



## Article

# A Method of Fusing Probability-Form Knowledge into Object Detection in Remote Sensing Images

Kunlong Zheng <sup>1,2</sup> , Yifan Dong <sup>1,2,\*</sup> , Wei Xu <sup>1,2</sup> , Yun Su <sup>1,2</sup> and Pingping Huang <sup>1,2</sup>

<sup>1</sup> College of Information Engineering, Inner Mongolia University of Technology, Hohhot 010051, China

<sup>2</sup> Inner Mongolia Key Laboratory of Radar Technology and Application, Hohhot 010051, China

\* Correspondence: yfdong@imut.edu.cn

**Abstract:** In recent years, dramatic progress in object detection in remote sensing images has been made due to the rapid development of convolutional neural networks (CNNs). However, most existing methods solely pay attention to training a suitable network model to extract more powerful features in order to solve the problem of false detections and missed detections caused by background complexity, various scales, and the appearance of the object. To open up new paths, we consider embedding knowledge into geospatial object detection. As a result, we put forward a method of digitizing knowledge and embedding knowledge into detection. Specifically, we first analyze the training set and then transform the probability into a knowledge factor according to an analysis using an improved version of the method used in existing work. With a knowledge matrix consisting of knowledge factors, the Knowledge Inference Module (KIM) optimizes the classification in which the residual structure is introduced to avoid performance degradation. Extensive experiments are conducted on two public remote sensing image data sets, namely DOTA and DIOR. The experimental results prove that the proposed method is able to reduce some false detections and missed detections and obtains a higher mean average precision (mAP) performance than the baseline method.

**Keywords:** convolutional neural networks (CNNs); remote sensing images; object detection; knowledge inference module



**Citation:** Zheng, K.; Dong, Y.; Xu, W.; Su, Y.; Huang, P. A Method of Fusing Probability-Form Knowledge into Object Detection in Remote Sensing Images. *Remote Sens.* **2022**, *14*, 6103. <https://doi.org/10.3390/rs14236103>

Academic Editor: Silvia Liberata Ullo

Received: 2 October 2022

Accepted: 17 November 2022

Published: 1 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

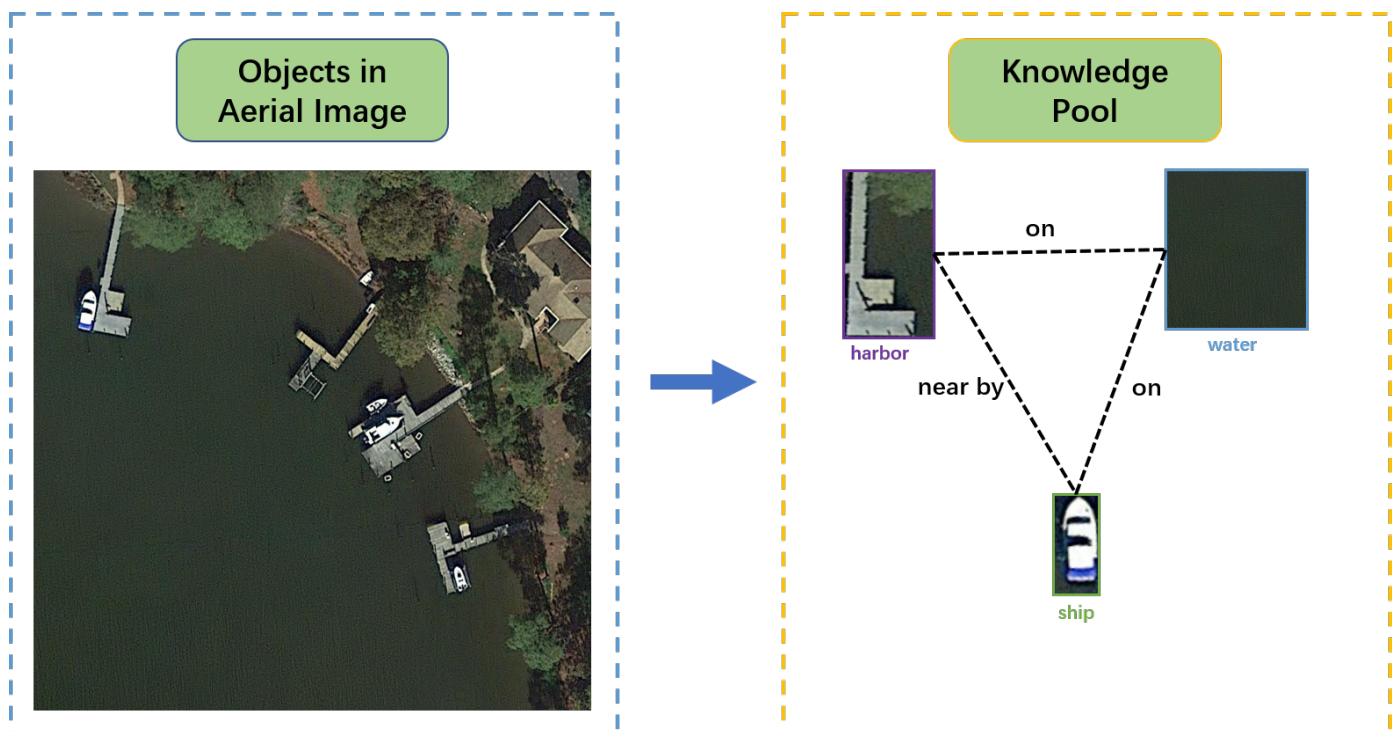
## 1. Introduction

Object detection classifies and locates geospatial objects in aerial images and is a fundamental task in remote sensing. Recently, there have been great advances in object detection in aerial images (ODAI) [1,2] with the advent of deep convolutional neural networks. Because of this, applications in some fields, such as (UAVs) and wireless sensor networks (WSNs), have also benefited dramatically [3,4]. Although a large number of excellent methods [5–9] have been invented in the past few years, updating the previous state-of-the-art methods, there are still problems, such as missing detections and false detections caused by densely distributed objects, varied object sizes, and occluded objects. To illustrate, Figure 1a is a visualization of false harbor and helicopter detection, where a plane is detected as the harbor, and a piece of land is regarded as a helicopter. In Figure 1b, a piece of agricultural land is detected as a soccer ball field. In Figure 1c, a shaded basketball court is not detected. In the right of Figure 1d, two bridges are not detected, which damages the detection performance. The traditional object detection paradigm only takes the extracted feature into consideration. However, this is the bird eye view in aerial images. Therefore, detection algorithm inference only uses the roof information from aerial images. Moreover, due to its complex background features, such as the shape similarity and appearance similarity between objects and the background, the wrong labels are predicted. Because of the ambiguous appearance, those algorithms with an advantage in terms of extracting features do not have a significant effect. This is because when the instances are ambiguous, the extracted features are not that powerful, which leads to missed detection and false detection.



**Figure 1.** False detection in red circle and missed detection in yellow circle: (a) false detection of a helicopter and harbor; (b) false detection of a soccer ball field; (c) missed detection of a basketball court; (d) missed detection of a bridge.

However, the human visual recognition system informs us how to tackle this problem, because humans take their surroundings, as well as objects into account when conducting recognition tasks [10], which might be beneficial for geospatial object detection. When humans see one thing, it is common for them to think of another thing related to the seen one. Normally, for instance, when a harbor comes into view, there might be ship parking along the harbor. In addition, the ship and the harbor appear with water, which provides the link between the water and the object. Relationships between objects and relationships between a scene and objects can be assembled into a knowledge pool, which is then used in the object detection task, the process of which is shown in Figure 2. With the utilization of such a knowledge pool, the human visual recognition system is superior to the artificial intelligent algorithm in terms of its detection performance. The visual recognition method mentioned above benefits from the utilization of knowledge, which is worth adopting in the object detection algorithm.



**Figure 2.** Relationships between the ship, harbor, and water assembled in the knowledge pool.

Obviously, it is by converting pictures into data that object detection algorithms process converted data and can detect objects. Thus, in order to improve the detection method added to work, such as the human visual system, the transformation of knowledge into data is essential. Refs. [11,12] were our sources of inspiration. Li et al. [11] presented

statistics regarding the probability of objects appearing in each scene and assembled these data into a probability matrix with the aim of improving scene classification. Consequently, thinking in opposition, it is profitable to use the probability of an object appearing in the relevant scene to guide detection. In [12], Xu et al. utilized a class relation matrix to boost the object detection performance. Specifically, they integrated a class relation matrix with a high-dimensional feature to obtain an enhanced feature, which is not that intuitively explainable due to the high-dimensional space. As a result, to deal with false detections, such as a harbor being detected on land, in this work, we explored the co-relations between the water area and objects by analyzing a data set. Moreover, in order to utilize the implicit relation as knowledge, we analyzed the conditional co-occurrence probabilities of different categories, which is the expression of the implicit relation.

In this paper, an approach to utilizing knowledge is proposed. Our overall contributions are as follows:

- We extract two kinds of knowledge in the form of probabilities, namely the correlations between classes and the correlations between the water area and classes, by analyzing the DOTA [13] and DIOR [2] training sets. Then, we transform the extracted knowledge into a knowledge factor using a novel equation improved from [14].
- We propose a method, namely the Knowledge Inference Module (KIM), of integrating knowledge into object detection in remote sensing images. Through an evaluation of two public aerial data sets, our method obtains a higher mAP than the baseline model, Oriented R-CNN [15] with fewer false and missed detections.

## 2. Related Work

### 2.1. Object Detection in Remote Sensing Images

It is obvious that remote sensing images are very different from natural scene images due to their large aspect ratio, arbitrary orientation, variation in appearance and scale, densely distributed objects, etc., leading to difficulties in directly transferring object detection in natural scene images to object detection in aerial images. Therefore, references [9,16,17] proposed different ways to improve the detection performance. Han et al. [9] used the Feature Alignment Module (FAM) to generate anchors, encode orientation information, and obtain orientation-sensitive features using the Oriented Detection Module (ODM). Yang et al. [17]  $R^3Det$  refined features by re-encoding the positional information of the bounding box to the corresponding feature points through pixel-wise feature interpolation. Ming et al. [16] constructed critical features through the Polarization Refinement Module (PAM) and used the Rotation Anchor Refinement Module (R-ARM) to finally obtain a powerful semantic representation. Yang et al. [7] used a circular smooth label (CSL) to solve the problem of having discontinuous boundaries due to angular periodicity or corner ordering. Ding et al. [6] proposed the RoI-Transformer, which contains Rotated RoI Warping (RRoI Warping), which extracts rotation-invariant features, and the Rotated RoI Learner (RRoI Learner), which acquires objects' orientation information, to address detection misalignment. The aforementioned [6] achieved great success in boosting the detection performance. However, in this method, the amount of computation increases, the detection speed decreases, and the GPU memory is burdened. Consequently, some methods [15,18] with reduced computation have been proposed with the condition of maintaining accuracy. Xie et al. [15] built the Oriented-RCNN including the Oriented RPN to generate high-quality anchors and the Oriented R-CNN head, which has a faster detection speed than the RoI-Transformer [6]. Han et al. [18] proposed a novel backbone, the Rotation-equivariant ResNet (ReResNet), which has a reduction in parameters of over 60% compared to ResNet [19].

### 2.2. Utilization of Existing Knowledge

Works that integrate knowledge into object detection can be divided into two kinds. One involves learning relationships between objects and scenes or relationships between

categories during the training process; another makes use of existing knowledge to improve the detection performance.

### 2.2.1. Utilization of Knowledge Learned in Training

Chen et al. [20] directly added a global image feature to an ROI feature, which is a simple and effective method. Liu et al. [21] designed a Structure Inference Network (SIN) with two parallel branches, one for global information extraction and another for local information extraction. Though this method is more sophisticated and useful, the GRU cell in each branch requires more GPU memory, making the process time-consuming. Siris et al. [22] applied an Attention [23] mechanism to combine global information and local information and produced a great performance in terms of salient object detection. Li et al. [24] obtained local features and contextual features from the Region of Interest (RoI), then integrated them into a Local-Contextual joint feature for geospatial object detection. Zhang et al. [25] proposed CAD-net, which can learn correlations between objects and scenes from global features and object features. Refs. [26–28] designed a module to enhance scene or global information in order to capture contextual information.

### 2.2.2. Utilization of Existing Knowledge

In [11,29], the prior scene-class graph was adopted to infer the relationship between a scene and an object through the Bayesian criterion. The adjacency matrix learned from the visual feature was adopted in the relation-reasoning module in [30,31]. Shu et al. [32] introduced the Graph Attention Network (GAT) and Graph Convolutional Network (GCN) to learn hidden knowledge from the obtained co-occurrence matrix and scene–object matrix. Fang [14] proposed a probability-based knowledge graph and graph-based knowledge graph with a cost function containing a knowledge graph to carry out knowledge-aware detection. In [33], an explicit knowledge module and an implicit knowledge module, containing an explicit knowledge graph and an implicit knowledge graph, respectively, were introduced, to enhance the RoI features.

## 3. Establishment of the Knowledge Matrix

### 3.1. Knowledge Matrix Establishment

In this section, the procedures used to create knowledge matrices, the category conditional co-occurrence knowledge matrix and the water area knowledge matrix from the data set are illustrated.

#### 3.1.1. Conditional Co-Occurrence Knowledge Matrix

We first count the probability  $n(l)$  that every category appears in images using Equation (1)

$$n(l) = \frac{N_{img}(l)}{N_{allimg}}. \quad (1)$$

where  $N_{img}(l)$  denotes the number of images in which category  $l$  occur, and  $N_{allimg}$  denotes the number of images in the trainset. An analysis of the DOTA trainset is shown in Table 1, and an analysis of DIOR is shown in Appendix A.

**Table 1.** The number of images in which the category occurs  $N_{img}(l)$  and the probabilities of the class occurrence  $n(l)$ .

| Object Categories  | $N_{img}(l)$ | $n(l)$ |
|--------------------|--------------|--------|
| Plane              | 197          | 0.1396 |
| Baseball diamond   | 122          | 0.0864 |
| Bridge             | 210          | 0.1488 |
| Ground track field | 177          | 0.1254 |
| Small vehicle      | 486          | 0.3444 |
| Large vehicle      | 380          | 0.2693 |
| Ship               | 326          | 0.2310 |
| Tennis court       | 302          | 0.2140 |
| Basketball court   | 111          | 0.0786 |
| Storage tank       | 161          | 0.1141 |
| Soccer ball field  | 136          | 0.0963 |
| Roundabout         | 170          | 0.1204 |
| Harbor             | 339          | 0.2402 |
| Swimming pool      | 144          | 0.1020 |
| Helicopter         | 30           | 0.0212 |

Then, we determine the probability of conditional co-occurrence. In detail, we first count  $N_{img}(l|l')$ , the number of images in which class  $l$  and class  $l'$  appear together. Then, we use  $N_{img}(l)$  dividing  $N_{img}(l|l')$ .

Fang et al. [14] proposed Equation (2)

$$S_{l,l'} = \max(\log \frac{n(l|l')N_{allimg}}{n(l)n(l')}, 0) \quad (2)$$

where  $n(l)$  denotes the probability of occurrence for category  $l$ , and  $N_{allimg}$  denotes the number of all images to transform the frequencies into a knowledge matrix. We applied this formula to the DOTA and DIOR data sets.

However, this knowledge matrix has some disadvantages. On one hand, the numerical dimension, which is over 10, is too large to be suitable for optimizing predicted class scores. The oversized numerical dimension has an excessive influence on the predicted class scores, causing the knowledge matrix to become the decisive element, whereas our original intention was to make use of the knowledge matrix to improve predictions. On the other hand, when two categories do not co-occur, the corresponding position in the knowledge matrix will be set as 0, which is a rigid way to deal with the aforementioned situation. Thus, we propose a novel processing approach that introduces a zero factor to replace 0. Additionally, simply setting zero could exert a negative impact on the generalization performance. This is because the none-conditional-co-occurrence of two categories in the data set only denotes that the two categories have little correlation, but it does not mean that the two categories never co-occur in the real world.

In order to address aforementioned problems, we modify Equation (2) and propose Equation (3).

$$S_{l,l'} = \begin{cases} \log \frac{\frac{n(l,l')}{n(l)n(l')} (n(l) + n(l'))}{N_{img}(l) + N_{img}(l')} & n(l,l') \neq 0 \\ \log \frac{\frac{\epsilon}{n(l)n(l')} (n(l) + n(l'))}{N_{img}(l) + N_{img}(l')} & n(l,l') = 0 \end{cases} \quad (3)$$

The modifications are as follows:

1. We abandon the  $\max()$  function, which means that the range of  $S_{l,l'}$  extends to the negative axis, which is suitable for situations where category  $l$  and category  $l'$  do not co-occur or barely co-occur;

2. We regard the log part  $\log \frac{n(l,l')}{n(l)n(l')}$  as a conditional co-occurrence factor and abandon  $N_{allimg}$  and add  $\frac{(n(l)+n(l'))}{N_{img}(l)+N_{img}(l')}$  as a scale factor into the equation in order to the scale knowledge factor into a proper numerical dimension and the make knowledge factor adaptive to the probability of category occurrence;
3. We split our novel equation into two branches, where the upper one is for situations where  $l$  and  $l'$  appear together in the training set and the lower one computes the knowledge factor when  $l$  and  $l'$  do not co-occur;
4. In the lower branch, we replace  $n(l, l')$  with the zero factor  $\epsilon$  in Equation (4), where  $N_{object}$  denotes the number of instances belonging to category  $l$ , making the equation more elegant when  $n(l, l')$  equals 0.

$$\epsilon = \frac{1}{N_{object}(l)N_{object}(l')} \quad (4)$$

### 3.1.2. Water Area Knowledge Matrix

In this section, we first count the number of images containing water areas and the number not containing water areas and compute the occurrence probability of water appearing and not appearing as 0.5102 and 0.4898 in DOTA, respectively. Then, we determine the probability of a class appearing with water area or not using Equations (5) and (6)

$$n(l|w) = \frac{N_{img}(l|w)}{N_{img}(l)} \quad (5)$$

$$n(l|\bar{w}) = \frac{N_{img}(l|\bar{w})}{N_{img}(l)} \quad (6)$$

where  $n(l|\bar{w})$  and  $N_{img}(l|\bar{w})$  denote the probability of class  $l$  in a water area and the number of images in which class  $l$  does not occur in a water area. Those with a water area are denoted by  $n(l|w)$  and  $N_{img}(l|w)$ . Probabilities of classes of DOTA appearing in a water area and not in a water area are demonstrated in Table 2. The probabilities of the classes of DIOR appearing in a water area and not in a water area are demonstrated in Table A2.

**Table 2.** Probabilities of classes appearing with water area and not with water area. Column  $n(l|w)$  denotes the probability of category  $l$  appearing with water area;  $n(l|\bar{w})$  is the probability of category  $l$  not appearing with water area.

| Object Categories  | $n(l w)$ | $n(l \bar{w})$ |
|--------------------|----------|----------------|
| plane              | 0.1748   | 0.8252         |
| baseball-diamond   | 0.5543   | 0.4457         |
| bridge             | 0.9755   | 0.0245         |
| ground-track-field | 0.6462   | 0.3538         |
| small-vehicle      | 0.2837   | 0.7163         |
| large-vehicle      | 0.1584   | 0.8416         |
| ship               | 0.9994   | 0.0006         |
| tennis-court       | 0.2945   | 0.7055         |
| basketball-court   | 0.2544   | 0.7456         |
| storage-tank       | 0.9402   | 0.0598         |
| soccer-ball-field  | 0.5215   | 0.4785         |
| roundabout         | 0.6166   | 0.3834         |
| harbor             | 1        | 0              |
| swimming-pool      | 1        | 0              |
| helicopter         | 0.0015   | 0.9985         |

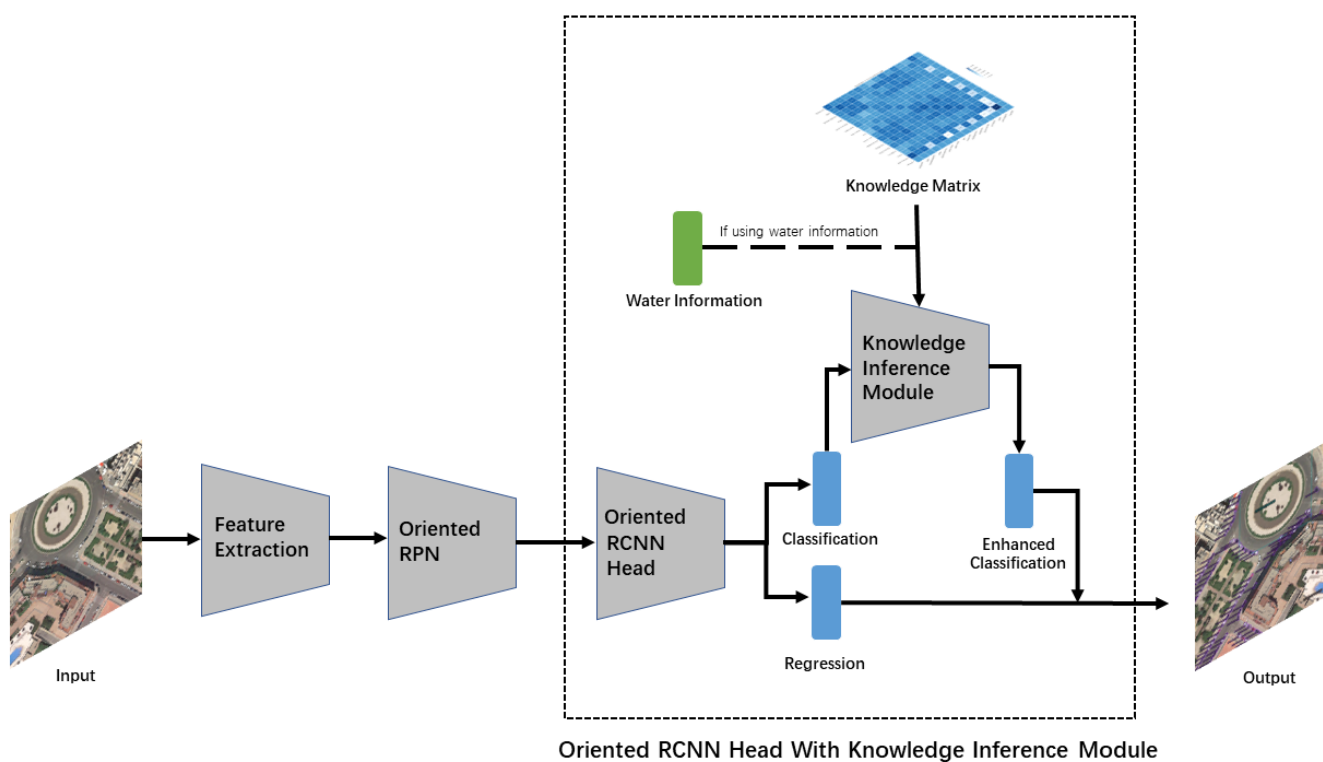
Similar to the conditional co-occurrence knowledge matrix, we also partly modify Equation (2) and propose Equation (7)

$$S_{l,*} = \begin{cases} \frac{\log \frac{n(l|*)}{n(l)n(*)} n(l)}{N_{img}(l)} & n(l|*) \neq 0 \\ \frac{\log \frac{\epsilon}{n(l)n(*)} n(l)}{N_{img}(l)} & n(l|*) = 0 \end{cases} \quad (7)$$

where  $\epsilon$ , taken as the zero factor, equals  $\frac{1}{N_{object}(l)}$ , and  $n(*)$  is the probability of a water area appearing, where  $*$  is  $w$  or  $\bar{w}$  meaning that there is a water area and that there is no water area, respectively. Equation (7) was also applied to DOTA and DIOR.

#### 4. Methods

In this paper, we propose a knowledge-inferencing module that uses a residual structure to optimize the predicted class scores, which helps the detector to perform better. Our method can be applied to any two-stage object detection framework. In this work, we chose the Oriented R-CNN [15] as the framework and baseline model. The overall structure of the framework filled with the knowledge-aware bounding box head is presented in Figure 3.

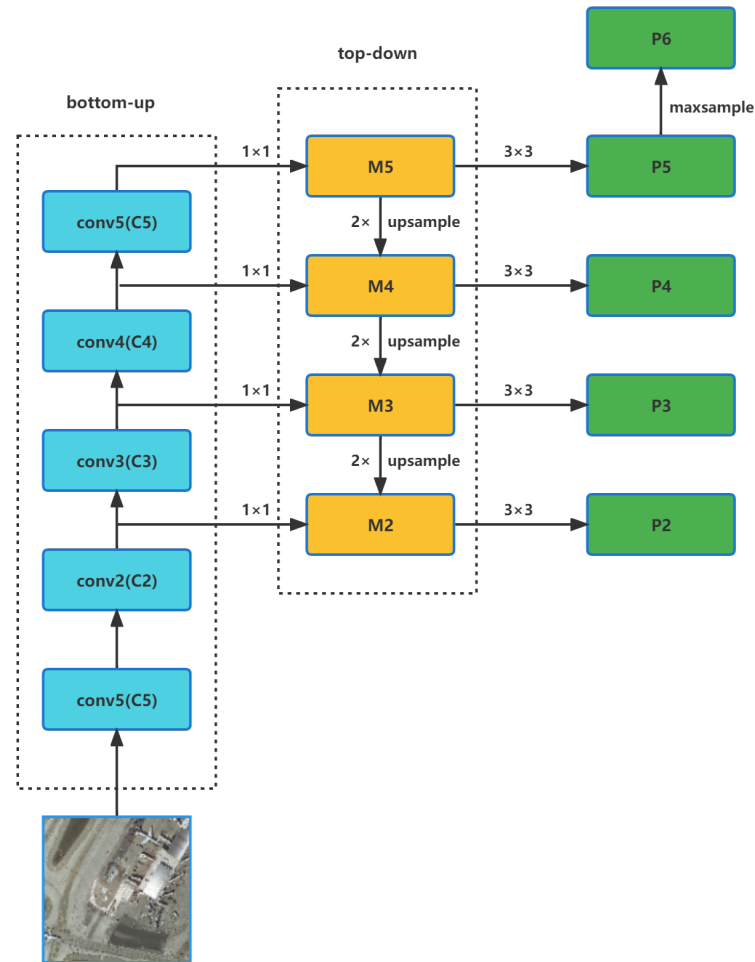


**Figure 3.** The overall structure of the framework filled with the Knowledge Inference module, a two-stage detector. Feature extraction module first extracts multi-scale features fed into the Oriented RPN. Region proposals are generated by the Oriented RPN and used for classifying and regression in the Oriented RCNN head. Finally, the classification scores are fed into the Knowledge Inference Module, resulting in knowledge-enhanced classification scores.

##### 4.1. Feature Extraction

Resnet-50 [19] and FPN [34] are adopted into the Feature Extraction module to extract multi-scale features with which size-various objects can more easily be detected. Figure 4 shows the overall structure of the Feature Extraction module. In the bottom-up part, the

input image is first input into a series of convolutional layers to generate low-semantic feature maps and high-semantic feature maps  $\{C_1, C_2, C_3, C_4, C_5\}$ . Moreover, the top-down part obtains more powerful features by fusing features. To be specific,  $M_4$  is equal to the sum of the result of the upsampling of  $M_5$  and the result of the  $1 \times 1$  convolution of  $C_4$ . In this way,  $M_3$  and  $M_2$  are generated.  $P_2, P_3, P_4$ , and  $P_5$  are obtained by feeding  $M_2, M_3, M_4$ , and  $M_5$  into a  $3 \times 3$  convolutional layer, and  $P_6$  is obtained by maxpooling  $P_5$ . Finally, this module outputs the fused multi-scale features  $\{P_2, P_3, P_4, P_5, \text{ and } P_6\}$ .



**Figure 4.** The overall structure of the Feature Extraction module.

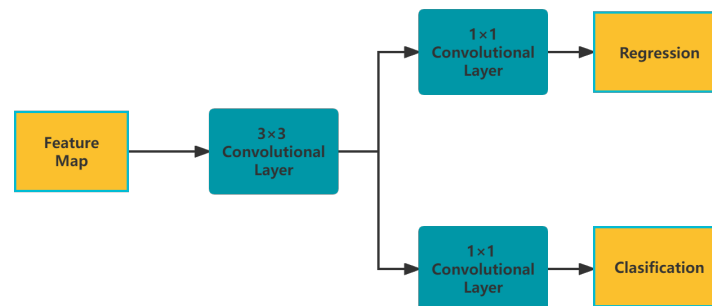
#### 4.2. Oriented RPN

The oriented RPN takes  $\{P_2, P_3, P_4, P_5, \text{ and } P_6\}$  as the input and output region proposals with location information and an objectness score. Figure 5 shows the structure of the Oriented RPN.

We set three horizontal anchors with aspect ratios of  $\{1:2, 1:1, \text{ and } 2:1\}$  for every location in the features of all scales. The anchors correspond to  $\{P_2, P_3, P_4, P_5, \text{ and } P_6\}$  and have pixel areas of  $\{32^2, 64^2, 128^2, 256^2, \text{ and } 512^2\}$ , which are represented by a 4-dimensional vector  $a = (a_x, a_y, a_w, a_h)$ .  $a_x$  and  $a_y$  denote the horizontal and vertical locations of the anchor center;  $a_w$  and  $a_h$  correspond to the width and height of the anchor. The upper branch of Figure 5, i.e., the regression branch, outputs the offset  $\delta = (\delta_x, \delta_y, \delta_w, \delta_h, \delta_\alpha, \delta_\beta)$  of proposals related to anchors. We decode the offset using Equation (8) to obtain oriented proposals  $(x, y, w, h, \Delta\alpha, \Delta\beta)$ , where  $(x, y)$  denotes the location of the proposed center coordinate,  $w$  and  $h$  correspond to the width and height of the external rectangle box of the proposal,  $\Delta\alpha$  and  $\Delta\beta$  denote the offsets of the proposal box vertex oriented to the midpoints of the top

and right sides of the corresponding external rectangle. The lower branch of Figure 5, i.e., the classification branch, outputs objectness scores.

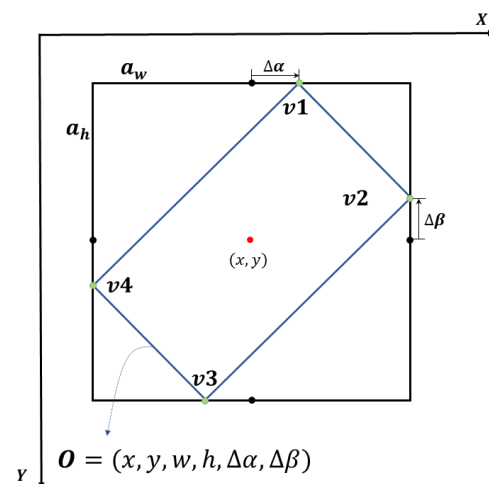
$$\begin{cases} \Delta\alpha = \delta_\alpha \cdot w, & \Delta\beta = \delta_\beta \cdot h \\ w = a_w \cdot e^{\delta_w}, & h = a_h \cdot e^{\delta_h} \\ x = \delta_x \cdot a_w + a_x, & y = \delta_y \cdot a_h + a_y \end{cases} \quad (8)$$



**Figure 5.** The structure of the Oriented RPN, which contains a  $3 \times 3$  convolutional layer and two sibling  $1 \times 1$  convolutional layers for classification and regression, respectively.

To represent the oriented object in elegant manner, the midpoint offset representation is introduced, the schematic of which is illustrated in Figure 6. In detail, the black horizontal box, i.e., the external rectangle of the blue one, is obtained from the anchor, where  $a_w$  and  $a_h$  are the width and height of the anchor, and the blue oriented box is the predicted oriented proposal box. The black dots and the light green dots are the midpoint of the external rectangle edges and the vertices of the oriented box, respectively. The predicted oriented proposal box can be represented as  $O = (x, y, w, h, \Delta\alpha, \Delta\beta)$ , which can be computed by Equation (8). Furthermore, the vertices of predicted oriented proposal are denoted by a set of coordinates  $v_1, v_2, v_3, v_4$ . Similarly,  $\Delta\beta$  is the distance between  $v_2$  and the midpoint  $(x, y - \frac{h}{2})$  of the top side, and because of the symmetry, the distance between  $v_3$  and the midpoint  $(x, y + \frac{h}{2})$  of the bottom side equals  $-\Delta\alpha$ . It is noticing that  $\Delta\alpha$  is distance between  $v_1$  and the midpoint  $(x, y - \frac{h}{2})$  of top side, and because of the symmetry distance between  $v_3$  and the midpoint  $(x, y + \frac{h}{2})$  of bottom side equals to  $-\Delta\alpha$ . Similarly,  $\Delta\beta$  is the distance between  $v_2$  and the midpoint  $(x + \frac{w}{2}, y)$  of the right side, and the distance between  $v_4$  and the midpoint  $(x, y + \frac{h}{2})$  of the left side equals  $-\Delta\beta$ . As a result, the vertices of the oriented proposal  $\{v_1, v_2, v_3, v_4\}$  can be computed using Equation (9).

$$\begin{cases} v_1 = (x, y - \frac{h}{2}) + (\Delta\alpha, 0) \\ v_2 = (x + \frac{w}{2}, y) + (0, \Delta\beta) \\ v_3 = (x, y + \frac{h}{2}) + (-\Delta\alpha, 0) \\ v_4 = (x - \frac{w}{2}, y) + (0, -\Delta\beta) \end{cases} \quad (9)$$



**Figure 6.** Schematic of the midpoint offset scheme.

In the training process, we assign positive and negative samples using the following rules:

1. An anchor that has an Intersection-over-Union(IoU) over 0.7 with any ground-truth box is regarded as a positive sample;
2. An anchor that has an IoU over 0.3 with a ground-truth box and the IoU is the highest;
3. An anchor that has an IoU lower than 0.3 is regarded as a negative sample;
4. Anchors that do not belong to the above cases are discarded during the training process.

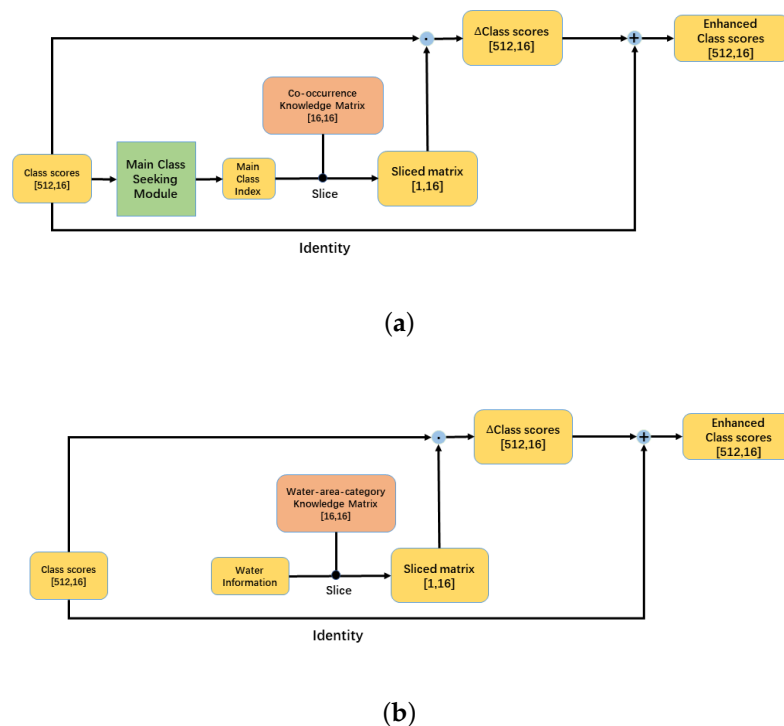
#### 4.3. Oriented RCNN Head with the Knowledge Inference Module

In this section, we apply our proposed Knowledge Inference module to the Oriented RCNN head in order to reduce missed and wrong detections by improving the predicted class scores. Specifically, the proposed module is applied on two kinds of knowledge: conditional co-occurrence knowledge and water area knowledge. Thus, the proposed module has two similar inferencing modes. The details can be seen in the middle part of Figure 3.

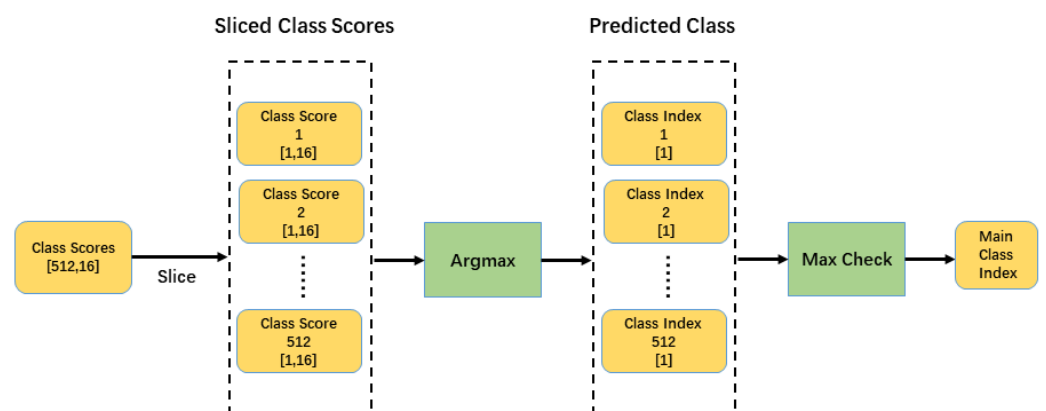
The oriented RCNN head first takes  $\{P_2, P_3, P_4, P_5, \text{ and } P_6\}$  and the oriented proposals from the oriented RPN as the input. In detail, feature vectors are obtained by rotating the RoI alignment to extract rotated RoI features according to the oriented proposals and transform them into fixed-length vectors. This is followed by two fully-connected layers. Then, we use two fully-connected sibling layers outputting classification scores and location predictions. For each image, we generate 512 predictions. Thus, the classification scores are denoted by the tensor of shape  $[512, K + 1]$ , where  $K + 1$  denotes the number of classes plus the background, and the location predictions are denoted by the tensor of shape  $[512, 5]$ . With the aim of optimizing the predicted classification scores with the knowledge matrix, we propose the Knowledge Inference Module, the structure of which is illustrated in Figure 7. To be specific, Figure 7a shows the structure of the Knowledge Inference module applied to class conditional co-occurrence knowledge. The class scores  $[512, 16]$  are first fed into the main-class-seeking-module to compute the major class in the image outputting the index of the main class. Then, the conditional co-occurrence matrix is sliced in terms of the main class index. The sliced matrix denotes the relationship between the main class and other classes, which is represented by the tensor of shape  $[1, 16]$ . Therefore, the  $\Delta$  class score is a tensor with knowledge integrated. It is obtained by dot-multiplying class scores  $[512, 16]$  and transposing the sliced matrix  $[16, 1]$ . However, our initial idea is to use knowledge to guide detection, so we apply a residual structure [19] into our proposed module, avoiding degradation brought about by using  $\Delta$  class scores only. Enhanced class scores are the result of  $\Delta$  class scores plus class scores. Additionally, the Knowledge

Inference module on water area knowledge is shown in Figure 7b and is similar to that of the category conditional co-occurrence matrix. The difference is that the Main Class Seeking module and the main class index are replaced by water information showing whether there is a water area in the image.

The structure of the Main-Class-Seeking module is illustrated in Figure 8. We first slice the tensor class scores  $[512, 16]$  into 512 small tensors  $[1, 16]$  and encode them into values of 1 to 512. Then, the sliced class scores are fed into the  $\text{argmax}()$  function outputting the classes with the highest classification scores. The function  $\text{max check}()$  is used to count the number of each category in the 512 predictions. This is performed sequentially, where the class with largest number is the main class.



**Figure 7.** Structures of the Knowledge Inference module applied on two kinds of knowledge: (a) the structure of the Knowledge Inference module applied on conditional co-occurrence knowledge; (b) the structure of the Knowledge Inference module applied on water area knowledge.



**Figure 8.** The Main Class Seeking module consists of a slice operation,  $\text{argmax}()$  function and a  $\text{max check}()$  function.

#### 4.4. Loss Function

To train the Oriented RPN and Oriented RCNN head, we introduce Cross-Entropy Loss  $L_{cls}$  for the classification task and Smooth L1 Loss  $L_{reg}$  for the regression task. The whole loss function  $L$  is defined as follows:

$$L(p_i, t_i) = \frac{1}{N} \sum_i^N L_{cls}(p_i, p_i^*) + \frac{1}{N} \sum_i^N p_i^* L_{reg}(t_i, t_i^*) \quad (10)$$

$$L_{cls}(p_i, p_i^*) = -[p_i^* \log(p_i) + (1 - p_i^*) \log(1 - p_i)] \quad (11)$$

$$L_{reg}(t_i, t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 & \text{if } |t_i - t_i^*| < 1 \\ |t_i - t_i^*| - 0.5 & \text{otherwise} \end{cases} \quad (12)$$

where  $N$  and  $i$  are, respectively, the number and index of the predicted anchors in a image,  $p_i$  is the probability of predicted anchors,  $p_i^*$  denotes the ground-truth label that belongs to  $\{0, 1\}$ , i.e., negative and positive, and  $t_i$  and  $t_i^*$  are the predicted box and the ground-truth box.

### 5. Experiments

In this section, we introduce the two geospatial object data sets used in this work. Then, evaluation metrics and implementation details are illustrated.

#### 5.1. Data Sets

To evaluate the proposed method, we conduct experiments on two public aerial image data sets, i.e., DOTA and DIOR.

DOTA is the most popular large-scale data set for geospatial object detection, containing 2806 images and 188,282 instances with arbitrary-oriented objects. Moreover, there are 15 classes in the data set: bridge, harbor, ship, plane, helicopter, small vehicle, large vehicle, baseball diamond, ground track field, tennis court, basketball court, soccer ball field, roundabout, swimming pool, and storage tank. The image width ranges from 800 to 4000 pixels. In this work, the training set was used for training, and the validation set was used for evaluation.

DIOR is another data set that is widely used for geospatial object detection. It contains 23,463 optimal remote sensing images and 192,472 object instances annotated by a horizontal bounding box. There are 20 object classes in total, namely, airplane, airport, baseball field, basketball court, bridge, chimney, dam, expressway service area, expressway toll station, harbor, golf course, ground track field, overpass, ship, stadium, storage tank, tennis court, train station, vehicle, and windmill. The image size of the DIOR data set is  $800 \times 800$  pixels. We trained the network on the training set and evaluated the method on the validation set.

#### 5.2. Evaluation Metrics

To evaluate the performance of the proposed method, we utilized four popular evaluation metrics, i.e., precision, recall, average precision, and mean average precision, the calculation formulas of which are shown as follows:

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$TN$ ,  $FN$ , and  $FP$  denote the number of true positives, the number of false negatives, and the number of false positives, respectively. *Precision* measures the number of correctly identified positive detections of the total number of positive detections and *Recall* measures the fraction of correctly identified positive detections of all positive samples.

$AP$  is computed by calculating the average value of precision from recall = 0 to recall = 1.

$$AP = \int_0^1 P(R) dR \quad (15)$$

$mAP$  is used to describe the multi-class object detection performance.

$$mAP = \frac{1}{N_{class}} \sum_{j=1}^{N_{class}} \int_0^1 P_j(R_j) dR_j \quad (16)$$

where  $N_{class}$  is the number of data set classes,  $j$  denotes the index of the class, and  $P_j$  and  $R_j$  are the precision rate and recall rate of the  $j$ -th class.

### 5.3. Implementation Details

The experiments were conducted on a single CPU, Intel Xeon CPU E5-2650 V4 at 2.20 GHz with a single GPU, NVIDIA Tesla P40 24 GB. The operating system was Ubuntu 18.04. The MMrotate [35] repository provided the training strategy. The size of the training image was  $1024 \times 1024$ , and the original DOTA images were split. All images in DIOR were  $800 \times 800$  in size. Thus, there was no need to split the DIOR images. The objects in DIOR were annotated in the horizontal direction by the left-top vertex  $(x_1, y_1)$  and right-bottom vertex  $(x_2, y_2)$ . Thus, we converted the annotations into a form suitable for the Oriented RCNN:  $(x_1, y_1, x_2, y_1, x_2, y_2, x_1, y_2)$ , corresponding to the vertices of oriented ground truth box in clockwise order. As for the hyperparameters, the optimizer was the stochastic gradient descent (SGD) with a learning rate of 0.005, a momentum of 0.9, and a weight decay of 0.0001. The batch size was 1, and the number of training epochs was 12.

## 6. Results

In this section, the results of the experiments on DOTA and DIOR are displayed.

### 6.1. DOTA Results

We applied the knowledge inference module to two kinds of knowledge: class co-occurrence knowledge and water area knowledge. The results show that our method achieved increases in mAP of 1.0% and 0.6%, respectively. Table 3 reports the comparison between the baseline model Oriented RCNN and our proposed method, in which the proposed method basically maintains the performance for both kinds of knowledge and improves the accuracy of several classes, for example 8.7% for the term helicopter with conditional co-occurrence knowledge, and 2.9% for the soccer ball field with water area knowledge.

**Table 3.** Comparison between the baseline model Oriented RCNN and our proposed method on DOTA data set.

| METHOD          | PL   | BD   | BR   | GTF  | SV   | LV   | SH   | TC   | BC   | ST   | SBF  | RA   | HA   | SP   | HC   | mAP  |
|-----------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| $R^3$ Det [17]  | 88.8 | 67.4 | 44.1 | 69.0 | 62.9 | 71.7 | 78.7 | 89.9 | 47.3 | 61.2 | 47.4 | 59.3 | 59.2 | 51.7 | 24.3 | 61.5 |
| CSL [7]         | 88.1 | 72.2 | 39.8 | 63.8 | 64.3 | 71.9 | 78.5 | 89.6 | 52.4 | 61.0 | 50.5 | 66.0 | 56.6 | 50.1 | 27.5 | 62.2 |
| $S^2$ A-net [9] | 89.1 | 72.0 | 45.6 | 64.8 | 65.0 | 74.8 | 79.5 | 90.1 | 60.2 | 67.3 | 49.3 | 62.2 | 60.6 | 53.4 | 37.6 | 64.8 |
| FR-O [5]        | 89.3 | 76.0 | 49.3 | 74.7 | 68.1 | 75.5 | 87.1 | 90.7 | 64.2 | 62.3 | 57.0 | 65.8 | 66.6 | 59.6 | 38.2 | 68.3 |
| RoI Trans [6]   | 89.9 | 76.5 | 48.1 | 73.1 | 68.7 | 78.2 | 88.7 | 90.8 | 73.6 | 62.7 | 62.0 | 63.4 | 73.7 | 57.2 | 47.9 | 70.3 |
| Baseline [15]   | 89.8 | 75.7 | 50.2 | 77.3 | 69.4 | 84.8 | 89.3 | 90.8 | 69.2 | 62.6 | 63.1 | 65.0 | 75.3 | 57.5 | 45.3 | 71.0 |
| Water area      | 89.6 | 75.6 | 50.3 | 76.4 | 68.4 | 84.3 | 89.4 | 90.7 | 72.9 | 62.6 | 66.0 | 67.2 | 75.6 | 56.5 | 48.8 | 71.6 |
| Co-occurrence   | 89.6 | 76.0 | 50.7 | 77.0 | 68.3 | 84.4 | 89.3 | 90.7 | 73.6 | 62.4 | 63.8 | 66.8 | 75.1 | 57.6 | 54.0 | 72.0 |

The baseline is the Oriented RCNN [15], the Water area denotes the Knowledge Inference module on water area knowledge, and Co-occurrence is the Knowledge Inference module applied on conditional co-occurrence knowledge, where: PL: plane, BD: baseball diamond, BR: bridge, GTF: ground field track, SV: small vehicle, LV: large vehicle, SH: ship, TC: tennis court, BC: basketball court, ST: storage tank, SBF: soccer ball field, RA: roundabout, HA: harbor, SP: swimming pool, and HC: helicopter.

Moreover, to a certain degree, some missed detections and wrong detections were improved. Figures 9 and 10 display the reductions in missed detection and wrong detection using conditional co-occurrence knowledge and water area knowledge, respectively. In each subfigure, the left half is the result of the baseline model and the right half is the result of the proposed method. We use yellow circles to draw missed detections and red circles to draw false detections. Additionally, the first three subfigures shown in Figures 9 and 10 display missed detections, and the second three subfigures in Figures 9 and 10 display the false detections. The positive detections are shown in Figure 11.



**Figure 9.** Cont.



(c)

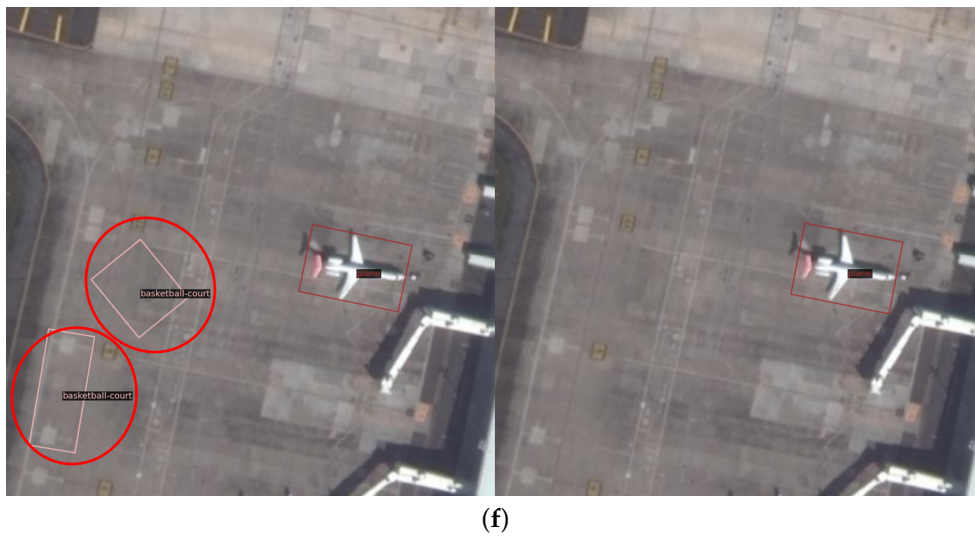


(d)

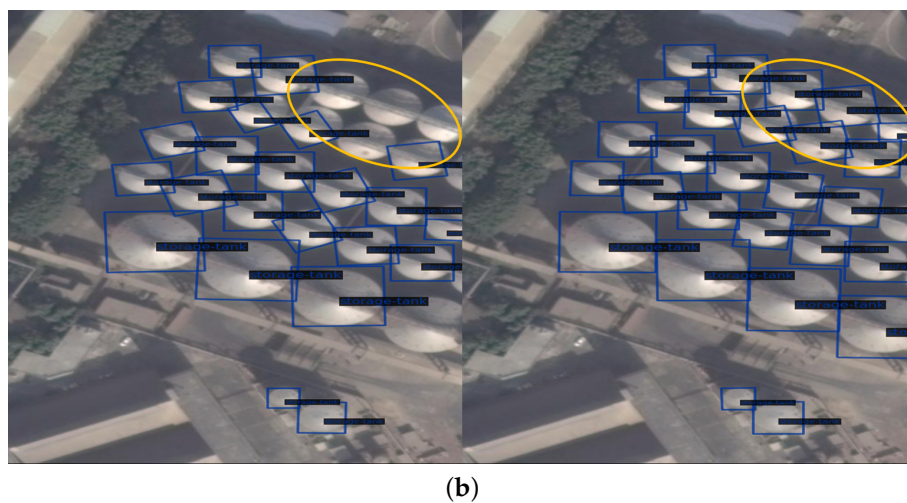
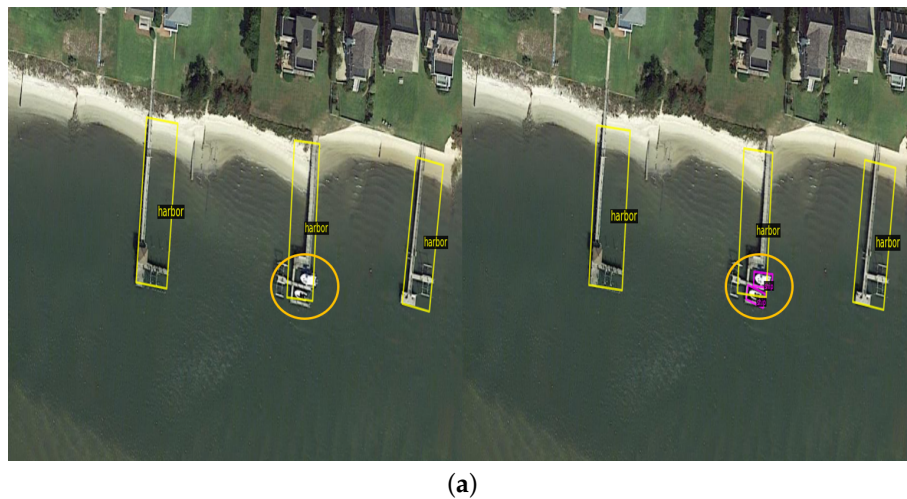


(e)

**Figure 9.** *Cont.*



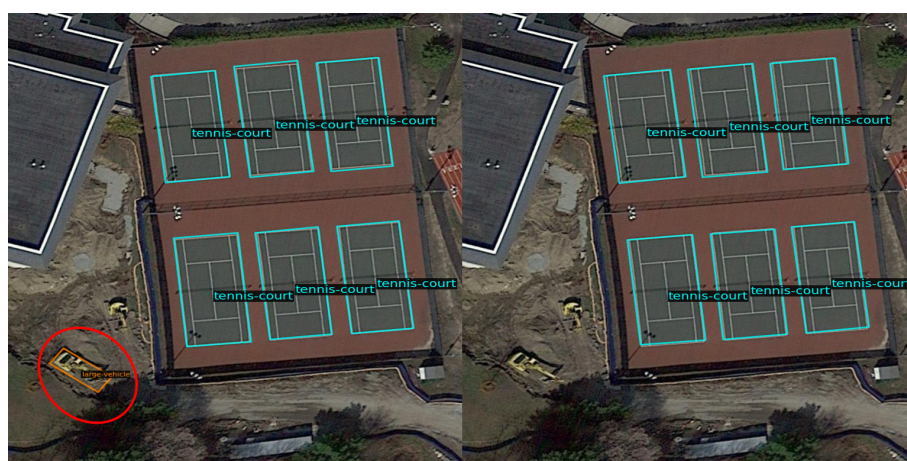
**Figure 9.** Visualization of the results of the Knowledge Inference module applied to category occurrence knowledge. (a) missed detection of swimming pools; (b) missed detection of roundabouts; (c) missed detection of basketball courts; (d) false detection of baseball diamonds; (e) false detection of storage tanks; (f) false detection of basketball courts.



**Figure 10.** Cont.



(c)

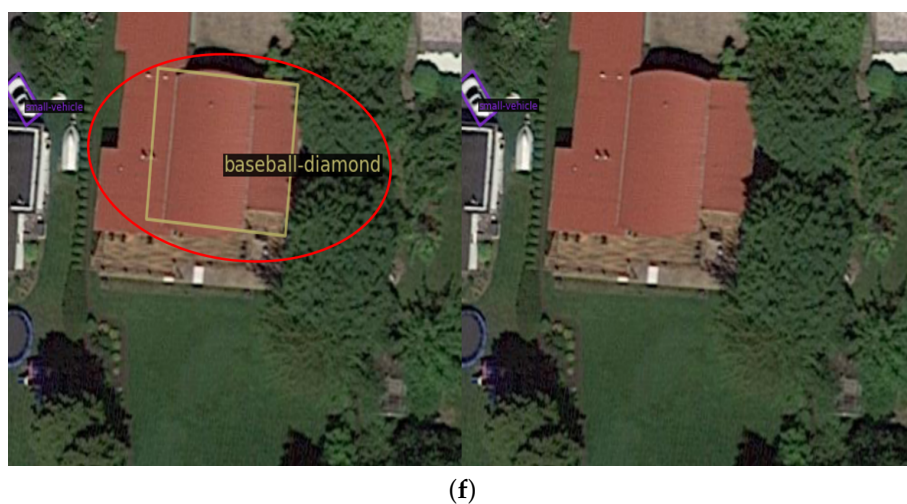


(d)

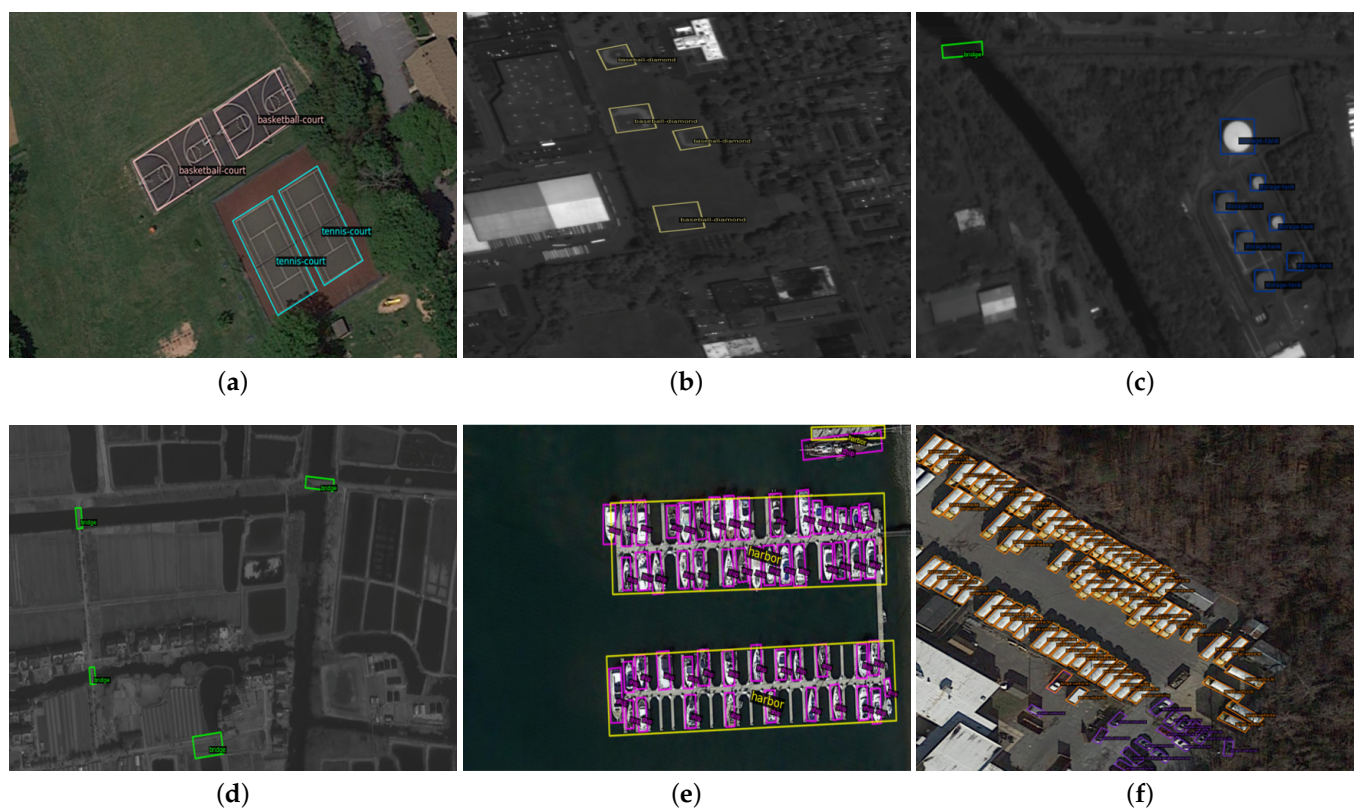


(e)

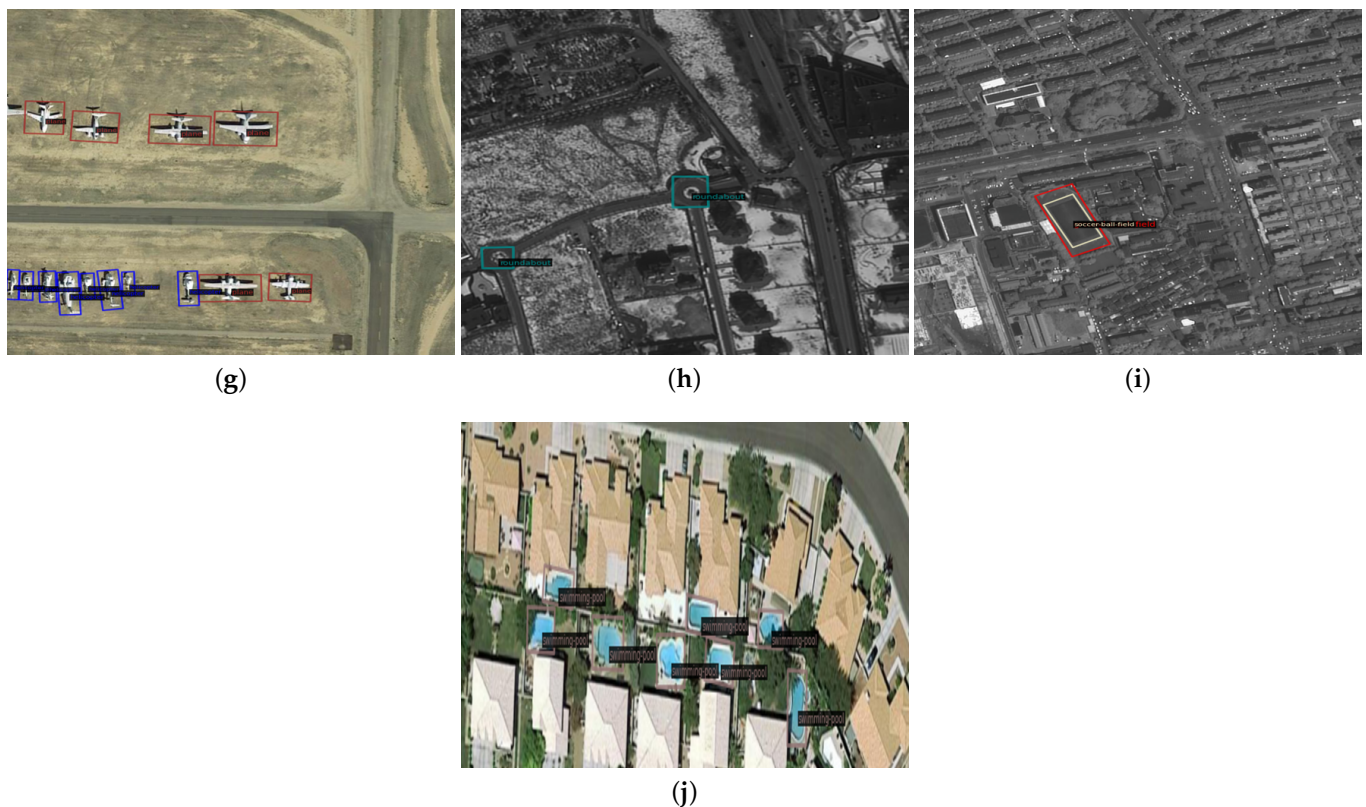
Figure 10. Cont.



**Figure 10.** Visualization of the results of the Knowledge Inference module applied to water area knowledge. (a) missed detections of ships in the middle of the image; (b) missed detection storage tanks; (c) missed detection of harbors; (d) false detection of large vehicles; (e) false detection of harbors; (f) false detection of baseball diamonds.



**Figure 11.** Cont.



**Figure 11.** Visualization of the positive results. (a) basketball courts and tennis courts; (b) baseball diamonds; (c) bridge and storage tanks; (d) bridges; (e) harbors and ships; (f) large vehicles and small vehicles; (g) planes and helicopters; (h) roundabouts; (i) ground field tracks and soccer ball fields; (j) swimming pools.

Improvement occurs due to the utilization of knowledge. On the one hand, knowledge is used to optimize the predicted class scores. Thus, the performance of the classification is promoted; on the other hand, the class predictions optimized by knowledge can help the network iterate better during backpropagation. As a result, the more powerful features can be extracted by the network.

In terms of the inferencing speed, we compared the baseline and Knowledge Inference module for two kinds of knowledge, as shown in Table 4. With the Knowledge Inference module, there was no significant drop in speed.

**Table 4.** Comparison of the inferencing speed and accuracy between the baseline and proposed methods for two kinds of knowledge in the DOTA data set.

| METHOD                    | FPS  | mAP  |
|---------------------------|------|------|
| Baseline                  | 11.8 | 71.0 |
| Water area                | 11.7 | 71.6 |
| Conditional co-occurrence | 11.5 | 72.0 |

## 6.2. DIOR Results

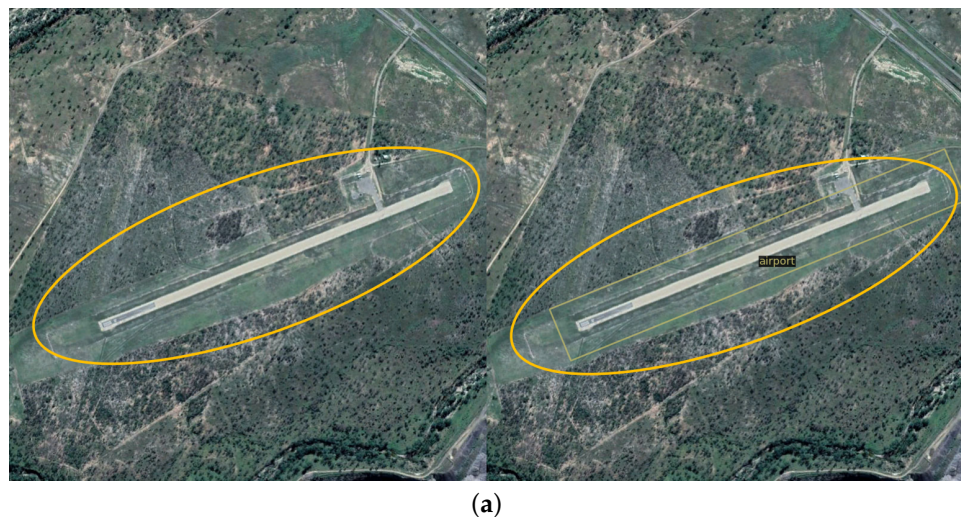
The experiments conducted on the DIOR data set achieved a better performance. The ap values and mAP values are shown in Table 5. As can be seen, both kinds of knowledge had beneficial impacts on the detection performance, and the mAP values increased by 0.5% and 3.9%, respectively. Similarly, missed detections and wrong detections were effectively eliminated.

**Table 5.** Comparison between the baseline model Oriented RCNN and our proposed method on DIOR data set .

| METHOD          | APL  | APT  | BF   | BC   | BR   | CM   | DA   | ESA  | EST  | GF   |
|-----------------|------|------|------|------|------|------|------|------|------|------|
| $R^3$ Det [17]  | 89.6 | 6.40 | 89.5 | 71.2 | 14.4 | 81.7 | 8.90 | 26.5 | 48.1 | 31.8 |
| CSL [7]         | 90.9 | 2.60 | 89.4 | 71.5 | 7.10 | 81.8 | 9.30 | 31.4 | 41.5 | 58.1 |
| $S^2$ A-Net [9] | 90.8 | 14.0 | 89.9 | 72.7 | 17.6 | 81.7 | 9.50 | 32.9 | 50.1 | 50.9 |
| RoI Trans [6]   | 90.8 | 12.1 | 90.8 | 79.8 | 22.9 | 81.8 | 8.20 | 51.3 | 54.1 | 60.7 |
| FR-O [5]        | 90.9 | 13.1 | 90.7 | 79.9 | 22.0 | 81.8 | 10.4 | 49.8 | 53.1 | 58.5 |
| Baseline        | 90.9 | 17.0 | 90.7 | 80.6 | 33.5 | 81.8 | 19.2 | 59.8 | 53.1 | 56.7 |
| Water area      | 90.9 | 21.2 | 90.7 | 80.3 | 34.2 | 81.8 | 20.7 | 60.0 | 52.9 | 55.1 |
| Co-occurrence   | 90.9 | 23.3 | 90.8 | 80.9 | 38.0 | 81.8 | 20.5 | 62.2 | 53.7 | 61.3 |
| GTF             | HA   | OPS  | SP   | STD  | ST   | TC   | TS   | VEH  | WD   | mAP  |
| 65.6            | 8.20 | 33.4 | 69.2 | 51.9 | 72.9 | 81.1 | 21.4 | 54.2 | 44.7 | 48.5 |
| 63.2            | 17.5 | 26.6 | 69.3 | 53.8 | 72.8 | 81.6 | 18.4 | 47.2 | 46.3 | 49.0 |
| 70.9            | 16.3 | 43.6 | 80.1 | 52.5 | 75.7 | 81.7 | 23.9 | 59.0 | 45.6 | 53.0 |
| 77.2            | 30.6 | 40.5 | 89.9 | 88.2 | 79.5 | 81.8 | 20.6 | 67.9 | 55.1 | 59.2 |
| 75.4            | 35.5 | 41.6 | 89.1 | 85.4 | 79.3 | 81.8 | 32.7 | 66.4 | 55.5 | 59.6 |
| 76.9            | 26.1 | 54.5 | 89.9 | 88.8 | 79.6 | 81.8 | 30.2 | 68.3 | 55.1 | 61.7 |
| 76.7            | 26.8 | 54.6 | 89.8 | 88.3 | 79.5 | 81.8 | 35.5 | 68.7 | 55.6 | 62.2 |
| 81.3            | 32.1 | 56.6 | 90.0 | 88.3 | 79.7 | 90.1 | 44.3 | 69.2 | 56.5 | 64.6 |

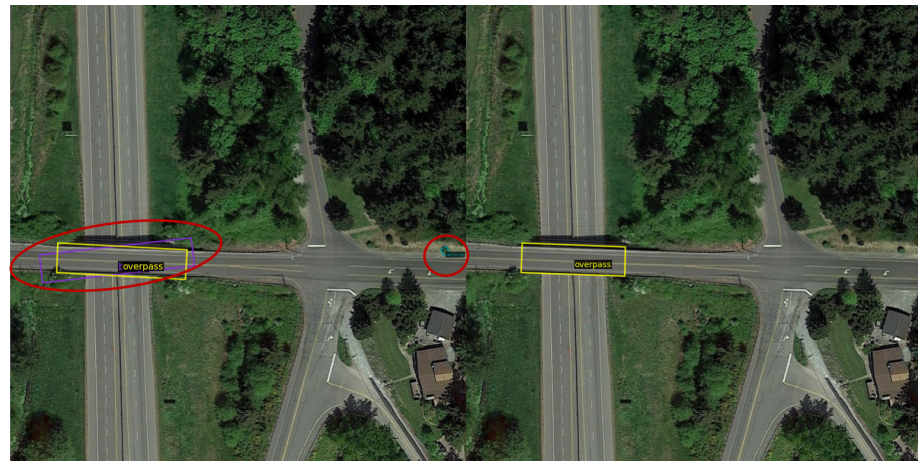
The baseline is the Oriented RCNN, the Water area denotes the Knowledge Inference module's knowledge on a water area, and conditional co-occurrence is the Knowledge Inference module for category conditional co-occurrence knowledge. APL: airplane, APT: airport, BF: baseball field, BC: basketball court, BR: bridge, CM: chimney, DA: dam, ESA: expressway service area, ETS: expressway-toll-station, GF: golf field, GTF: ground track field, HA: harbor, OPS: overpass, SP: ship, STM: stadium, ST: storage tank, TC: tennis court, TS: trainstation, VEH: vehicle, WD: windmill.

For the DIOR data set, we also visualized the impacts of two kinds of knowledge on the baseline, as shown in Figures 12 and 13. As for the visualization of the DOTA data set, the yellow circle and red circle denote missed detections and false detections, respectively. The positive detections are shown in Figure 14.

**Figure 12.** Cont.



(b)

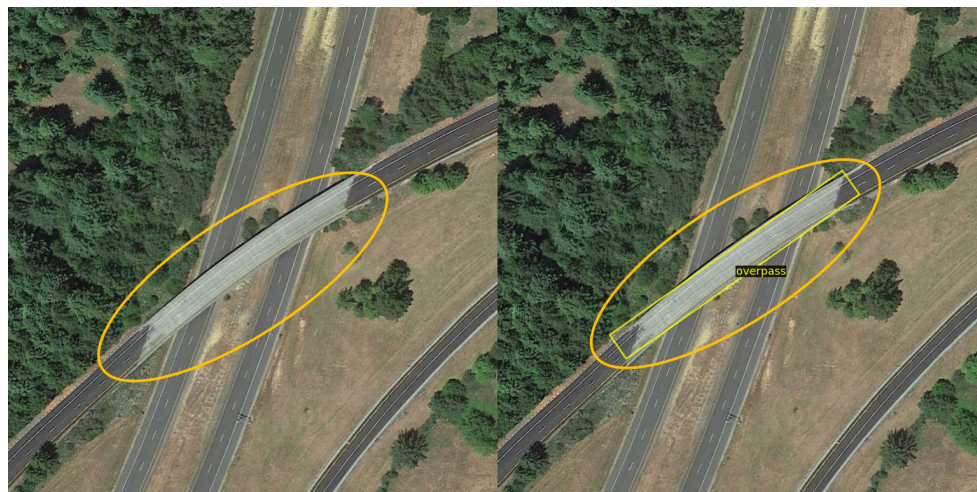


(c)



(d)

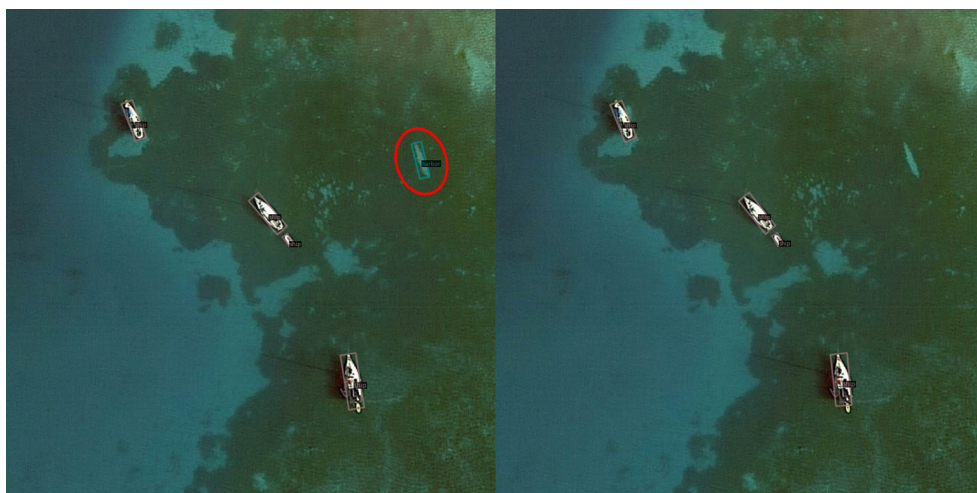
**Figure 12.** Visualization of the results of the Knowledge Inference module applied on category occurrence knowledge: (a) missed detection of airports; (b) missed detection of expressway service areas; (c) false detection of bridges in the purple box and vehicles; (d) false detection of expressway toll stations.



(a)

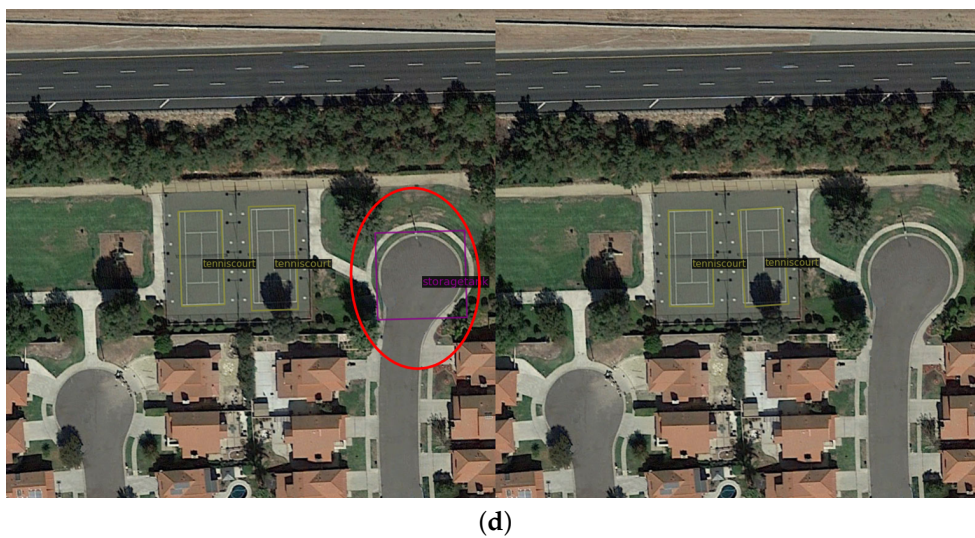


(b)

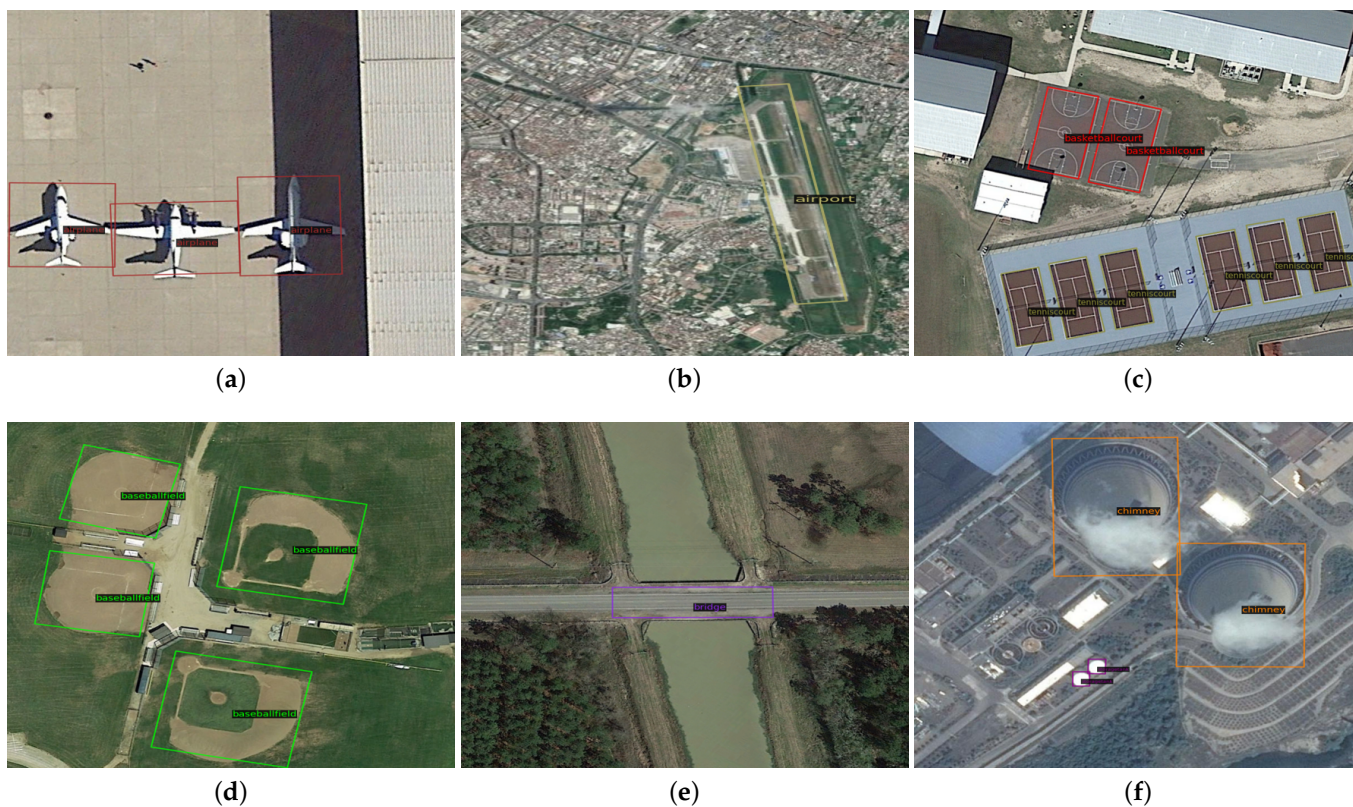


(c)

Figure 13. Cont.



**Figure 13.** Visualization of the results of the Knowledge Inference module applied on category occurrence knowledge. (a) missed detection of overpasses; (b) missed detection of windmills; (c) false detection of harbors; (d) false detection of storage tanks.



**Figure 14.** Cont.



**Figure 14.** Visualization of the positive results. (a) airplanes; (b) airports; (c) basketball courts and tennis courts; (d) baseball fields; (e) bridges; (f) chimneys and storage tanks; (g) dams; (h) expressway service areas; (i) golf fields; (j) harbors and ships; (k) overpasses; (l) ground track fields and stadiums; (m) train stations; (n) expressway toll stations; (o) windmills.

A comparison of the inferencing speed and accuracy between the baseline and proposed methods for two kinds of knowledge is shown in Table 6. As can be seen, the Knowledge Inference module improved the detection accuracy with a negligible negative influence on the inferencing speed.

**Table 6.** Comparison of the inferencing speed and accuracy between the baseline and proposed methods for two kinds of knowledge in the DIOR data set.

| METHOD                    | FPS | mAP  |
|---------------------------|-----|------|
| Baseline                  | 9.3 | 61.7 |
| Water area                | 9.1 | 62.2 |
| Conditional co-occurrence | 9.1 | 64.6 |

## 7. Conclusions

In this paper, in order to utilize knowledge to reduce false detections and missed detections caused by variation in the object appearance, varied object sizes, and complicated backgrounds, a series of steps were taken. We first established a knowledge matrix between the classes and a knowledge matrix between water areas and classes by analyzing the training set and proposed a novel equation, which can effectively avoid generalization degradation, to transform the relationship into form applicable for inferencing. Then, we proposed a method, the Knowledge Inference module, for integrating knowledge into object detection. The experiments were conducted on two public remote sensing data sets: DOTA and DIOR. The experimental results show that, compared to the baseline model, the proposed method achieved higher mAP values with fewer false detections and missed detections at an almost equal inferencing speed.

**Author Contributions:** Conceptualization, K.Z. and Y.D.; methodology, K.Z. and Y.D.; software, K.Z.; resources, W.X.; writing—original draft preparation, K.Z.; writing—review and editing, Y.D. and W.X.; visualization, K.Z. and Y.S.; supervision, P.H.; funding acquisition, Y.D. and W.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Inner Mongolia Application Technology Research and Development Funding Project (2019GG138) and the Natural Science Foundation of the Inner Mongolia Autonomous Region (2020ZD18).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

**Table A1.** The number of images where the category occurs  $N_{img}(l)$  and the probabilities of class occurrences  $n(l)$ .

| Object Categories       | $N_{img}(l)$ | $n(l)$ |
|-------------------------|--------------|--------|
| airplane                | 344          | 0.0586 |
| airport                 | 326          | 0.0556 |
| baseball field          | 552          | 0.0941 |
| basketball court        | 336          | 0.0573 |
| bridge                  | 378          | 0.0644 |
| chimney                 | 202          | 0.0344 |
| dam                     | 238          | 0.0406 |
| expressway service area | 279          | 0.0475 |
| expressway toll station | 285          | 0.0486 |
| golf field              | 216          | 0.0368 |
| ground track field      | 537          | 0.0916 |
| harbor                  | 329          | 0.0561 |
| overpass                | 410          | 0.0699 |
| ship                    | 649          | 0.1107 |
| stadium                 | 289          | 0.0493 |
| storage tank            | 390          | 0.0665 |
| tennis court            | 605          | 0.1032 |
| train station           | 244          | 0.0416 |
| vehicle                 | 1561         | 0.2662 |
| windmill                | 404          | 0.0689 |

**Table A2.** Probabilities of classes of DIOR appearing with water area and not appearing with water area. Column  $n(l|w)$  denotes the probability of category  $l$  appearing with water area;  $n(l|\bar{w})$  is the probability that category  $l$  appears with no water area.

| Object Categories       | $n(l w)$ | $n(l \bar{w})$ |
|-------------------------|----------|----------------|
| airplane                | 0.0412   | 0.9588         |
| airport                 | 0.4482   | 0.5518         |
| baseball field          | 0.0714   | 0.9286         |
| basketball court        | 0.0981   | 0.9019         |
| bridge                  | 0.9525   | 0.0475         |
| chimney                 | 0.1228   | 0.8772         |
| dam                     | 1        | 0              |
| expressway service area | 0.2719   | 0.7281         |
| expressway toll station | 0.1335   | 0.8665         |
| golf field              | 0.9114   | 0.0886         |
| ground track field      | 0.1293   | 0.8707         |
| harbor                  | 1        | 0              |
| overpass                | 0.1179   | 0.8821         |
| ship                    | 0.9998   | 0.0002         |
| stadium                 | 0.1126   | 0.8874         |
| storage tank            | 0.3126   | 0.6874         |
| tennis court            | 0.1366   | 0.8634         |
| train station           | 0.2398   | 0.7602         |
| vehicle                 | 0.2140   | 0.7860         |
| windmill                | 0.0489   | 0.9511         |

## References

- Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [\[CrossRef\]](#)
- Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [\[CrossRef\]](#)
- Wu, X.; Li, W.; Hong, D.; Tao, R.; Du, Q. Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 91–124. [\[CrossRef\]](#)
- Fascista, A. Toward Integrated Large-Scale Environmental Monitoring Using WSN/UAV/Crowdsensing: A Review of Applications, Signal Processing, and Future Perspectives. *Sensors* **2022**, *22*, 1824. [\[CrossRef\]](#) [\[PubMed\]](#)
- Mo, N.; Yan, L. Improved faster RCNN based on feature amplification and oversampling data augmentation for oriented vehicle detection in aerial images. *Remote Sens.* **2020**, *12*, 2558. [\[CrossRef\]](#)
- Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning RoI transformer for oriented object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2849–2858.
- Yang, X.; Yan, J. Arbitrary-oriented object detection with circular smooth label. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 677–694.
- Guo, Z.; Liu, C.; Zhang, X.; Jiao, J.; Ji, X.; Ye, Q. Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8792–8801.
- Han, J.; Ding, J.; Li, J.; Xia, G.S. Align deep features for oriented object detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–11. [\[CrossRef\]](#)
- Torralba, A.; Oliva, A.; Castelano, M.S.; Henderson, J.M. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychol. Rev.* **2006**, *113*, 766. [\[CrossRef\]](#) [\[PubMed\]](#)
- Li, Z.; Wu, Q.; Cheng, B.; Cao, L.; Yang, H. Remote sensing image scene classification based on object relationship reasoning CNN. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 1–5. [\[CrossRef\]](#)
- Xu, H.; Jiang, C.; Liang, X.; Lin, L.; Li, Z. Reasoning-rcnn: Unifying adaptive global reasoning into large-scale object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6419–6428.
- Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
- Fang, Y.; Kuan, K.; Lin, J.; Tan, C.; Chandrasekhar, V. Object detection meets knowledge graphs. In Proceedings of the International Joint Conferences on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017.

15. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 3520–3529.
16. Ming, Q.; Miao, L.; Zhou, Z.; Dong, Y. CFC-Net: A critical feature capturing network for arbitrary-oriented object detection in remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [\[CrossRef\]](#)
17. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 3163–3171.
18. Han, J.; Ding, J.; Xue, N.; Xia, G.S. Redet: A rotation-equivariant detector for aerial object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–23 June 2021; pp. 2786–2795.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. Chen, C.; Gong, W.; Chen, Y.; Li, W. Object detection in remote sensing images based on a scene-contextual feature pyramid network. *Remote Sens.* **2019**, *11*, 339. [\[CrossRef\]](#)
21. Liu, Y.; Wang, R.; Shan, S.; Chen, X. Structure inference net: Object detection using scene-level context and instance-level relationships. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6985–6994.
22. Siris, A.; Jiao, J.; Tam, G.K.; Xie, X.; Lau, R.W. Scene context-aware salient object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 4156–4166.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Processing Syst.* **2017**, *30*. [\[CrossRef\]](#)
24. Li, K.; Cheng, G.; Bu, S.; You, X. Rotation-insensitive and context-augmented object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 2337–2348. [\[CrossRef\]](#)
25. Zhang, G.; Lu, S.; Zhang, W. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 10015–10024. [\[CrossRef\]](#)
26. Zhang, K.; Wu, Y.; Wang, J.; Wang, Y.; Wang, Q. Semantic context-aware network for multiscale object detection in remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
27. Liu, J.; Li, S.; Zhou, C.; Cao, X.; Gao, Y.; Wang, B. SRAF-Net: A Scene-Relevant Anchor-Free Object Detection Network in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [\[CrossRef\]](#)
28. Feng, X.; Han, J.; Yao, X.; Cheng, G. TCANet: Triple context-aware network for weakly supervised object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 6946–6955. [\[CrossRef\]](#)
29. Cheng, B.; Li, Z.; Xu, B.; Dang, C.; Deng, J. Target detection in remote sensing image based on object-and-scene context constrained CNN. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
30. Xu, H.; Jiang, C.; Liang, X.; Li, Z. Spatial-aware graph relation network for large-scale object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9298–9307.
31. Xu, H.; Fang, L.; Liang, X.; Kang, W.; Li, Z. Universal-rcnn: Universal object detector via transferable graph r-cnn. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12492–12499.
32. Shu, X.; Liu, R.; Xu, J. A Semantic Relation Graph Reasoning Network for Object Detection. In Proceedings of the 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), Suzhou, China, 14–16 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1309–1314.
33. Jiang, C.; Xu, H.; Liang, X.; Lin, L. Hybrid knowledge routed modules for large-scale object detection. *Adv. Neural Inf. Processing Syst.* **2018**, *31*. [\[CrossRef\]](#)
34. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
35. Zhou, Y.; Yang, X.; Zhang, G.; Wang, J.; Liu, Y.; Hou, L.; Jiang, X.; Liu, X.; Yan, J.; Lyu, C.; et al. MMRotate: A Rotated Object Detection Benchmark using PyTorch. *arXiv* **2022**, arXiv:2204.13317.