent Research Project Fund of Hefei University 18-19RC26, the Key Research Project of Anhui Province 201904d07020002, the University Natural Sciences Research Project of Anhui Province KJ2020A0661 and KJ2020A0665.

**Disclosure statement**

No potential conflict of interest was reported by the authors.

**References**

Bachagha, N., Wang, X., Luo, L., Li, L., Khatteli, H., & Lasaponara, R. (2020). Remote sensing and GIS techniques for reconstructing the military fort system on the Roman boundary (Tunisian section) and identifying archaeological sites. Remote Sensing of Environment, 236, 111418. https://doi.org/10.1016/j.rse.2020.111418 236 doi:10.1016/j.rse.2019.111418

Bao, H., Ming, D., Guo, Y., Zhang, K., Zhou, K., & Du, S. (2020). DFCNN-based semantic recognition of urban functional zones by integrating remote sensing data and POI data. Remote Sensing, 12(7), 1088. https://doi.org/10.3390/rs12071088

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. https://arxiv.org/abs/2004.10934

Cao, Y., Niu, X., & Dou, Y. (2016, August). Region-based convolutional neural networks for object detection in very high resolution remote sensing images. 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 548–554). IEEE. (pp. 548–554). IEEE, Changsha, China. https://doi.org/10.1109/FSKD.2016.7603232

Chen, C., Zhong, J., & Tan, Y. (2019). Multi-oriented and small object detection with convolutional neural networks for aerial image. Remote Sensing, 11(18), 2176. https://doi.org/10.3390/rs11182176

Chen, J., Zhang, B., & Wang, C. (2015). Backscattering feature analysis and recognition of civilian aircraft in TerraSAR-X images. IEEE Geoscience and Remote Sensing Letters,12(4), 796–800. https://doi.org/10.1109/LGRS.2014.2362845

Cheng, G., Han, J., Zhou, P., & Xu, D. (2018). Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. IEEE Transactions on Image Processing, 28(1), 265–278. https://doi.org/10.1109/TIP.2018.2867198

Cheng, G., Zhou, P., & Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Transactions on Geoscience and Remote Sensing, 54 (12), 7405–7415. https://doi.org/10.1109/TGRS.2016.2601622

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004, May). Visual categorization with bags of keypoints. Workshop on statistical learning in computer vision, ECCV Prague, Czech Republic. (Vol. I, No. 1–22, pp. 1–2). https://www.cs.cmu.edu/~cefras/courses/LBMV07/Papers/csurka-ecvc-04.pdf

Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05) (Vol. 1, pp. 886–893). IEEE. San Diego, CA, USA. https://doi.org/10.1109/CVPR.2005.177

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255). IEEE, Miami, FL, USA. https://doi.org/10.1109/CVPR.2009.5206848

Deng, Z., Lei, L., Sun, H., Zou, H., Zhou, S., & Zhao, J. (2017). An enhanced deep convolutional neural network for densely packed objects detection in remote sensing
images. In 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP) (pp. 1–4). IEEE. https://doi.org/10.1109/RSIP.2017.7958800

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1627–1645. https://doi.org/10.1109/TPAMI.2009.167

Freund, Y. (1995). Boosting a weak learning algorithm by majority. Information and Computation, 121(2), 256–285. https://doi.org/10.1006/inco.1995.1136

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In In icml (Vol. 96, pp. 148–156), Bari, Italy, Morgan Kaufmann. https://dl.acm.org/doi/10.5555/3091696.3091715

Girshick, R. (2015). Fast r-cnn. Proceedings of the IEEE international conference on computer vision (pp. 1440–1448), Santiago, Chile. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(1), 142–158. https://doi.org/10.1109/TPAMI.2015.2437384

Harel, J., Koch, C., & Perona, P. (2007). Graph-based visual saliency. In Advances in neural information processing systems (pp. 545–552), Vancouver, British Columbia, Canada, MIT Press. https://papers.nips.cc/paper/3095-graph-based-visual-saliency.pdf

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. Proceedings of the IEEE international conference on computer vision (pp. 2961–2969), Venice, Italy. https://doi.org/10.1109/ICCV.2017.322

Hou, X., & Zhang, L. (2007, June). Saliency detection: A spectral residual approach. 2007 IEEE Conference on computer vision and pattern recognition (pp. 1–8), Minneapolis, MN, USA, IEEE. https://doi.org/10.1109/CVPR.2007.383267

Hu, Y., Li, X., Zhou, N., Yang, L., Peng, L., & Xiao, S. (2019). A sample update-based convolutional neural network framework for object detection in large-area remote sensing images. IEEE Geoscience and Remote Sensing Letters, 16(6), 947–951. https://doi.org/10.1109/LGRS.2018.2889247

Hua, X., Wang, X., Rui, T., Zhang, H., & Wang, D. (2020). A fast self-attention cascaded network for object detection in large scene remote sensing images. Applied Soft Computing, 94, 106495. https://doi.org/10.1016/j.asoc.2020.106495

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11), 1254–1259. https://doi.org/10.1109/34.730558

Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., & Luo, Z. (2017). R2cnn: Rotational random cnn for orientation robust scene text detection. arXiv preprint arXiv:1706.09579. https://arxiv.org/ftp/arxiv/papers/1706/1706.09579.pdf

Kedzierslawski, R., Le Caillec, J. M., & Czarnecki, W. (2019). Simulation of subsurface imaging for remote sensing and buried object detection from airborne platforms. European Journal of Remote Sensing, 52(1), 583–598. https://doi.org/10.1080/22797254.2019.1699546

Khan, I., & Sinha, R. (2019). Discovering 'buried' channels of the Palaue-Yamuna river in NW India using geophysical evidence: Implications for major drainage reorganization and linkage to the Harappan Civilization. Journal of Applied Geophysics, 167, 128-139. https://doi.org/10.1016/j.jappgeo.2019.05.017

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001). Morgan Kaufmann, Williams College, Williamstown, MA, USA. https://dl.acm.org/doi/10.5555/645530.655813

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Li, K., Cheng, G., Bu, S., & You, X. (2017). Rotation-in sensitive and context-augmented object detection in remote sensing images. IEEE Transactions on Geoscience and Remote Sensing, 56(4), 2337–2348. https://doi.org/10.1109/TGRS.2017.2778300

Li, K., Wan, G., Cheng, G., Meng, L., & Han, J. (2020). Object detection in optical remote sensing images: A survey and a new benchmark. ISPRS Journal of Photogrammetry and Remote Sensing, 159(2020), 296–307. https://doi.org/10.1016/j.isprsjprs.2019.11.023

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117–2125), Honolulu, HI, USA. https://doi.org/10.1109/CVPR.2017.106

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. European conference on computer vision (pp. 21–37). Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of Computer Vision, 60(2), 91–110. https://doi.org/10.1023/B:VISL.0000029664.99615.94

Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., & Xue, X. (2018). Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia, 20(11), 3111–3122. https://doi.org/10.1109/TMM.2018.2818020

Melillos, G., Themistocleous, K., Papadavid, G., & Hadjimitsis, D. G. (2018). Detection of Military Underground Structures through the Remote Sensing Investigation of Phenological Cycle of Crops. Advances in Remote Sensing, 7(03), 235. https://doi.org/10.4236/ars.2018.73016 03 07 https://doi.org/10.4236/ars.2018.73016

Oreno, H. A., & Petrie, C. A. (2017). Large-scale, multi-temporal remote sensing of palaeo-river networks: A case study from Northwest India and its implications for the Indus Civilisation. Remote Sensing, 9(7), 735. https://doi.org/10.3390/rs9070735

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779–788), Las Vegas, NV, USA. https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., & Breitkopf, U. (2012). The ISPRS benchmark on urban object classification and 3D building reconstruction. ISPRS Annals of the Photogrammetry,
Remote Sensing and Spatial Information Sciences 1-3, Nr. 1, 1(1), 293–298. https://doi.org/10.15488/5042
Shi, B., Bai, X., & Belongie, S. (2017). Detecting oriented text in natural images by linking segments. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2550–2558), Honolulu, HI, USA. https://doi.org/10.1109/CVPR.2017.371
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://arxiv.org/pdf/1409.1556.pdf
Sun, F., & Wu, Z. Z. (2020). Inclined and dense object detection for remote sensing images, Zenodo.org. https://doi.org/10.5281/zenodo.3999187
Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. International journal of computer vision, 104(2), 154–171. https://doi.org/10.1007/s11263-013-0620-5
Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media. https://doi.org/10.1007/978-1-4757-2440-0
Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2008). Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2), 210–227. https://doi.org/10.1109/TPAMI.2008.79
Wu, Z. Z., Wan, S. H., Wang, X. F., Tan, M., Zou, L., Li, X. L., & Chen, Y. (2020). A benchmark data set for aircraft type recognition from remote sensing images. Applied Soft Computing, 89(2020), 106132. https://doi.org/10.1016/j.asoc.2020.106132
Yao, Y., Jiang, Z., Zhang, H., Zhao, D., & Cai, B. (2017). Ship detection in optical remote sensing images based on deep convolutional neural networks. Journal of Applied Remote Sensing, 11(4), 042611. https://doi.org/10.1117/1.JRS.11.042611
Ye, Z., Dong, R., Bai, L., & Nian, Y. (2020). Adaptive collaborative graph for discriminant analysis of hyperspectral imagery. European Journal of Remote Sensing, 53(1), 91–103. https://doi.org/10.1080/22797254.2020.1735947
Zhang, W., Sun, X., Fu, K., Wang, C., & Wang, H. (2013). Object detection in high-resolution remote sensing images using rotation invariant parts based model. IEEE Geoscience and Remote Sensing Letters, 11(1), 74–78. https://doi.org/10.1109/LGRS.2013.2246538
Zhong, Y., Ma, A., soon Ong, Y., Zhu, Z., & Zhang, L. (2018). Computational intelligence in optical remote sensing image processing. Applied Soft Computing, 64, 75-93. https://doi.org/10.1016/
Zhu, L., Chen, Y., Yuille, A., & Freeman, W. (2010, June). Latent hierarchical structural learning for object detection. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 1062–1069), San Francisco, CA, USA. IEEE. https://doi.org/10.1109/CVPR.2010.5540096