Article
3D Pose Estimation for Object Detection in Remote Sensing Images

Jin Liu * and Yongjian Gao

State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China; gaoyongjian@whu.edu.cn

* Correspondence: jliu@sgg.whu.edu.cn

Received: 3 January 2020; Accepted: 22 February 2020; Published: 25 February 2020

Abstract: 3D pose estimation is always an active but challenging task for object detection in remote sensing images. In this paper, we present a new algorithm for predicting an object’s 3D pose in remote sensing images, called Anchor Points Prediction (APP). Compared to previous methods, such as RoI Transform, our object results of the final output can obtain direction information. We predict the object’s multiple feature points based on the neural network to obtain the homograph transformation relationship between object coordinates and image coordinates. The resulting 3D pose can accurately describe the three-dimensional position and attitude of the object. At the same time, we redefine the method IoU_APP for calculating the direction and posture of the object. We tested our algorithm on the HRSC2016 dataset and the DOTA dataset with accuracy rates of 0.863 and 0.701, respectively. The experimental results show that the accuracy of the APP algorithm is significantly improved. At the same time, the algorithm can achieve one-stage prediction, which makes the calculation process easier and more efficient.

Keywords: object detection; remote sensing images; 3D pose

1. Introduction

In recent years, with the deepening of research and the improvement of computing power, deep learning has become more and more widely used in various fields. At the same time, the object detection algorithm has made great progress so far. In particular, remote sensing images has been a specific but active topic in computer vision [1,2]. Recent progresses in object detection in aerial images have benefited a lot from the R-CNN frameworks [1,3–6]. These methods use horizontal bounding boxes as the region of IoU and then rely on region-based features for category identification [2,7,8]. Faster-RCNN [4,5] leads to an elegant and effective solution where proposal computation is nearly cost-free given the detection network’s computation. A multi-stage object detection framework, the Cascade R-CNN, is proposed for the design of high-quality object detectors [6,9]. Additionally, FPN uses feature pyramids for object detection [10]; Yolt achieves object detection of high-resolution remote sensing images based on Yolo v3 [11,12]; and Yolo v3 is significantly faster than other methods in achieving the same accuracy [13]. These classic algorithms have different adaptation scenarios and greatly promote the development of this field. However, in remote sensing images, the object is often placed obliquely, so using an inclined box to detect the object will be more adaptive to the scene. These algorithms use a horizontal rectangular box to detect the object, so it does not accurately reflect the object pose of the remote sensing image to some extent. Also, these horizontal RoIs typically lead to misalignments between the bounding boxes and objects [8,14,15]. The RoI Transform algorithm locates the inclined box by predicting the rotation angle of the object box [8,16]. However, this algorithm has some problems. The first problem is that the rotation angle \( \theta \) of the regression inclined box is
ambiguous in most cases. This means that $\theta$ is equal to $0^\circ$ and $180^\circ$ corresponding to IoU is equal, but if the algorithm does not contain direction information, it will be considered the same type. The second problem is efficiency. It is a two-stage algorithm and the localization method relies entirely on Faster-RCNN [5,8]. The algorithm can only use the rectangle obtained by Faster-RCNN, and cannot use the feature information of the object region. CornerNet is a new one-stage approach to object detection by predicting the coordinates of the top-left and bottom-right points that does away with anchor boxes, which is more accurate and efficient [17–20]. However, predicting two points does not fully describe the information of the inclined box [5,21,22].

A 3D pose describes the three-dimensional pose of the camera relative to the object’s own coordinate system, not the pose of the object relative to the ground plane. Whether the reference object has z-coordinates and whether 3D information can be estimated are not the same thing. The object is on a plane and does not affect the rotation and translation of the camera observing the object relative to the three axes in three dimensions. Therefore, even objects on a two-dimensional plane will be observed in three-dimensional space out of the plane, which also has the 3D pose problem.

To solve the above problems, we propose the Anchor Points Prediction (called APP) algorithm. Different from other methods, we predict the position and attitude of the object by at least four corner points through the full convolution network, and can obtain the 3D pose by decomposing the homograph transformation matrix, and the algorithm is more efficient. The corner pooling layer used in the algorithm greatly improves the points prediction accuracy [17].

We give the correspondence between the predicted points and the available object information in Table 1, and a comparison of the traditional method and our method is shown in Figure 1. We have reason to believe that object detection by point prediction will become a new trend in the future.

| Prediction Points | Object Information |
|-------------------|--------------------|
| 2                 | RoI                |
| 4                 | inclined box       |
| $\geq 4$          | 3D pose            |

Table 1. Correspondence between predicted points and available object information [17,23].

![Figure 1](image-url) Object detection comparison. (a)(c) Traditional inclined box. The inclined box has symmetry, so it is not possible to uniquely describe the direction of the object in the 2D image space, so the object has four possible directions. (b)(d) The 3D pose diagram obtained from APP. The X-axis of the object is marked with red, the Y-axis is marked with green, and the Z-axis is marked with blue. The X-axis points to the right side of the object; the negative direction of Y-axis indicates the static direction of the object; and the Z-axis points to the ground.

2. Object Detection Based On APP

Any object detection problems can be attributed to the prediction of key points. The traditional rectangle object detection methods can be attributed to the prediction of two key points, such as
Faster-RCNN, YOLO, and SSD (the upper-left and lower-right points of the rectangle) [5,21,24,25]; the 3D pose of the general object can be attributed to the prediction of eight points; the human body posture OpenPose can be attributed to the prediction of 18 key points of the human body [26]. We attribute the predicted inclined object box to a prediction of four points. The traditional inclined box detection methods have no direction information and may result in high accuracy. In addition, as shown in Figure 2, the boxes of the two objects whose center points are close but opposite in direction may cause the object to be lost in the NMS operation. The full name of NMS is non-maximum supply [27]. This method is used to search the local maximum and suppress the maximum. The purpose of this method is to eliminate redundant frames and find the best location for object detection.

![Figure 2](image)

**Figure 2.** Two objects in opposite directions are prone to loss in NMS operations.

Unlike traditional methods, we define a new calculation method that takes into account the overlap rate and direction consistency between the tilted boxes. Assuming there are two sets of object feature points, \( \{P_{11}...P_{1n}\}, \{P_{21}...P_{2n}\} \), we define the \( \text{IoU}_{APP} \) calculation formula between the two sets of points as follows:

\[
\text{IoU}_{APP} = \frac{1}{1 + \alpha \sqrt{\frac{d_{12}}{d_1 + d_2}}}. \tag{1}
\]

We use \( d_{12} = \sum_{i=1}^{n} (P_{1i} - P_{2i})^2 \), \( d_1 = \sum_{i=1}^{n} (P_{1i} - \overline{P_1})^2 \) and \( d_2 = \sum_{i=1}^{n} (P_{2i} - \overline{P_2})^2 \). \( \overline{P_1} \) and \( \overline{P_2} \) are the coordinates of the center point of point set 1 and point set 2, respectively. The definition mainly considers the deviation of the coordinate offset of the corresponding point from the size of the object itself. This deviation is relative. The larger the deviation, the smaller the \( \text{IoU}_{APP} \). It is clear that the range of values of \( \text{IoU}_{APP} \) is the same as the original IoU definition, which is \([0, 1] \). The larger the \( \text{IoU}_{APP} \), the closer the two object cells are, and the \( \text{IoU}_{APP} \) is equal to 1 when the two object cells are completely identical; the \( \text{IoU}_{APP} \) is infinitely close to 0 when the two object cells are very different.

The two sets of the object feature points may be the two prediction units to be combined in the object detection NMS process, or may be the similarity calculation between ground truth and the predicted values.

![Figure 3](image)

**Figure 3.** Two objects in opposite directions are prone to loss in NMS operations.

According to Figure 3, we can get the calculation formulas for \( \text{IoU}_{APP} \), IoU, and \( \text{IoU}_{RBox} \) as follows, and we can get the relationship curve as shown in Figure 4.
\[ \text{IoU}_{\text{APP}} = \frac{1}{1 + 8\alpha \sin^2 \frac{\theta}{2}}, \alpha = 1, 2, 3, \ldots \]
\[ \text{IoU} = \frac{1}{\left(1 + 2\frac{\cos \theta \sin \theta}{\cos \theta + |\sin \theta|}\right)^2} \]
\[ \text{IoU}_{\text{RBox}} = \frac{1}{|\cos \theta| + |\sin \theta|}. \]

(2)

Figure 4. Relationship between IoU_{\text{APP}}, IoU, and IoU_{\text{RBox}}.

It can be seen from the above Figure that when the object position scale is constant, only the IoU_{\text{APP}} is significantly affected by the object direction angle \( \theta \), so only the IoU_{\text{APP}} can describe the accuracy of the object direction angle. The IoU and IoU_{\text{RBox}} are not affected by \( \theta \) and cannot describe the accuracy of the object direction angle.

The mAP (mean Average Precision) obtained from the experimental data of RoI Transform is based on IoU_{\text{RBox}} [8]. The mAP, which is used to evaluate the accuracy of object detection methods, is based on IoU between prediction boxes and ground truth boxes [28]. As can be seen from the above Figure, as long as the neural network that can recommend the horizontal box aligns with the center point, the IoU_{\text{RBox}} is always greater than 0.5. That is, even if the direction angle prediction is wrong (predicted to be any angle from 0° to 360°), it is also hit when counting mAP, so the resulting mAP is virtually high. Thus, we proposed the solution IoU_{\text{APP}}. The IoU_{\text{APP}} uses the method of regression coordinates to detect the object, and the method of evaluating mAP is more reasonable.

3. Anchor Points Prediction Algorithm

3.1. Neural Network Design

To predict the inclined box of the remote sensing object, we built a full convolutional network that predicts three scales in three different layers, each scale being the output by the APP of three different anchors’ array. Different anchors are used to detect objects with different aspect ratios in the image domain, as shown in Figure 5.
Our custom region layer is used to output the relative coordinates, the categories, and the information about whether the object exists or not. In the region layer, we used yolov3’s definition of anchors to implement n-weight anchor predictions based on the width and height of the object on the imaging surface. Each anchor represents a specific 2D width height object. Following this concept, this particular 2D width and height corresponds to different APP range distributions. Each cell of the output array contains \( n \times (4 + 1 + c + 2 \times 4) \) output neurons. The meaning of the parameters are shown in Table 2.

| Parameter | 4 | n | 1 | c | 2 \times 4 |
|-----------|---|---|---|---|-----------|
| Description | bounding box coordinates | number of anchors | existing object | number of classes | \( (\Delta x_1, \Delta y_1) \) | \( (\Delta x_2, \Delta y_2) \) | \( (\Delta x_3, \Delta y_3) \) | \( (\Delta x_4, \Delta y_4) \) |  |

We define the offset coordinate of point \( i \) in the range of a specific region as \( (p_w x_i, p_h y_i) \). As shown in Figure 6, \( p_w \) and \( p_h \) are the width and height of a particular anchor. We can use Equation (3) to calculate \( P_i \).

\[
\begin{align*}
  u_i &= C_x + \frac{1}{2} + p_w x_i \\
  v_i &= C_y + \frac{1}{2} + p_h y_i
\end{align*}
\]

Then, the actual pixel offset coordinate of point \( i \) relative to the anchor boxes is \( P_i = (u_i, v_i) \). In addition, we define the loss function in the training procedure as follows, and we give the meaning of each parameter in Table 3.
Loss(OutputLayer) = \begin{cases} 
\lambda_{\text{noobj}}(0 - \text{Obj})^2 & \text{No object in local scope} \\
\lambda_{\text{obj}}(1 - \text{Obj})^2 + \lambda_{\text{pts}} \sum_{i=1}^{4} (P_i - \bar{P}_i)^2 + \lambda_{\text{ROI}} \text{Loss}_{\text{ROI}} & \text{object within a local scope}
\end{cases}

(4)

We are more focused on the positioning learning of the inclined box determined by APP, so we reduce the weight of the horizontal box. Then, we use \( \lambda_{\text{ROI}} = 0.01 \) to assist learning, and we can even set it to 0 to ignore the weight of the horizontal box. \( \text{Loss}_{\text{ROI}} \) is the regression error of the center point and width of the object RoI, following yolo v2 [12].

### Table 3. The meaning of the parameters of Loss function.

| Parameter | \( \lambda_{\text{noobj}} \) | \( \lambda_{\text{obj}} \) | \( \lambda_{\text{ROI}} \) | \( \lambda_{\text{pts}} \) | \( \bar{P} \) | \( \bar{P}^\text{t} \) |
|-----------|----------------|----------------|----------------|----------------|---------|----------------|
| Description | loss weight of non-object | loss weight of object | loss weight of object RoI | loss weight of each point | predicted value of the point \((u, v)\) | ground truth of the point \((u^\text{t}, v^\text{t})\) |

The principle of judging whether there is an object in a local range is to calculate the maximum IoU between the default anchors’ boxes and all the ground truth boxes in that range. If the IoU exceeds a threshold, there will be an object. The principle of discrimination here is consistent with the processing of yolo v2 and yolo v3 [12,13]. Therefore, the sum of the differences in the three-layer APP prediction result and the ground truth is

\[
\text{Loss} = \sum_{\text{output Layer}} \text{Loss (Output Layer)}.
\]

(5)

### 3.2. Training Procedure

Training datasets. We experimented with the DOTA dataset. The original DOTA images are high-resolution remote sensing images, which is not convenient for direct processing using the neural network. Therefore, first of all, the raw data needs to be standardized. The method we took was to randomly select an image point, then center the point, and align the center point \((W/2, H/2)\) of the transformed image \((W, H)\) for random affine transformation. The scale of the transformation is 0.5 to 1.5 times the original image. Then, we obtain the sample images.

Training and testing. At training, we used 80% DOTA images, and all the processed images were resized into 416*416 and sent to the neural network. After training, the remaining 20% of images were used for testing. In the training process, the choice of multiple anchors followed the strategy of yolo v3, and the process of backpropagation of the loss layer was divided into two phases.

Phase 1: Scan each output of the output layer array. According to the ground truth set and the boxes determined by predicted APP coordinates, the output region can be obtained from the maximum IoU between them. If \( \text{IoU}_{\text{max}} \) is less than \( \varepsilon \), the corresponding object presence expectation output value will be set to 0 for backpropagation correction.

Phase 2: Scan the rows and columns of each GTBox, and correct the largest anchor of the IoU between the default rectangle of \( n \) anchors at this position and the GTBox. Then, set the expected value of the object field to 1, and the loss of APP is calculated according to Equation (19), and the expected value of the softmax segment is set to perform backpropagation correction.

Application. According to the four-sided output bounding box surrounded by four APP coordinates, the four points of the inclined box are further obtained by Equation (19), and the point coordinates are converted to the large image according to Equation (5).

### 3.3. Calculation of the Object of 3D Pose

The conventional methods often calculate the 3D pose of the object by matching the local features extracted in the 2D image with the features in the object 3D model to be detected, but these methods are not accurate enough [29]. Therefore, based on the key point coordinates of the object output from the
region layer, we use the perspective transformation method to calculate the 3D pose. Figure 7 shows the computational process of the objects’ 3D pose. We use two methods, PnP [30] and homograph. The following describes these two methods.

### 3.3.1. PnP Method

As shown in Figure 7, we can get the coordinates \( \{ P_1, P_2, P_3, P_4 \} \) of the four feature points of the object in the training part. These feature point coordinates are relative to the coordinates of the complete satellite image. If the image is cropped and resized, these coordinates need to be transformed to the coordinate system of the original satellite image. The inference part obtains the 3D pose of the object based on the correspondence between the four feature points and the body coordinates of the four or eight corner points of the object. Assume that the length and width of the bounding box of the object is \( W \) and \( H \), and the height of the object from the ground is \( H_g \). We define the eight points of the bounding box of the object’s own coordinate system as:

\[
\begin{align*}
&\left\{-\frac{W}{2}, H, \pm \frac{H_g}{2}\right\}, \left\{\frac{W}{2}, -H, \pm \frac{H_g}{2}\right\}, \\
&\left\{-\frac{W}{2}, -H, \pm \frac{H_g}{2}\right\}, \left\{\frac{W}{2}, H, \pm \frac{H_g}{2}\right\}
\end{align*}
\]

Since the distance from the camera to the ground object is much larger than the object’s own height \( H_g \), the image coordinates of the two feature points at different heights of the object at the same latitude and longitude are very close on the image. Therefore, the corresponding relationship between the eight points of the object’s bounding box and the key points \( \{ P_1, P_2, P_3, P_4 \} \) of the image coordinates is:

\[
\begin{align*}
&\left\{\frac{-W}{2}, -H, \pm \frac{H_g}{2}\right\} \rightarrow -P_1 \quad \text{left top} \\
&\left\{\frac{W}{2}, -H, \pm \frac{H_g}{2}\right\} \rightarrow -P_2 \quad \text{right top} \\
&\left\{-\frac{W}{2}, H, \pm \frac{H_g}{2}\right\} \rightarrow -P_3 \quad \text{right bottom} \\
&\left\{\frac{W}{2}, H, \pm \frac{H_g}{2}\right\} \rightarrow -P_4 \quad \text{left bottom}
\end{align*}
\]

According to this correspondence, we call the PnpSolve function in OpenCV to solve the camera external parameters \( R \) and \( T \), where \( R \) is the attitude matrix of the camera relative to the object, and \( T \) is the position of the object relative to the camera.

### 3.3.2. Homograph Method

We define the object as an inclined box with a width \( W \) and a length \( H \). The origin of the object’s coordinate system is defined at the center of the box. Considering the particularity of remote sensing
images, the four vertices of the object in the object coordinate system are similar on a ground plane $\pi$, and the four vertices of the object in the object coordinate system are marked as:

$$
\begin{align*}
\begin{cases}
\left[ \frac{-W}{2}, \frac{-H}{2} \right], & \left[ \frac{W}{2}, \frac{-H}{2} \right] \\
\left[ \frac{W}{2}, \frac{H}{2} \right], & \left[ \frac{-W}{2}, \frac{H}{2} \right]
\end{cases}
\end{align*}
$$

(8)

According to the DOTA data format, the four points are sorted clockwise from the upper left corner. Assuming that the aspect ratio of the object is also unknown, set to $\alpha$, then the four object points can be written as:

$$
\begin{align*}
\begin{cases}
\left[ \frac{-W}{2}, \frac{-\alpha W}{2} \right], & \left[ \frac{W}{2}, \frac{-\alpha W}{2} \right] \\
\left[ \frac{W}{2}, \frac{\alpha W}{2} \right], & \left[ \frac{-W}{2}, \frac{\alpha W}{2} \right]
\end{cases}
\end{align*}
$$

(9)

According to the principle of satellite remote sensing imaging, the line of sight of the imaging camera of the remote sensing image is perpendicular to the ground plane $\pi$, so the conversion of the four points on the object plane to the image plane follows the rotation transformation matrix:

$$
\begin{bmatrix}
u \\
v_1
\end{bmatrix} =
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
-a_{12} & a_{11} & a_{23}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
1
\end{bmatrix}
$$

(10)

Conversely, the conversion of four points on the image plane to the four points on the object plane follows the inverse of Equation (8):

$$
\begin{bmatrix}
X \\
Y
\end{bmatrix} =
\begin{bmatrix}
a'_{11} & a'_{12} & a'_{13} \\
-a'_{12} & a'_{11} & a'_{23}
\end{bmatrix}
\begin{bmatrix}
u \\
v_1
\end{bmatrix}
$$

(11)

According to the basic principle of affine transformation,

$$
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
-a_{12} & a_{11} & a_{23}
\end{bmatrix} =
\begin{bmatrix}
a'_{11} & a'_{12} & a'_{13} \\
-a'_{12} & a'_{11} & a'_{23}
\end{bmatrix}^{-1}
$$

$$
= \frac{1}{a'_{11}^2 + a'_{12}^2}
\begin{bmatrix}
a'_{11} & -a'_{12} & a'_{13} a'_{23} - a'_{12} a'_{11} \\
-a'_{12} & a'_{11} & -(a'_{11} a'_{23} + a'_{13} a'_{12})
\end{bmatrix}
$$

(12)

In order to solve the four parameters $a_{11}'$, $a_{12}'$, $a_{13}'$, $a_{23}'$ of affine transformation, we obtain the formulas of the four parameters and $\alpha$ according to Equation (7):

$$
\begin{bmatrix}
u \\
v
\end{bmatrix}
=\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{23} \\
-a_{12} & a_{11} & a_{23} & a_{13}
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
1 \\
-\alpha
\end{bmatrix}
$$

(13)
The four-point coordinates in Equation (9) are sequentially brought into X and Y in the above equation,

\[
\begin{align*}
[X_1, Y_1] &= [\frac{-W}{2}, \frac{-aW}{2}] \\
[X_2, Y_2] &= [\frac{W}{2}, \frac{-aW}{2}] \\
[X_3, Y_3] &= [\frac{W}{2}, \frac{aW}{2}] \\
[X_4, Y_4] &= [\frac{-W}{2}, \frac{aW}{2}] 
\end{align*}
\]

(14)

Then, we can get

\[
\begin{bmatrix}
    u_1 & v_1 & 1 & 0 & 0 \\
    v_1 & -u_1 & 0 & 1 & -Y_1 \\
    u_2 & v_2 & 1 & 0 & 0 \\
    v_2 & -u_2 & 0 & 1 & -Y_2 \\
    u_3 & v_3 & 1 & 0 & 0 \\
    v_3 & -u_3 & 0 & 1 & -Y_3 \\
    u_4 & v_4 & 1 & 0 & 0 \\
    v_4 & -u_4 & 0 & 1 & -Y_4 \\
\end{bmatrix}
\begin{bmatrix}
    a_{11}' \\
    a_{12}' \\
    a_{13}' \\
    a_23' \\
    \alpha
\end{bmatrix}
= 
\begin{bmatrix}
    X_1 \\
    X_2 \\
    X_3 \\
    X_4 \\
    0
\end{bmatrix}
\]

(15)

After solving Equation (14) by the least two multiplication method, it is brought into Equation (16) to get \(a_{11}', a_{12}', a_{13}', a_{23}'.\) According to the principle of perspective transformation, we can get

\[
K^{-1}H = \begin{bmatrix}
    f & 0 & c_x \\
    0 & f & c_y \\
    0 & 0 & 1
\end{bmatrix}
^{-1}
\begin{bmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{12} & a_{11} & a_{23} \\
    0 & 0 & 1
\end{bmatrix}
= \begin{bmatrix}
    \frac{1}{f} & 0 & \frac{-c_x}{f} \\
    0 & \frac{1}{f} & \frac{-c_y}{f} \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{12} & a_{11} & a_{23} \\
    0 & 0 & 1
\end{bmatrix}
= \begin{bmatrix}
    \frac{a_{11}}{f} & \frac{a_{12}}{f} & \frac{a_{13}-c_x}{f} \\
    \frac{-a_{12}}{f} & \frac{-a_{11}}{f} & \frac{a_{23}-c_y}{f} \\
    0 & 0 & 1
\end{bmatrix}
\]

(16)

Columns 1, 2 of \(K^{-1}H\) are then unitized to obtain the matrix:

\[
[K^{-1}H]_1 = \frac{f}{\sqrt{a_{11}^2 + a_{12}^2}}
\begin{bmatrix}
    \frac{a_{11}}{f} & \frac{a_{12}}{f} & \frac{a_{13}-c_x}{f} \\
    \frac{-a_{12}}{f} & \frac{-a_{11}}{f} & \frac{a_{23}-c_y}{f} \\
    0 & 0 & 1
\end{bmatrix}
\]

(17)

The attitude matrix \(R = [c_1, c_2, c_1 \times c_2].\) The columns 1 and 2 of \([K^{-1}H]_1\) construct the columns 1, 2 \(c_1, c_2\) of the matrix R. The third column of \([K^{-1}H]_1\) is the offset \(T\) of the object relative to the camera.

Based on the above, we summarize the calculation process of the 3D pose for remote sensing image objects.

1) Predict the APP coordinates of each object through the neural network.
2) According to Equation (15), we can get the inverse affine transformation parameters \(a_{11}', a_{12}', a_{13}', a_{23}'\) and the object width-to-length ratio \(\alpha.\)
3) According to Equation (12), we can get the affine transformation matrix from the object coordinate system to the image coordinate system:

\[
A = \begin{bmatrix}
    a_{11} & a_{12} & a_{13} \\
    -a_{12} & a_{11} & a_{23}
\end{bmatrix}
\]

(18)

4) Matrix \([K^{-1}H]_1\) is obtained from Equation (17).
5) Attitude matrix $R$ and displacement $T$ can be obtained by decomposing matrix $[K^{-1}H]_1$.

3.3.3. Object Spatial Location Using Remote Sensing Image

As shown in Figure 8, the geometric transformation of the satellite relative to the earth $\begin{bmatrix} R_s & t_s \\ 1 \end{bmatrix}$ can be obtained accurately; the rigid body connection transformation $\begin{bmatrix} R_c & t_c \\ 1 \end{bmatrix}$ of the satellite camera relative to the satellite can also be measured. Then, through the method based on APP, the transformation of the object relative to the camera can be obtained as $\begin{bmatrix} R^T & T \\ 1 \end{bmatrix}$. To sum up, it can be calculated that the 3D transform of the object relative to the earth is converted into $\begin{bmatrix} R^T & T \\ 1 \end{bmatrix} \begin{bmatrix} R_c & t_c \\ 1 \end{bmatrix} \begin{bmatrix} R_s & t_s \\ 1 \end{bmatrix}$.

![Figure 8. Application of remote sensing image for object spatial location.](image)

4. Experiments and Analysis

4.1. Experimental Details

Experimental data. We conducted experiments on the DOTA and HRSC2016 datasets. For the DOTA dataset, we cut the image into subgraphs with a resolution of $1024 \times 1024$. At training, we used batch-size = 64, and the learning rate was 0.0001 in the first 10,000 training sessions, and after every 20,000 increments, the learning rate was reduced by 0.1 times until the learning rate was equal to 0.000001. Our full convolutional network supports the input of images with different resolutions. We tested two different resolution inputs for $416 \times 416$ and $1024 \times 1024$. The $416 \times 416$ model can get an output of $13 \times 13, 26 \times 26, 52 \times 52$ scales, and the $1024 \times 1024$ model can get an output of $32 \times 32, 64 \times 64, 128 \times 128$ scales, which correspond to three different object types of large, medium, and small scales, respectively. In the detecting, we first divided the DOTA image into blocks. The blocks needed to overlap to avoid cross-border loss of the object. Finally, we calculated and counted the mAP of the object. mAP is the mean Average Precision, which means the average of the AP of all object categories. Tables 4 and 5 are the statistics on the HRSC2016 and DOTA data sets, respectively. As can be seen from these two tables, the APP algorithm has the highest mAP.

Table 4. Comparisons with the state-of-the-art methods on HRSC2016 [8].

| Method     | CP[18] | BL2[14] | RC1[19] | RC2[19] | R^2CNN[26] | RRD[30] | Rel Trans[8] | APP     |
|------------|--------|---------|---------|---------|------------|---------|---------------|---------|
| mAP        | 55.7   | 60.6    | 75.7    | 75.7    | 84.3       | 66.2    | 86.3          |         |
Table 5. Comparisons with state-of-the-art detectors on DOTA. There are 15 categories, including Baseball diamond (BD), Ground track field (GTF), Small vehicle (SV), Large vehicle (LV), Tennis court (TC), Basketball court (BC), Stor-age tank (ST), Soccer-ball field (SBF), Roundabout (RA), Swimming pool (SP), and Helicopter (HC) [8].

| Method   | Plane | BD | Bridge | GTF | SV | LV | Ship | TC | BC | ST | SBF | RA | Harbor | SP | HC | mAP  |
|----------|-------|----|--------|-----|----|----|------|----|----|----|-----|----|--------|----|----|------|
| FR-O [33] | 79.42 | 77.13 | 17.7 | 64.05 | 35.3 | 30.02 | 37.16 | 89.41 | 69.64 | 59.28 | 50.3 | 52.91 | 47.89 | 47.4 | 46.3 | 54.13 |
| RRPN [34] | 80.94 | 65.75 | 50.34 | 67.44 | 59.65 | 50.91 | 50.81 | 90.67 | 66.92 | 72.39 | 55.06 | 52.25 | 55.14 | 52.35 | 49.22 | 60.67 |
| R2CNN [35] | 88.52 | 71.2 | 31.66 | 59.3 | 51.85 | 56.19 | 57.25 | 90.91 | 72.84 | 67.38 | 56.69 | 52.84 | 53.08 | 51.94 | 53.58 | 61.01 |
| DPSRP [8] | 81.18 | 77.42 | 35.48 | 72.41 | 56.74 | 50.42 | 53.56 | 89.97 | 79.68 | 76.48 | 61.59 | 59.64 | 53.34 | 64.04 | 47.78 | 63.90 |
| Roll Trans. [8] | 88.53 | 77.91 | 37.63 | 76.08 | 66.53 | 62.97 | 66.57 | 90.46 | 76.75 | 59.04 | 56.73 | 62.54 | 61.29 | 55.56 | 67.74 |
| RoI Trans. [8] | 87.98 | 75.38 | 43.93 | 65.10 | 68.95 | 77.04 | 87.63 | 81.59 | 78.96 | 76.54 | 57.20 | 63.95 | 62.22 | 49.01 | 68.72 |
| PnP | 89.06 | 78.23 | 43.52 | 76.39 | 68.42 | 71.62 | 79.05 | 90.42 | 81.51 | 59.48 | 58.91 | 47.24 | 62.19 | 0.46 | 70.13 |

We also tested five sets of models with different parameters. According to Equation (8), we could get the IoU between the predicted object point set and the true value point set, and calculate the corresponding mAP of each of these models. Tables 6 and 7 are model parameters and experimental results, respectively. By comparing the experimental results, we found that the mAP of the first set of model parameters was the highest.

Table 6. Models with different parameters. We used five groups of loss weights to get different parameter models and compared their recognition accuracy.

| Number | $\lambda_{noobj}$ | $\lambda_{obj}$ | $\lambda_{ROI}$ | $\lambda_{pts}$ | $\lambda_{softmax}$ |
|--------|-------------------|-----------------|-----------------|-----------------|-------------------|
| 1      | 1                 | 2               | 0.0001          | 1               | 1                 |
| 2      | 1                 | 3               | 0.0001          | 1               | 1                 |
| 3      | 1                 | 5               | 0.0001          | 2               | 1                 |
| 4      | 1                 | 2               | 0.0001          | 1.5             | 1                 |
| 5      | 1                 | 2               | 0.1             | 1               | 1                 |

Table 7. mAP for each model. It can be seen from the table that the model obtained by row 1 has the highest overall accuracy.

| Method   | Plane | BD | Bridge | GTF | SV | LV | Ship | TC | BC | ST | SBF | RA | Harbor | SP | HC | mAP  |
|----------|-------|----|--------|-----|----|----|------|----|----|----|-----|----|--------|----|----|------|
| 1 89.06 | 78.23 | 43.52 | 76.39 | 68.42 | 71.62 | 79.05 | 90.42 | 81.51 | 80.51 | 59.48 | 58.91 | 47.24 | 62.19 | 0.46 | 70.13 |
| 2 87.98 | 75.38 | 43.93 | 65.10 | 68.95 | 77.04 | 87.63 | 81.59 | 78.96 | 76.54 | 57.20 | 63.95 | 62.22 | 49.01 | 68.72 |
| 3 85.97 | 77.11 | 49.01 | 71.26 | 65.10 | 68.95 | 77.04 | 87.63 | 81.59 | 78.96 | 57.20 | 63.95 | 62.22 | 49.01 |
| APP     | 89.06 | 78.23 | 43.52 | 76.39 | 68.42 | 71.62 | 79.05 | 90.42 | 81.51 | 59.48 | 58.91 | 47.24 | 62.19 |

The efficiency of the algorithm is one of the most important indicators for measuring the quality of the algorithm. Our model can input images in two different resolutions. As shown in Table 8, we used a one-stage process including NMS operation, so it was more efficient. Table 9 is the mean length of the objects. Assuming that the width W of all objects is equal to 1, the average length of each type of object also can be solved.

Table 8. Speed comparison with other methods (unit: ms). The LR-O, RoI Trans, and DPSRP denote RoI Transformer and the Light-Head R-CNN OBB, deformable Position Sensitive RoI pooling, respectively [8].

| Image Size | LR-O | RoI Trans. | DPSRP | APP |
|------------|------|------------|-------|-----|
| 416 x 416  | -    | -          | -     | 13.5|
| 1024 x 1024| 141  | 170        | 206   | 140 |

Table 9. Mean length of the objects.

| Object   | Plane | BD | Bridge | GTF | SV | LV | Ship | TC | BC | ST | SBF | RA | Harbor | SP | HC | avg |
|----------|-------|----|--------|-----|----|----|------|----|----|----|-----|----|--------|----|----|-----|
| length   | 1.0064 | 1.0169 | 1.7342 | 1.4534 | 2.1273 | 1.8274 | 2.0055 | 1.9401 | 1.6262 | 1.0374 | 1.3142 | 1.0276 | 4.4175 | 1.9194 | 2.9527 |
| 1.9863 |
Experimental results. We improved yolo v3 and added four corner points of the APP prediction object based on the original region layer. The local object was predicted according to Equation (19), and then the coordinates were converted to the large image to obtain the homograph transform. Further, we decomposed the three-dimensional attitude $R$ and displacement $T$ of the object relative to the camera. Figure 9 is the result of the experiment. Figure 10 is an incorrectly labeled image that we can detect correctly after training.

Figure 9. 3D pose renderings. The red arrow direction and the green arrow direction in the Figure are the X-axis and Y-axis of the object itself, respectively.
Figure 10. The inner box is not accurate in the DOTA1.0 training dataset, but we get a more accurate box by prediction.

For large images such as DOTA, we used the method of block synthesis. The large image is divided into a number of sub-blocks, each of which is just the standard size of the neural network input layer $[S_w, S_h]$, and the neural network was used to detect the object APP coordinates $(u_i, v_i)$ in each sub-block range. Assuming that the coordinate of the upper-left corner of a sub-block is $(\text{left}, \text{top})$, the coordinate of the point in the full image is

$$
\begin{align*}
  u_{\text{big}} &= \text{left} + u_i \\
  v_{\text{big}} &= \text{top} + v_i
\end{align*}
$$

(19)

Converting all the objects of the sub-image to the large image, and in finally performing the NMS operation, in order to avoid the object being cut by the block, overlap between blocks was needed, and the overlap length was longer than the minimum object length. The overlap length of the horizontal and vertical sides was initially set at 20% of the basic length, as shown in Figure 11.

Figure 11. Large image segmentation.

4.2. Error Analysis

We used the method of projecting pixel points to comprehensively evaluate the prediction accuracy of the 3D pose. The process is shown in Figure 7. When evaluating the accuracy of the 3D pose, the position and attitude of the object in three-dimensional space could be considered comprehensively by using the method of pixel projection error [36]. According to the nature of the attitude matrix $R$, the three rows of the matrix represent the three unit vectors of the x, y, and z axis of the camera coordinate system relative to the object, and then the three rows of $R$, in turn, represent the x, y, and z axis relative to the three unit vectors of the camera coordinate system. Since the y-axis of the object ontology coordinate system is the negative direction of the object orientation, the negative vector $-r_{12}, -r_{22}$ of the second column of the matrix $R$ describes the direction of the object. Therefore, the azimuth angle of the object relative to the camera coordinate system $\theta = \text{atan} (r_{12}, r_{22})$. The coordinate calculation Equation is

$$
\begin{bmatrix}
  u_i \\
  v_i \\
  1
\end{bmatrix} = KR (X_i - T) \quad i = 1, 2, 3, 4,
$$

(20)
where $K$ is the internal parameter matrix of the camera, $R$ and $T$ are the attitude matrix and the
 displacement matrix obtained according to the algorithm of Section 4.1, and $X_i$ is the coordinates of
the four projection points of the object box close to the ground. The pixel error is

$$e^2 = \left| \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} - \begin{bmatrix} \bar{u}_i \\ \bar{v}_i \\ 1 \end{bmatrix} \right|^2. \quad (21)$$

According to this formula, we could get the angular error and pixel error of the predicted object. We
tested five different sets of parameters to get different models. It can be seen from Table 10 that the
error of the first set of parameters is the smallest.

| Number | $\lambda_{\text{noobj}}$ | $\lambda_{\text{obj}}$ | $\lambda_{\text{ROI}}$ | $\lambda_{\text{pts}}$ | $\lambda_{\text{softmax}}$ | Average Angle Error | Global projection pixel error |
|--------|----------------|----------------|----------------|---------------|----------------|----------------|------------------|
| 1      | 1              | 2              | 0.0001         | 1             | 1              | 4.28           | 2.03             |
| 2      | 1              | 3              | 0.0001         | 1             | 1              | 4.32           | 2.05             |
| 3      | 1              | 5              | 0.0001         | 2             | 1              | 4.85           | 2.07             |
| 4      | 1              | 2              | 0.0001         | 1.5           | 1              | 4.93           | 2.16             |
| 5      | 1              | 2              | 0.1            | 1             | 1              | 4.78           | 2.37             |

5. Conclusion

In this paper, we proposed a new Anchor Points Prediction algorithm that can accurately
determine the position and attitude of the object’s three-dimensional space. Differing from the
traditional methods of predicting object RoI or inclined box, we used the neural network to predict
multiple feature points to detect the objects. This algorithm is a one-stage algorithm, and its accuracy
and efficiency have been greatly improved. It not only uniquely determines the direction of the object,
but also calculates the 3D pose of the object from the APP coordinates. We believe that the APP
algorithm can be better applied to object detection. Moreover, the point prediction algorithm has
broad application prospects and may become a new trend in the future. Our method also has some
shortcomings. For a slender object like Harbor, the bounding box is relatively large, and the object
occupies only a small part of the bounding box. In this way, the features in the extracted ROI region
will be inaccurate, leading to a decrease in the accuracy of predicting key points of the object.

Author Contributions: conceptualization, J. L.; methodology, J. L.; software, J. L.; validation, J. L.; formal analysis,
J. L. and Y. G.; writing–original draft preparation, J. L.; writing–review and editing, J. L.; visualization, J. L.;
supervision, J. L.; project administration, J. L.

Funding: This research was funded by National Natural Science Foundation of China under Grant No. 41771457.

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

Abbreviations

The following abbreviations are used in this manuscript:

- **MDPI**: Multidisciplinary Digital Publishing Institute
- **DOAJ**: Directory of open access journals
- **TLA**: Three letter acronym
- **LD**: linear dichroism

References

1. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* 2017, 55, 2486–2498. doi:10.1109/TGRS.2016.2645610.
2. Wang, G.; Wang, X.; Fan, B.; Pan, C. Feature Extraction by Rotation-Invariant Matrix Representation for Object Detection in Aerial Image. *IEEE Geosci. Remote Sens. Lett.* 2017, 14, 851–855. doi:10.1109/LGRS.2017.2683495.

3. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Zou, H. Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled Region-Based Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* 2017, 10, 3652–3664. doi:10.1109/JSTARS.2017.2694890.

4. Girshick, R. Fast R-CNN. In Proceedings of The IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015.

5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks; Curran Associates, Inc.: Montreal, Canada, 2015; pp. 91–99.

6. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake, Utah, USA, 18–22 June 2018.

7. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, Columbus, USA, 24–27 June 2014.

8. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning Rol Transformer for Oriented Object Detection in Aerial Images. In Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Angeles, California, USA, 16–19 June 2019.

9. Ren, H.; El-Khamy, M.; Lee, J. CT-SRCNN: Cascade Trained and Trimmed Deep Convolutional Neural Networks for Image Super Resolution. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, Sierra Nevada, USA, 12–15 March 2018; pp. 1423–1431. doi:10.1109/WACV.2018.00160.

10. Lin, T.Y.; Dollar, 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 (CVPR), Honolulu, Hawaii, USA, 21–26 July 2017.

11. Etten, A.V. You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery, Cornell University: Ithaca, New York, USA, 2018. [arXiv:1805.09512].

12. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, 21–26 July 2017.

13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement, Cornell University: Ithaca, New York, USA, 2018. [arXiv:1804.02767].

14. Liu, Z.; Wang, H.; Cheng, Y.; Wang, W.; Yang, Y. Ship Rotated Bounding Box Space for Ship Extraction From High-Resolution Optical Satellite Images With Complex Backgrounds. *IEEE Geosci. Remote Sens. Lett.* 2016, 13, 1074–1078. doi:10.1109/LGRS.2016.2565705.

15. Liu, Z.; Hu, J.; Deng, S.; Yang, Y. Rotated region based CNN for ship detection. In Proceedings of The 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 18–20 September 2017; pp. 900–904. doi:10.1109/ICIP.2017.8296411.

16. Liu, K.; Mattys, G. Fast Multiclass Vehicle Detection on Aerial Images. *IEEE Geosci. Remote Sens. Lett.* 2015, 12, 1938–1942. doi:10.1109/LGRS.2015.2439517.

17. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of The European Conference on Computer Vision (ECCV), Munich, Germany, 08–14 September 2018.

18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F.; Burges, C.J.C.; Bottou, L.; Weinberger, K.Q., Eds.; Curran Associates, Inc.: Lake Tahoe, Sierra Nevada, USA, 2012; pp. 1097–1105.

19. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition, Cornell University: Ithaca, New York, USA, 2014. [arXiv:1409.1556].

20. 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 (CVPR), Las Vegas, Nevada, USA, 26 June–1 July 2016.

21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of The European Conference on Computer Vision, Amsterdam, Netherlands, 8–16 October 2016.
22. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points, Cornell University: Ithaca, New York, USA 2019, [arXiv:1904.07850].
23. Rad, M.; Lepetit, V. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In Processings of The IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
24. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Processings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, USA, 26 June–1 July 2016.
25. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Proce. Lett. 2016, 23, 1499–1503. doi:10.1109/LSP.2016.2603342.
26. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In Processings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, 21–26 July 2017.
27. Neubeck, A.; Van Gool, L. Efficient Non-Maximum Suppression. In Processings of the 18th International Conference on Pattern Recognition (ICPR’06), Hong Kong, China, 20–24 August 2006; pp. 850–855. doi:10.1109/ICPR.2006.479.
28. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. Int. J. Comput. Vision 2010, 88, 303–338. doi:10.1007/s11263-009-0275-4.
29. Li, Y.; Wang, G.; Ji, X.; Xiang, Y.; Fox, D. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In Processings of The European Conference on Computer Vision (ECCV), Munich, Germany, 08–14 September 2018.
30. Liu, J.; He, S. 6D Object Pose Estimation without PnP, Cornell University: Ithaca, New York, USA, 2019. [arXiv:1902.01728].
31. Zhang, Z.; Guo, W.; Zhu, S.; Yu, W. Toward Arbitrary-Oriented Ship Detection With Rotated Region Proposal and Discrimination Networks. IEEE Geosci. Remote Sens. Lett. 2018, 15, 1745–1749. doi:10.1109/LGRS.2018.2856921.
32. Xia, G.S.; Bai, X.; Shi, B.; Xia, G.s.; Bai, X. Rotation-Sensitive Regression for Oriented Scene Text Detection. In Processings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake, Utah, USA, 18–22 June 2018.
33. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. IEEE Trans. Multimedia 2018, 20, 3111–3122. doi:10.1109/TMM.2018.2818020.
34. Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; Navab, N. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In Processings of The IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.