Pose Detection of Aerial Image Object Based on Constrained Neural Network

HONGYUN ZHANG, JIN LIU, AND YONGJIAN GAO
State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

Corresponding author: Jin Liu (jliu@sgg.whu.edu.cn)

ABSTRACT Constraints often exist in the high-dimensional data output in object detection, such as the inverse vector \(\{\cos \theta, \sin \theta\}\) of the two-dimensional object and the attitude quaternion of the three-dimensional object. The range of each component of the output value of the traditional neural network is unconstrained, which is difficult to meet the needs of practical problems. To solve this problem, this paper designed the transformation network layer according to the high dimensional space transformation theory and constructed a constrained neural network model to detect the pose of objects from a single aerial image. Firstly, in the yolov3 network structure, according to the size of the three field scales, three scale transformation network layers are added correspondingly to implement the constrained unit quaternion field. Secondly, according to the characteristics of quaternion, we proposed a special loss function. Then a new constrained neural network called the quaternion field pose network (qfiled PoseNet) model is constructed, which can predict the probability field of the object and the contained unit quaternion field respectively. Next, the object’s probability field is generated to determine the 2D bounding box of the object, and the unit quaternion field is generated to determine the 3D rotation \(R\). Finally, combining the rotation matrix \(R\) and the 2D bounding box of the object to calculate the 3D translation \(T\). We used our method to experiment on DOTA1.5 data set and the HSRC2016 data set respectively. The experimental results show that our method can detect the pose of the object well.

INDEX TERMS Constrained neural network, unit quaternion, 3D rotation, 3D translation.

I. INTRODUCTION

Pose detection and recognition of objects is a technical method to identify all kinds of interested objects in complex environment images and obtain the 3D relative pose relationship between the object and the camera. It is one of the core technologies of the next generation intelligent robot, battlefield object reconnaissance, autonomous navigation, and satellite image interpretation. Its breakthrough will greatly promote the development of relevant research fields [1].

Object pose detection includes 3D translation and 3D rotation. According to the current work of most researchers, pose detection methods are mainly divided into three types: keypoint matching method, RGB-D method, and deep learning method. Among them, the keypoint matching method mainly establishes 2D-3D correspondence on local features [2] and then uses the PnP algorithm to estimate the object pose parameters. However, this kind of method can achieve high accuracy for objects with rich texture, but it is difficult to solve objects without texture [3]. The RGB-D method has also developed greatly with the emergence of depth cameras. Hintertosser et al. [4] proposed surface template matching of 3D point clouds, and then proposed point pair features and several variants [5] to improve robustness to complex background and noise. However, the RGB-D methods require high costs.

In recent years, the deep learning method has become the mainstream for solving the pose of objects, including camera pose and object pose. References [6] and [7] both regress camera pose by training CNN (Convolutional Neural Networks). In reference [6], the authors use quaternion as 3D rotation but omit the spherical constraint, using an unreasonable loss function. References [7] address the problem by developing a more scientific loss function. In reference [8], [9], the authors use CNN to regress 3D rotation, their works focus on 3D rotation estimation while 3D translation is not included. In reference [10], SSD (Single
Shot multi-box Detector) [11] algorithm was improved for object pose detection. The authors transform pose detection into two stages, view angle classification and in-plane rotation classification. However, the classification accuracy of each stage would affect the total accuracy of object pose detection. Algorithm BB8 [12] first uses the segmentation network to locate the object, then uses another CNN to predict the 2D projection of the 3D bounding box corners of the object, and finally uses the PnP algorithm to estimate the pose parameters of the object. This algorithm achieves high precision but takes a long time. Similar to the algorithm BB8, reference [13] improves YOLO (You Only Look Once) object detection framework [14] to predict the 2D projection of the 3D bounding box, then uses the PnP algorithm to estimate the pose parameters of the object. References [12] and [13] both regression too many 2D projection points, which increases the difficulty of model learning and reduces the learning speed. Although the object attitude detection method based on neural network has achieved some research results, there are still some problems.

Each output element of traditional neural network is independent, and the original output range of a single neuron is \((-\infty, +\infty)\). At present, few studies consider the problem of multiple implicit constraints of neural networks. For example, the output neurons \(o_1, \ldots, o_n\) of a neural network computing layer need to meet a multivariate stealth constraint \(\phi (o_1, \ldots, o_n) = 0\). This situation is very common in many practical problems. For example, there is unit orthogonal constraint between 9 elements in the 3D rotation matrix \(R\), the quaternion describing the object’s 3D rotation has the constraint that the sum of squares of the output values is equal to 1, the vector \((\cos \theta, \sin \theta)\) describing the ground projection direction of aerial object has the constraint that the sum of squares is equal to 1. According to Lie algebra theory, \(n\) implicit constraints are equivalent to establishing an \(N-n\) dimensional manifold in the \(N\)-dimensional neuron output space. Therefore, it is possible to use the mature methods and conclusions in Lie algebra to study how to solve the constraint problem in neural networks and establish the most effective pose feature expression.

According to the Lie algebra theory [15], this paper designs the transformation network layer and constructs a constrained neural network model to detect the pose of object in aerial images. Firstly, based on the yolo3 model, the transformation network layer is correspondingly added to the three-branch structure of yolo3 output to predict the quaternion. Secondly, according to the characteristics of quaternion, we propose a special loss function to construct the qfiled-PoseNet model. The model can predict the object’s probability field and the unit quaternion field respectively. Next, the object’s probability field is generated to determine the 2D bounding box of the object, and the unit quaternion field is generated to determine the 3D rotation. Finally, according to the rotation matrix \(R\) and 2D bounding box of object to calculate the 3D translation \(T\). We verified the feasibility of the method on the DOTA1.5 data set and the HSRC2016 data set. Experimental results show that our method can achieve satisfactory results. Our work has the following advantages and contributions:

1) There are no constraints between the output components of the traditional neural network. This paper innovatively proposes a network that can meet the constraints of each output component.

2) In this paper, the transformation network layer is designed to construct a constrained neural network model, which can output the unit quaternion field used to construct the rotation matrix \(R\).

3) For the network whose output is required to be a unit vector, a special loss function is proposed to solve the problem of 3D rotation regression.

The remainder of the paper is organized as follows. After the overview of related work, we introduce the proposed object pose detection method. Then we display the experimental results, followed by the final conclusion.

II. PROPOSED ALGORITHM

A. QFILED-POSENET MODEL

Quaternion is a compact and effective vector describing 3D rotation [16]. It was discovered by the famous mathematician Hamilton and has been widely used in the field of computer vision. We use \(q = \{q_0, q_1, q_2, q_3\}\) to represent the unit quaternion describing the 3D rotation of the object. The conversion relationship between the unit quaternion and the rotation matrix \(R\) can be expressed as [17] (1), as shown at the bottom of the next page.

The line of sight of remote sensing objects is vertically downward, so the rotation matrix \(R\) of the camera is defined as,

\[
R = \begin{bmatrix}
\cos \theta & \sin \theta \\
-\sin \theta \cos \theta & \cos \theta \\
0 & 0 & 1
\end{bmatrix}
\] (2)

We first label the coordinate data of four points (1-4) of an object, and \(\theta\) is the unit vector formed from the center point of the object to the midpoint of points 1-2, as shown in FIGURE 1. Calculate \(\sin \theta\) and \(\cos \theta\) respectively, and then calculate the attitude quaternion \(q\) according to the attitude quaternion calculation formula (3).

During training, the ground truth data comes from the four key points of the object on the aerial image. The coordinates of four points of the object in the data set are converted into quaternions as follows,

\[
q = \begin{bmatrix}
\cos \frac{\theta}{2} \\
0 \\
0 \\
\sin \frac{\theta}{2}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\sqrt{\frac{1+\cos \theta}{2}} \\
0 \\
0 \\
\sqrt{\frac{1-\cos \theta}{2}}
\end{bmatrix}
\] (3)

During prediction, the \(R\) matrix is easily obtained from the predicted \(q\),

\[
R = \begin{bmatrix}
q_0^2 - q_3^2 + 2q_2q_3 & 2q_0q_3 & 2q_0q_2 - q_3^2 & q_0^2 - q_2^2 \\
-2q_0q_3 & q_0^2 - q_3^2 - 2q_2q_3 & 2q_0q_2 + q_3^2 & -2q_0q_2 \\
2q_0q_2 + q_3^2 & -2q_0q_3 & q_2^2 - q_3^2 - 2q_0q_3 & 2q_0q_3 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (4)
The value range of convolution kernel is unlimited, and the output value range weighted by neighborhood of convolution network is also unlimited $[-\infty, \infty]$. The convolution kernel components between different output channels of convolution network are completely independent, so the output components of different channels are also independent. Therefore, the prediction component of neural network is difficult to meet the mutual constraints between different components. For example, the quaternion describing the object attitude has the constraint that the sum of squares is equal to 1, but the traditional neural network is difficult to meet this condition. Therefore, for the learning problem, the point product method is used to define the loss function.

Let the quaternion of the rotation obtained through the transformation function be defined as equation (5), so that the transformed $q = \{q_0, q_1, q_2, q_3\}$ satisfies the constraint $\phi(q) = 0, \phi(q)$ as shown in equation (6),

$$
\begin{align*}
q_1 &= f_1(Q_1, \ldots, Q_n) \\
\vdots \\
q_n &= f_n(Q_1, \ldots, Q_n) \\
\phi(q) &= q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1 
\end{align*}
$$

According to the chain derivation formula rule, (8) is obtained from equation (7). Therefore, solving the transformation function problem becomes solving the system of linear differential equations (8).

$$
\begin{align*}
3 \sum_{j=0}^{3} \frac{\partial \phi}{\partial q_j} \frac{\partial q_j}{\partial Q_l} &= 0, i = 0, \ldots, 3 \\
\begin{bmatrix}
\frac{\partial \phi}{\partial q_1} & \frac{\partial \phi}{\partial q_2} & \frac{\partial \phi}{\partial q_3} \\
\frac{\partial \phi}{\partial q_4} & \frac{\partial \phi}{\partial q_5} & \frac{\partial \phi}{\partial q_6} \\
\frac{\partial \phi}{\partial q_7} & \frac{\partial \phi}{\partial q_8} & \frac{\partial \phi}{\partial q_9}
\end{bmatrix} = 0 
\end{align*}
$$

Therefore, so like [18], we design the transformation function equation (9), which is exactly a set of solutions of the differential equation (8) and satisfies the constraint condition equation (6). Then the forward propagation formula of the neural network conversion layer is,

$$
q_i = \frac{Q_i}{L}, L = \sqrt{Q_0^2 + Q_1^2 + Q_2^2 + Q_3^2}
$$

where, $i = 0, 1, 2, 3$. Therefore, a unitized constraint layer is added to regress the rotation. In [18], the authors have proven that the normalization layer helps improve the accuracy of prediction.

Let the quaternion of the rotation obtained through the output of the unitized constraint layer be expressed as $q = \{q_0, q_1, q_2, q_3\}$, and meet the condition $\phi(q) = 0$. The schematic diagram of the unitized constraint layer is shown in FIGURE 2.

The constraint proposed in this paper constrains each output component, which is independent of the ground truth and needs to be strictly satisfied $\phi(q) = 0$. It is a prerequisite for training the network model.

Quaternions are hemispherically distributed in 4D space, which makes $q_i$ and $-q_i$ represent the same rotation. However, in the output of neural network, the distance between $q_i$ and $-q_i$ is the farthest. In order to solve this problem, the point product method is used to define the loss

$$
R = \begin{bmatrix}
q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\
2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\
2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 
\end{bmatrix}
$$
The loss function $E_q$ is expressed as

$$\text{loss}_q = E_q = 1 - (q \cdot \bar{q})^2 \quad (10)$$

where $\bar{q} = \{\bar{q}_0, \bar{q}_1, \bar{q}_2, \bar{q}_3\}$ is the true value of the corresponding component of the quaternion.

After the loss function $E_q$ is defined, the back propagation formula is expressed as [19],

$$\frac{\partial E_q}{\partial Q_i} = \sum_{j=0}^{3} \frac{\partial E_q}{\partial q_i} \times \frac{\partial q_i}{\partial Q_i} = -2 (q \cdot \bar{q}) \begin{bmatrix} \frac{\partial q_0}{\partial Q_i} \\ \frac{\partial q_1}{\partial Q_i} \\ \frac{\partial q_2}{\partial Q_i} \\ \frac{\partial q_3}{\partial Q_i} \end{bmatrix} \frac{\partial \bar{q}_0}{\partial q_i} \begin{bmatrix} \bar{q}_0 \\ \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \end{bmatrix}$$

$$= -2 (q \cdot \bar{q}) \begin{bmatrix} \frac{\partial q_0}{\partial Q_i} \\ \frac{\partial q_1}{\partial Q_i} \\ \frac{\partial q_2}{\partial Q_i} \\ \frac{\partial q_3}{\partial Q_i} \end{bmatrix} \begin{bmatrix} \bar{q}_0 \\ \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \end{bmatrix} \quad (11)$$

where,

$$\begin{align*}
\frac{\partial q_i}{\partial Q_i} &= \frac{1}{L} \left( 1 - \frac{Q_i^2}{L^2} \right) = \frac{1}{L} \left( 1 - q_i^2 \right) i = j \\
\frac{\partial q_i}{\partial Q_i} &= -\frac{Q_i}{L^3} = -\frac{q_i q_j}{L} i \neq j
\end{align*} \quad (12)$$

We use formulas (11) and (12) to realize the gradient transfer of the unitization operator in back propagation.

In order to regress the 3D rotation of multiple objects in the scene, it is necessary to expand the scalar value into a field for calculation. We generalize the four input scalar values $\{Q_0, Q_1, Q_2, Q_3\}$ into four input feature maps and the four output scalar values $\{q_0, q_1, q_2, q_3\}$ into four output feature maps, so as to construct a quaternion output model supporting full convolution operation. The structure is shown in FIGURE 3.

The network transformation layer defined by us for regression unit quaternion field is added to three output branches of yolov3 with different sizes to construct the qfield-PoseNet model. Yolov3 is a 106 layer structure. It outputs three groups of tensors in 82, 94, and 106 layers respectively, and obtains three object sets with different scales. We extended 24 layers to form a 130 layer network structure. At layers 75, 87, and 99, this network predicts 2D horizontal boxes of objects with different scales. And then, in darknet, cross-connect and jump are realized through the route layer. The 114, 122, and 130 layer outputs the corresponding object’s unit quaternion fields of large, medium, and small scales. This ensures that each output object’s 2D bounding box corresponds to a quaternion field. The network structure has changed from the original 3 branch structure to the current $3 \times 2$ branch structure. The network structure is shown in FIGURE 4.

FIGURE 4 describes the topological connection relationship of the network.

The total loss function of the qfield-PoseNet model can be expressed as,

$$\text{loss} = \text{loss}_{\text{yolov3}} + \text{loss}_q$$

$$= \text{loss}_{\text{yolov3}} + \sum_{W \times H} 1^{\text{obj}}_i (E_q) \quad (13)$$

where $\text{loss}_{\text{yolov3}}$ is the loss function of yolov3 model, $M$, $H$ represents the size of the tensor of the quaternion field, and $1^{\text{obj}}_i$ denotes if object appears in cell $i$.

The difficulty of object detection method based on neural network is that neural network is more suitable for learning mutually independent dense base spatial data, but not for learning constrained sparse manifold data. Therefore, we summarize the general methods of learning constrained sparse spatial data as follows:

(1) In the training stage, the sparse data $R$ in high-dimensional space is transformed and compressed into dense spatial data $Q$.

(2) In the reasoning stage, the unconstrained output $Q$ of the neural network is transformed into the original data $R$ through the inverse transformation in the training stage.

B. CALCULATE AERIAL IMAGE OBJECT TRANSLATION

We calculate the translation $T$ of the object according to the Bounding Box Equation technology proposed in the reference [19].
According to the matrix equation,
\[ AT = X_{\text{box}} \]  
(14)

where,
\[
A = \begin{bmatrix}
1 & 0 & -\frac{x_L}{f} & -c_x \\
0 & 1 & -\frac{x_R}{f} & -c_x \\
0 & 1 & -\frac{y_T}{f} & -c_y \\
0 & 1 & -\frac{y_B}{f} & -c_y
\end{bmatrix}
\]  
(15)

c is the main point of the image, \( f \) is the focal length, \([x_L, y_T, x_R, y_B]\) is the range of the object rectangle in the image (16), as shown at the bottom of the next page.

The 3D translation \( T \) is obtained by the least square method, expressed as,
\[
T = (AA^T)^{-1} A^T X_{\text{box}}
\]  
(17)

III. EXPERIMENT
In order to speed up the experiment, we use two computers at the same time. The configurations of the two computers are: i7-8700cpu @ 3.19ghz, 32GB memory, 1080ti graphics card, windows10 system for hrsc 2016 data set, and i7-8700cpu @ 3.19ghz, 32GB memory, 2080ti graphics card, windows10 system for dota 1.5 data set. The development environment is vs2013 + opencv2.4.10. DOTA 1.5 data set (https://captain-whu.github.io/DOAI2019) and HSRC2016 data set (https://sites.google.com/site/hrsc2016/) are two typical remote sensing data sets.

We use the proposed method to experiment with these two data sets. In order to quantitatively verify the effectiveness of the experimental results, the following data indicators are used to evaluate the object detection and recognition.

(1)mAP
\[
mAP = \frac{1}{n} \sum_{i=1}^{n} \int_0^1 \text{Precision}_i \times \text{Recall}_i
\]  
(18)

(2)IoU
The full name of IOU is intersection over union, and the calculation method is the ratio of intersection and union of “predicted border” and “real border”. It is usually used to represent the 2D frame accuracy index of the object
\[
\text{IOU} = \frac{A \cap B}{A \cup B}
\]  
(19)

(3)Cos(eRE)
In order to judge the similarity of two rotation matrices, we construct a judgment function,
\[
e_{RE} = \arccos \left( \frac{\text{Tr} \left( RR^{-1} \right) - 1}{2} \right)
\]  
(20)
where \( \tilde{R} \) is the predicted value of the rotation matrix, \( \hat{R} \) is the true value of the rotation matrix.

When calculating the mAP, we require object 2D frame accuracy index (IOU) > 0.5 and \( \text{COS} (\text{eRE}) > 0.95 \). This ensures that the objects not only meet the correctness of the horizontal frame but also ensure the correctness of 3D rotation.

\[(4)f_1\]

The \( f_1 \) value represents the harmonic average of precision and recall,

\[ f_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \] (21)

A. ABLATION STUDY

In order to verify the effectiveness of the proposed network model with constraint layer, the network model with constraint layer and the network model without constraint layer is used to test DOTA1.5 data set. A variety of performance curves are obtained, including average loss, recognition accuracy index map (mAP), recognition rate index \( f_1 \) under 0.25 threshold, Recall, Precision, IOU, and oriented accuracy index (avg_dot). The oriented accuracy index is the dot product of prediction vector and truth vector, which is used to judge the accuracy of orientation prediction. As shown in Fig.5, the horizontal axis represents the number of epochs, and the vertical axis represents the corresponding value. The green represents the network model with constraint layer, and the blue represents the network model without constraint layer. The variable avg_dot in the 7th subgraph represents \( \text{Tr} \left( \tilde{R}\hat{R}^{-1} \right) - 1/2 \), which represents the cosine of the angle between the predicted attitude and the true attitude. When IOU > 0.5, it can be seen that the avg_dot of the constraint network is closer to 1, indicating that the predicted attitude of the constraint network is closer to the true value. Because the attitude estimation is more accurate, the performance indexes of the network model with constraint layer are better than those without constraint layer. After adding the constraint network, the output of the model naturally meets the constraint conditions. The network can learn in the constraint domain space, which greatly improves the speed and accuracy of learning.

B. DOTA 1.5 DATA TEST RESULTS

We experimented with DOTA1.5 data. We test 16 kinds of object extraction experiments such as aircraft, ships, and vehicles on 80875 images of the data set. The experiment is carried out on the image with the pixel size of 640 \( \times \) 640, and the test speed is 0.046s on a single RTX 2080Ti. The results are shown in TABLE 1. It can be seen from TABLE 1 that the mAP value reaches 71.66.

At present, few scholars have studied the object pose detection based on aerial images. Therefore, this paper compares with the method using DOTA1.5 data as experimental data. The comparison results are shown in TABLE 2. It can be seen that the mAP value of the proposed method is higher than that of the comparison algorithm, especially for Plane, whose AP value is as high as 97. However, the AP value of the CC (Container Crane) class is low. This is because

\[
X_{\text{box}} = \begin{bmatrix}
(u_{KL}r_{31} - r_{11})x_{IL} + (u_{KL}r_{32} - r_{12})y_{IL} + (u_{KL}r_{33} - r_{13})z_{IL} \\
(u_{KR}r_{31} - r_{11})x_{IR} + (u_{KR}r_{32} - r_{12})y_{IR} + (u_{KR}r_{33} - r_{13})z_{IR} \\
(v_{K}r_{31} - r_{21})x_{IT} + (u_{KR}r_{32} - r_{22})y_{IT} + (u_{KR}r_{33} - r_{23})z_{IT} \\
(v_{K}r_{31} - r_{21})x_{IB} + (u_{K}r_{32} - r_{22})y_{IB} + (u_{K}r_{33} - r_{23})z_{IB}
\end{bmatrix}
\] (16)
the number of samples of CC is too small, which leads to the low mAP value of this class and reduces the overall mAP value.

FIGURE 6 shows the PR curve of each type of object according to the experimental results of DOTA1.5. In the figure, recall is the abscissa axis and precision is the ordinate axis. The ratio of the area enclosed by the PR curve and the abscissa and ordinate axis to the total area is the AP value of this kind of object, which can be statistically obtained during the experiment. The mean value mAP of the AP value of 16 objects is the final position of the green line in the second subgraph in FIGURE 5. As can be seen from TABLE 1, this value is about 71.66. In addition, according to FIGURE 6, it can be seen that the AP value of Plane is the highest and that of CC is the lowest. This is because the number of samples of CC class is too small, and the results correspond to TABLE 2.

Based on the attitude quaternion of the object, the three-dimensional position and attitude of the object relative to the camera can be further obtained by using the method of literature [19]. The experimental effect of DOTA1.5 data set is shown in FIGURE 7.

In the figure, the red arrow is the x-axis direction of the object’s body. The green arrow is the y-axis direction of the object’s body, that is, the tail direction of the object. The blue arrow is the z-axis direction of the object’s body, pointing to the ground. We only learned the directions of planes, ships, large vehicles and small vehicles. Other objects, such as storage tanks and swim pools, have no directions. We do not learn in the three new branches, but only learn the horizontal box of the object in the first three branches. The proposed algorithm can show the exact direction of the object. There may be four possible directions for the output of the traditional tilt box, which is not clear.
TABLE 2. Performance index comparison result.

| method          | Plane | BD  | Bridge | GRF  | SV   | LV   | Ship | TC   | HC   | ST   | SBF  | RA   | Harbor | SP   | HC   | CC   | mAP |
|-----------------|-------|-----|--------|------|------|------|------|------|------|------|------|------|-------|------|------|------|-----|
| Propose method  | 97.57 | 89.23 | 59.20  | 61.50 | 68.17 | 85.71 | 96.24 | 90.07 | 74.39 | 77.07 | 72.97 | 85.19 | 76.27  | 70.32 | 41.35| 0.67 | 71.66|
| RetinaNet-GO[25]| 71.43 | 77.64 | 42.12  | 64.65 | 44.53 | 56.79 | 73.31 | 90.84 | 76.02 | 59.96 | 46.95 | 69.24 | 59.65  | 64.52 | 48.06| 0.83 | 59.36|
| FR-GO[27]       | 71.89 | 74.47 | 44.45  | 59.87 | 51.28 | 68.98 | 79.37 | 90.78 | 77.38 | 67.50 | 47.75 | 69.72 | 61.22  | 65.28 | 60.47| 1.54 | 62.00|
| Mask R-CNN[27]  | 76.84 | 73.51 | 49.90  | 57.80 | 51.31 | 71.34 | 79.75 | 90.46 | 74.21 | 66.07 | 46.21 | 70.61 | 63.07  | 64.86 | 57.81| 9.42 | 62.67|
| HTCP[29]        | 77.80 | 73.67 | 51.40  | 63.99 | 51.54 | 73.31 | 80.31 | 90.48 | 75.12 | 67.34 | 48.51 | 70.63 | 64.84  | 64.48 | 55.87| 5.15 | 63.40|
| R-DR[27]        | 79.20 | 82.81 | 51.92  | 71.41 | 52.38 | 75.73 | 80.92 | 90.83 | 72.81 | 68.64 | 49.29 | 72.93 | 73.56  | 70.55 | 63.33| 11.53| 66.66|

FIGURE 9. Visual effect of object detection and recognition in HSRC2016 data set.

In FIGURE 7, the main object of the first three pictures in the first row is car, and the main object of the last three pictures is plane. Both car and plane have directions, so the proposed algorithm can not only detect the position of the object, but also detect the direction of the object at the same time. The orientation of car and plane are the opposite direction of the green arrow in the figure. The comparison algorithm only uses the rotation box to find the object, and does not fully confirm the orientation of the object. The main object of the first picture in the second row are ship and harbor. It can be seen from the figure that the proposed algorithm can accurately detect the position and direction of the ship. At the same time, the harbor can also be detected, but because the object has no direction, the algorithm only detects its position and does not calculate its direction. The last three pictures show the harbor and swimming pool. The first three pictures in the third row in the figure is bridge, which are difficult to interpret manually, but they can be identified by using the proposed algorithm. The last three are storage tanks. The pose detection of remote sensing objects in complex environments (different scales, different illumination, blur, weak and small) is very challenging. For many small objects, it is difficult for human eyes to identify the direction.

C. HSRC2016 DATA SET TEST RESULT

We experimented with the HSRC2016 data set. The ship detection experiment was carried out on 350 images of the data set. The accuracy of each index and the recognition speed is shown in TABLE 3. In order to verify the effectiveness of the proposed algorithm, it is compared with three other algorithms [25]–[29] that also experiment on...
TABLE 3. Precision indexes of HRSC 2016 data set test result.

| Technical indicators | mAP | fl | Cos(eRE) | IOU |
|----------------------|-----|----|----------|-----|
| propose method       | 92.54 | 90.86 | 99.01 | 70.29 |
| Roi rans. [23]       | 86.2 | - | - | - |
| GV [24]              | 88.2 | - | - | - |
| R3Det [27]           | 89.33 | - | - | - |
| RRD [29]             | 84.3 | - | - | - |
| CSL [29]             | 89.62 | - | - | - |

HSRC2016 data set. The comparison results are shown in TABLE 3. It can be seen from TABLE 3 that the mAP values of the three comparison algorithms are lower than 90, while the mAP value of the proposed algorithm is as high as 92.54.

FIGURE 8 shows the PR curve of each type of object according to the experimental results of HSRC2016 data set. In the figure, recall is the horizontal axis and precision is the vertical axis. The ratio of the area enclosed by the red curve and the horizontal and vertical axis to the total area is the AP value of ship. Since the HSRC2016 data set has only one class of objects, the AP value of the ship is the mAP value of the experimental results. It can be seen from the figure that there are very few missed objects in the experimental results, and the detection accuracy is high. It corresponds to the mAP value in TABLE 3.

The experimental effect of HSRC2016 data set is shown in FIGURE 9. The object of HSRC2016 data set is ship with direction. In the figure, the red arrow is the x-axis direction of the object’s body. The green arrow is the y-axis direction of the object’s body, pointing to the sky. Therefore, the direction of the ship is the direction indicated by the green arrow in the figure. In all comparison algorithms, the rotating box is only used to detect the object, rather than determine the direction of the object in the actual sense. The experimental results show that the proposed algorithm can not only detect the ship object but also determine the direction of the object.

IV. CONCLUSION

This paper designed the network transformation layer and constructs a new constraint network model based on yolov3 structure, which we call the qfiled PoseNet model. The model can predict the probability field of object and unit quaternion field respectively. The object’s probability field is generated to determine the 2D bounding box of the object, and the unit quaternion field is generated to determine the 3D rotation R. And the combining the rotation matrix R and the 2D bounding box of the object to calculate the 3D translation. This paper implements a fast algorithm to infer the 3D pose and position of the object relative to the camera according to the aerial image. The proposed algorithm can detect the pose of the object in the aerial image, that is, while detecting the position of the object, it can obtain its actual orientation.

REFERENCES

[1] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, “PVNet: Pixel-wise voting network for 6DoF pose estimation,” presented at the IEEE/CVF Conf. Vis. Pattern Recognit. (CVPR), Los Angeles, NV, USA, 2019.
[2] I. Gordon and D. G. Lowe, “What and where: 3D object recognition with accurate pose,” in Toward Category-Level Object Recognition, vol. 4170, Berlin, Germany: Springer, 2006, pp. 67–82.
[3] T.-T. Do, M. Cai, T. Pham, and I. Reid, “Deep-6DPose: Recovering 6D object pose from a single RGB image,” 2018, arXiv:1802.10367.
[4] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, “Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes,” presented at the Asian Conf. Comput. Vis., Singapore, 2012.
[5] S. Hinterstoisser, V. Lepetit, N. Rajkuma, and K. Konolige, “Going further with point pair features,” presented at the Eur. Conf. Comput. Vis., Amsterdam, The Netherlands, 2016.
[6] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-DOF camera relocalization,” presented at the IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, 2015.
[7] A. Kendall and R. Cipolla, “Geometric loss functions for camera pose regression with deep learning,” presented at the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, 2017.
[8] S. Mahendran, H. Ali, and R. Vidal, “3D pose regression using convolutional neural networks,” presented at the IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Honolulu, HI, USA, 2017.
[9] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, “Siamese regression networks with efficient mid-level feature extraction for 3D object pose estimation,” 2016, arXiv:1607.02257.
[10] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, “SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again,” presented at the IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, 2017.
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” presented at the Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 21–37.
[12] M. Rad and V. Lepetit, “BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth,” presented at the IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, 2017.
[13] B. Tekin, S. N. Sinha, and P. Fua, “Real-time seamless single shot 6D object pose prediction,” presented at the IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Lake City, UT, USA, 2018.
[14] J. Redmon, S. Divvala, and R. Girshick, “You only look once: Unified, real-time object detection,” presented at the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016.
[15] M. D. Gould, “Introduction to Lie algebras and representation theory,” in Graduate Texts in Mathematics. Berlin, Germany: Springer-Verlag, 1972.
[16] R. Goldman, “Modeling perspective projections in 3-dimensions by rotations in 4-dimensions,” Graph. Models, vol. 75, no. 2, pp. 41–55, Mar. 2013.
[17] R. W. Brockett, “Robotic manipulators and the product of exponentials formula,” in Mathematical Theory of Networks and Systems. Berlin, Germany: Springer-Verlag, 1984, pp. 120–127.
[18] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, “Siamese regression networks with efficient mid-level feature extraction for 3D object pose estimation,” 2016, arXiv:1607.02257.
[19] Y. Hong, J. Liu, Z. Jahangir, S. He, and Q. Zhang, “Estimation of 6D object pose using a 2D bounding box,” Sensors, vol. 21, no. 9, p. 2939, Mar. 2021.
[20] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” presented at the Proc. IEEE Int. Conf. Comput. Vis., Venice, Italy, 2017.
[21] S. Q. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, p. 28.
[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” presented at the IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, 2017.
[23] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, “Hybrid task cascade for instance segmentation,” presented at the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Angeles CA, USA, 2019.

[24] J. Han, J. Ding, N. Xue, and G. S. Xia, “ReDet: A rotation-equivariant detector for aerial object detection,” presented at the IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021.

[25] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, “Learning RoI transformer for oriented object detection in aerial images,” presented at the IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, 2019.

[26] Y. Xu, M. Fu, Q. Wang, Y. Wang, and K. Chen, “Gliding vertex on the horizontal bounding box for multi-oriented object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 4, pp. 1452–1459, Apr. 2021.

[27] X. Yang, J. Yan, Z. Feng, and T. He, “R3Det: Refined single-stage detector with feature refinement for rotating object,” 2019, arXiv:1908.05612.

[28] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, “Rotation-sensitive regression for oriented scene text detection,” presented at the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018.

[29] G. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “DOTA: A large-scale dataset for object detection in aerial images,” presented at the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018.

**HONGYUN ZHANG** received the Ph.D. degree from Liaoning Technical University, in 2020. She is currently a Postdoctoral Student with Wuhan University. Her main research interest includes remote sensing image processing.

**JIN LIU** received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, in 2005. He is currently an Associate Professor with Wuhan University. His research interests include pattern recognition, computer vision, and artificial intelligence.

**YONGJIAN GAO** is currently pursuing the master’s degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. His research interests include object detection, machine learning, and computer vision.

---

**HONGYUN ZHANG**

**JIN LIU**

**YONGJIAN GAO**