Article

Oriented Vehicle Detection in Aerial Images Based on YOLOv4

Tai-Hung Lin and Chih-Wen Su *

Department of Information & Computer Engineering, Chung Yuan Christian University, Taoyuan 320, Taiwan
* Correspondence: lucas@cyu.edu.tw

Abstract: CNN-based object detectors have achieved great success in recent years. The available detectors adopted horizontal bounding boxes to locate various objects. However, in some unique scenarios, objects such as buildings and vehicles in aerial images may be densely arranged and have apparent orientations. Therefore, some approaches extend the horizontal bounding box to the oriented bounding box to better extract objects, usually carried out by directly regressing the angle or corners. However, this suffers from the discontinuous boundary problem caused by angular periodicity or corner order. In this paper, we propose a simple but efficient oriented object detector based on YOLOv4 architecture. We regress the offset of an object’s front point instead of its angle or corners to avoid the above mentioned problems. In addition, we introduce the intersection over union (IoU) correction factor to make the training process more stable. The experimental results on two public datasets, DOTA and HRSC2016, demonstrate that the proposed method significantly outperforms other methods in terms of detection speed while maintaining high accuracy. In DOTA, our proposed method achieved the highest mAP for the classes with prominent front-side appearances, such as small vehicles, large vehicles, and ships. The highly efficient architecture of YOLOv4 increases more than 25% detection speed compared to the other approaches.

Keywords: object detection; oriented bounding box; aerial image

1. Introduction

With the rise of convolutional neural networks (CNNs) in the past decade, object detection performance has been rapidly enhanced based on its rich feature representation. Various detection schemes have been proposed to make good use of the features. They are mainly divided into two categories: two-stage detectors [1–4] and single-stage detectors [5,6]. Although each has its merits, they both use Horizontal Bounding Boxes (HBB) to locate all kinds of objects from arbitrary views. However, for some specific applications, such as vehicle/building detection in remote sensing images and scene text detection, an Oriented Bounding Box (OBB) fits the target more closely than an HBB. On the other hand, objects with large aspect ratios and that are densely arranged also make it challenging for detectors to locate the entire object area while excluding the non-object area by HBB. Previous research has shown that OBB can help detect the above cases, which usually occur in remote sensing images. Furthermore, the orientation of an object may also provide vital information for further processing.

Recently, increasingly oriented object detectors, which adjust based on the classic object detector, have achieved promising results in remote sensing. At the same time, the maturity of cost-effective drones in manufacturing has led to the widespread use of aerial photography in many areas, such as precision agriculture, rescue work, wildlife monitoring, etc. Among the numerous aerial photography applications, traffic surveying is one of the applications that requires accurate vehicle location via OBB rather than HBB. Since most vehicles can be adequately located by OBB in the bird view, the area of each vehicle can be efficiently represented by the corners of OBB. The estimation of OBB is usually accomplished by the regression of its representation model, which usually contains five to eight parameters. However, many previous researchers calculated these parameters based...
on the limited quadrilateral and OBB’s angle definitions in OpenCV. Ambiguous corner ordering and angular periodicity cause the discontinuous boundary problem and obstruct the regression of parameters [7,8]. The boundary discontinuity problem often causes the model’s loss value to suddenly increase at the boundary situation.

Aiming to solve the problem above, we propose a simple, practical framework for oriented object detection, which can be used in traffic surveys. To avoid the angular discontinuity, we regress the front point offset of the object rather than its angle.

As mentioned above, the HBBs have been widely used to indicate the position and area of objects in general object detection tasks. For instance, the classic two-stage object detector Faster RCNN [1], which evolved from RCNN [2] and Fast RCNN [3], is to filter out the proposals from Region Proposal Network layer (RPN) in the first stage and goes through a second stage for position tuning and objects classification. RPN requires pre-defined anchors to generate proposals also designed in the form of HBB. R-FCN [4] combined Region of Interest (ROI) pooling and position-sensitive score maps to improve detection speed and alleviate object position insensitivity in CNN. On the other hand, a one-stage detector, such as YOLO [5], SSD [6] or RetinaNet [9], evaluates images in a single phase to classify and locate objects simultaneously without second-stage calibration. Although the accuracy of a one-stage detector is lower than that of a two-stage detector initially, a single-stage structure allows for higher detection speeds. To bridge the gap inaccuracy, some functional mechanisms, including anchor box, Feature Pyramid Network (FPN) [10] and Cross Stage Partial Network (CSPNet) [11], are gradually introduced in the YOLO series [12–15]. The state-of-the-art object detectors mentioned above all use HBB to describe the position of arbitrary shape objects.

In order to obtain the position and orientation of rectangular objects while eliminating background interference, OBB becomes a better alternative to HBB. Researchers have proposed many fascinating oriented object detectors based on two-stage and one-stage models. Earlier oriented object detectors were based on the two-stage scheme and applied to detect scene text. RRPN [16] directly adds an angle parameter to the RPN of Faster RCNN to make proposals directional. Then, the arbitrarily oriented proposals were converted to the fixed-size feature maps by Rotation Region of Interest (RROI) pooling to detect the oriented scene text. In R²CNN [17], Jiang Y. et al. mentioned that the shape of a bounding box is very similar, but the angle difference is enormous when rotated $90^\circ$ and $-90^\circ$. The discontinuity of the angle may cause difficulties in network learning. Therefore, they adopted the coordinates of the first two corners clockwise and the height of the bounding box to represent an inclined rectangle $(x1, y1, x2, y2, h)$. To catch significant aspect ratio texts, they also performed ROI pooling with different pool sizes ($7 \times 7$, $11 \times 3$, $3 \times 11$). To compensate for the lack of angle information in the general object detector, the ROI transformer [18] learns the transformation parameters to convert horizontal ROI to the rotated ROI and extract rotation-invariant features for the classification and localization afterward. In recent years, more and more oriented object detectors have been proposed and applied to aerial photographs. Due to the large variability of object size/orientation and the complexity of the background in aerial scenes, many orientated object detectors focus on deeper feature extraction and attention mechanisms. R-DFPN [19] proposed a dense feature pyramid network (DFPN) to better integrate features of different scales to increase the usage rate of features. ICN [20] combines multiple modules, such as an image cascade network, feature pyramid network and deformable inception network, to achieve satisfactory performance in detecting oriented objects. SCRDet [7], CADet [21] and RADet [22] add different attention modules before RPN to better handle small objects and cluttered oriented objects. A gliding vertex [23] guarantees detection accuracy through regressing vertices offset without changing network architecture.

Due to the lower computational cost and memory requirement, some oriented object detectors were developed based on a one-stage scheme. Textbox++ [24] and RRD [25] adopt the vertex regression on SSD while the latter changes the backbone convolution into oriented response convolution. Considering the difficulty of angle training in R-DFPN,
SAR [26] converts some of the rotating parameters from the angular regression task to the classification task to avoid boundary and order problems. CSL [27] uses the circular smooth label technique to obtain a more robust angular prediction through classification without suffering boundary conditions caused by periodic angles. Unlike a two-stage detector, a single-stage detector does not have ROI pooling or an ROI align phase to reduce the impact of feature misalignment. R3Det [28] adds a refinement stage based on RetinaNet to achieve feature alignment by reconstructing the feature map and using a deeper network backbone to extract richer features. In addition to adding the refinement stage, RSDet [8] directly regresses the four corner coordinates of the oriented bounding box to avoid inherent regression inconsistencies. A modulated rotation loss is designed to address the problem of loss discontinuity.

No matter two-stage or one-stage oriented object detectors, there are three main ways to determine an oriented bounding box: the five-parameter regression-based method, the eight-parameter regression-based method and the classification-based method. Five-parameter regression-based methods usually introduce an additional parameter \( \theta \) to regress the angle of the bounding box. The range of \( \theta \) is usually limited to 90 degrees [7,19,21,28] or 180 degrees [16]. However, the network is prone to suffering from the loss discontinuity problem in the boundary case. In eight-parameter regression-based methods [8,20,23–25], the coordinates of the quadrilateral’s four corners are regressed. The boundary discontinuity problem still exists due to the ordering of corner points. Classification-based methods [27,29] turn the angle regression task into the classification task, which truly addresses the essence of boundary problem. Nevertheless, it loses the precision caused by angle discretization and also increases the difficulty of tuning the network caused by importing the new hyperparameter window function. Moreover, almost none of the above methods can directly determine the front side of an object. It is practical to provide the front information of an object in traffic analyses or sailing analyses.

To achieve highly efficient oriented object detection, we adopt YOLOv4 architecture and extend its head part to estimate object orientation. To overcome the boundary discontinuity problem, the front point of the object is introduced. Experimental results demonstrate that our proposed method outperforms the others in terms of vehicle detection, which plays a vital role in traffic analysis.

2. Materials and Methods

2.1. The Problems of Regression-Based Rotating Object Detector

As described in Section 1, the Boundary Discontinuity Problem (BDP) is a significant obstruction of the estimation of OBB. An OBB can be determined by the center \((x, y)\), width \(w\), height \(h\) and angle \(\theta\) of it, or just by its four corners \((x_1, y_1), (x_2, y_2), (x_3, y_3),\) and \((x_4, y_4)\). For the former five-parameter OBB models, the angle \(\theta\) is defined as the acute angle between the x-axis and the first side (width) of OBB that it touches counterclockwise as the definition in OpenCV, limited to the range \([-\pi/2, 0)\) as the Figure 1a. In this case, the acute angle could be formed by x-axis and any side of OBB, which means the width \(w\) and height \(h\) could easily be confused and cause Exchangeability of Edges (EoE). To alleviate the impact of EoE, some studies further define \(\theta\) as the acute angle between the x-axis and the first long side of OBB, thus extending the range from \([-\pi/2, 0)\) to \([-\pi, 0)\). The length of the long side is then explicitly specified as the width \(w\) and that of the short side is specified as the height \(h\) as shown in Figure 1b. However, the value of the acute angle is still not continuous, while the long side of OBB is almost parallel to the x-axis. The above periodicity of angular (PoA) may cause significant loss changes once the angle falls at either end of its permissible range.
which may be undesirable when representing artificial buildings and vehicles. In addition, the eight-parameter OBB model determines rectangles and arbitrary quadrilaterals, which may be undesirable when representing artificial buildings and vehicles. In this case, an OBB is represented by six parameters, as shown in Figure 1c, and width is defined as the length of the front/rear side and height refers to the length of the other side. Inspired by the YOLO grid cell strategy, the angle prediction is converted to that of the front point offset parameters to indicate the orientation of the object, the eight-parameter OBB model discards the angle information. Although the problem of PoA does not occur in the regression of OBB’s four corners, the consistency of the alignment of these four corners remains an issue. In addition, the eight-parameter OBB model determines rectangles and arbitrary quadrilaterals, which may be undesirable when representing artificial buildings and vehicles.

2.2. Front Point Offset Regression for Angle Prediction

To overcome the EoE and PoA problems caused by the five-parameter OBB model and to extend the angle range to \([-2\pi, 0]\), the parameter \(\theta\) is replaced by a couple of parameters \(x_f\) and \(y_f\), where \((x_f, y_f)\) indicates the midpoint of the front side of the objects, such as vehicles and buildings. In this case, an OBB is represented by six parameters, as shown in Figure 1c, and width is defined as the length of the front/rear side and height refers to the length of the other side. Inspired by the YOLO grid cell strategy, the angle prediction is converted to that of the front point offset \((b_x f, b_y f)\) with respect to the grid, as shown in Figure 2. Network outputs \((t_x, t_y, t_w, t_h)\) pass through a sigmoid function \(\sigma\) and adjust the size with the predefined anchor \((p_w, p_h)\) to reconstruct the predict bounding box \((b_x, b_y, b_w, b_h, b_x f, b_y f)\). Here, the front point and center respect the same grid for each OBB to make the training more convergent. Since the prediction of angle is translated from periodic angle space into the distance space domain, the PoA problem no longer exists.

![Figure 1. Three different definitions of the angle, width and height of the OBB. Green point and red point represent center point and front point of the OBB, respectively.](image)
In addition, the assignment of the front point makes the alignment of corners no longer ambiguous and confusing, which fundamentally avoids the EoE problem.

\begin{align*}
  b_x &= 2\sigma(t_x) - 0.5 + C_x \\
  b_y &= 2\sigma(t_y) - 0.5 + C_y \\
  b_w &= p_w(2\sigma(t_w))^2 \\
  b_h &= p_h(2\sigma(t_h))^2 \\
  b_{xf} &= t_{xf} + C_x \\
  b_{yf} &= t_{yf} + C_y
\end{align*}

Figure 2. The network predicts 6 coordinates \(t_x, t_y, t_w, t_h, t_{xf} \) and \( t_{yf} \) of each bounding box. \((C_x, C_y)\) is the cell offset from the top left corner of the image. Red point and blue point denote the front point and center point of bounding box, respectively. Both front point and center point share the same grid cell in the network.

2.3. Overview of the Network

The proposed network architecture is based on YOLOv4, as illustrated in Figure 3. YOLOv4 is one of the fastest and most accurate object detectors available today. Its architecture consists of three parts: backbone, neck and head. The backbone, CSPDarknet53, in YOLOv4 optimizes darknet53 with Cross Stage Partial Network (CSP) to reduce the reuse of gradient information, thereby decreasing computational cost and increasing learning ability. Yolov4 utilizes Spatial Pyramid Pooling (SPP) for expanding the field of view in the neck part. It combines Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN) for fusing low-level and high-level features from different scales to achieve better localization. The head part is the prediction layer which contains a certain amount of grid cells at three different scale levels. Each grid cell outputs the prediction results into a \((C + 5) \times A\) dimensional vector, with \(C\) representing the probability of each class, and \(A\) is the number of predefined anchors in each prediction layer. The remaining five dimensions encode a horizontal bounding box into \(x, y, w, h\) and confidence. We take advantage of the above YOLOv4 architecture and add two additional parameters \(t_{xf}\) and \(t_{yf}\) for each grid cell to predict front point offset. As a result, the proposed method can localize objects by OBB and obtain the following advantages:

1. Avoid boundary discontinuity problem

Both the angle regression and corner regression methods will produce certain boundary discontinuity problems that lead to instability in loss calculation. In contrast, our proposed method regresses front point offset and avoids the sharp increase of loss generated by the boundary discontinuity problems.

2. Less predefined anchors
It is not only necessary to use anchors of different sizes and aspect ratios to provide anchors closer to the ground truth for the five-parameter oriented object detector but also to add anchors at multiple angles to help the model converge, which also increase computational cost in the process. Our approach does not focus on the regression of angle. We regress the front point offset to maintain the overall performance of the model without increasing the number of anchors.

3. Explicit front end information

Since our method locates the front point coordinates of the object, we can avoid boundary discontinuous and obtain the object’s orientation within \([-2\pi, 0]\). This helps to indicate the front end of the stopped vehicle and to understand whether the vehicle is in violation and the possible direction of travel. In contrast, the other OBB definitions are more likely to face the discontinuity problem of extreme values of angles and provide limited angle ranges.

![Figure 3](image)

**Figure 3.** The network architecture of our proposed method. The feature extraction (backbone) and feature fusion (neck) parts are kept the same as YOLOv4. Meanwhile, two additional channels \(t_x\) and \(t_y\) are added into the grid cells (head) at different scale levels. Thus, each grid cell contains an \((C + 7)\) dimensional vector, where \(C\) is the number of classes, \(A\) is the number of predefined anchors in each prediction layer and each anchor contains seven parameters: \(l_x, l_y, l_xf, l_yf, l_w, l_h\) and confidence \(p_{obj}\). \(\odot\) denotes feature map concatenation.

2.4. Loss Function

The loss function is decomposed into four parts, which are defined as follows:

\[
L = L_{cls} + L_{obj} + L_{box} + L_{fp}
\]

\[
L_{cls} = \frac{\lambda_1}{N_{pos}} \sum_{i=0}^{S^2} \sum_{j=0}^{A} 1^{obj}_{ij} \sum_{ceClasses} [p_{ij}(c) \log(p_{ij}(c)) + (1 - p_{ij}(c)) \log(1 - p_{ij}(c))] \tag{6}
\]

\[
L_{obj} = \frac{\lambda_2}{N} \sum_{i=0}^{S^2} \sum_{j=0}^{A} 1^{obj}_{ij} [\hat{C}_{ij} \log(C_{ij}) + (1 - \hat{C}_{ij}) \log(1 - C_{ij})] + \hat{1}^{noobj}_{ij} [\hat{C}_{ij} \log(C_{ij}) + (1 - \hat{C}_{ij}) \log(1 - C_{ij})] \tag{7}
\]

\[
L_{box} = \frac{\lambda_3}{N_{pos}} \sum_{i=0}^{S^2} \sum_{j=0}^{A} 1^{obj}_{ij} L_{IoU} \tag{8}
\]

\[
L_{fp} = \frac{\lambda_4}{N_{pos}} \sum_{i=0}^{S^2} \sum_{j=0}^{A} 1^{obj}_{ij} \text{smooth}_{L1}(f, f_p) \tag{9}
\]

where \(S^2\) is the number of grid cells in the feature map and \(A\) denotes the number of anchors in each grid cell. \(N_{pos}\) is the number of positive samples. \(N\) is the number of positive samples and negative samples. \(L_{cls}\) denotes classification loss. \(L_{obj}\) denotes objectness loss. We use binary cross entropy to calculate both \(L_{cls}\) and \(L_{obj}\). \(L_{box}\) is the OBB.
regression loss based on IoU (Intersection over Union). \( L_{fp} \) denotes the regression loss of front point offset. \( 1_{ij}^{obj} \) and \( 1_{ij}^{noobj} \) are both binary values, which indicate positive samples or not. \( \lambda_1 \) to \( \lambda_4 \) are the hyper-parameters to balance the weights of each loss term. Since the factor of the direction of front point is not considered in the IoU, the IoU can be very small, even if the front point of two OBBs is located in a completely different direction. To penalize the loss of IoU by using the angular difference caused by the front points, the correction of the vectors formed by the center point and front point is introduced in \( L_{IoU} \) as below:

\[
L_{IoU} = IoU \cdot \cos \left( \frac{\angle (\vec{P_c}P_f, \vec{T_c}T_f)}{2} \right)
\]

where \( P_c \) and \( P_f \) indicate the center point and front point of a predicted box. \( T_c \) and \( T_f \) represent the center point and front point of ground truth box. The cosine value of half of the angle formed by vectors \( \vec{P_c}P_f \) and \( \vec{T_c}T_f \) is in the range \([0, 1]\), which ensures that the value of \( L_{IoU} \) remains in the range \([0, 1]\).

3. Datasets and Results

The proposed method was implemented by PyTorch [30] with Tesla V100 and 32G memory. To verify the performance of our proposed method, we conducted experiments on three datasets, which contain OBB labeling information.

3.1. Datasets and Evaluation Protocols

3.1.1. DOTA

DOTA [31] is one of the large-scale datasets for object detection in the aerial image. It contains 2806 images from different sensors and platforms. The size of images range from approximately 800 \( \times \) 800 to 4000 \( \times \) 4000. The fully annotated DOTA images comprise 188,282 instances, each of which is annotated by an arbitrary (8 d.o.f) quadrilateral in 15 classes: plane (PL), baseball diamond (BD), bridge (BR), ground track field (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), round about (RA), harbor (HA), swimming pool (SP) and helicopter (HC). Half of the original images were randomly selected as the training set, 1/6 as the validation set, and 1/3 as the testing set. We cropped original images into 1024 \( \times \) 1024 patches with a stride set to 512. Finally, we obtained 29,457 patches for the training process.

3.1.2. HRSC2016

HRSC2016 [32] is a high-resolution ship dataset collected from six different harbors on Google Earth. It contains 1061 images with sizes ranging from 300 \( \times \) 300 to 1500 \( \times \) 900. A rotating rectangle labels ground truths. We followed the original division to obtain the training, validation and test sets with 436, 181 and 444 images, respectively. The size of HRSC2016 is significantly smaller than the DOTA in terms of the number of images, the number of categories and the diversity of content. In DOTA1.0, the number of ships used for training was around tens of thousands. Most of the images contain dozens to thousands of targets. On the other hand, HRSC2016 has about 3000 targets in all.

3.1.3. Our Dataset

To verify the robustness of the proposed method in traffic applications, we used UAVs to capture images from several road intersections at 65~100 m above the ground. All the images are sized 1920 \( \times \) 1080. Unlike remote sensing images, detection targets and vehicles are much closer to the camera, making the targets look much more significant in our images. We randomly selected 723 images for training and validation and 180 for testing. All images
are annotated with OBB in eight classes: sedan, truck, bus, tractor, trailer and motorbike. The number of each class is listed in Table 1.

**Table 1.** Evaluation results of our dataset.

| Class      | Ground Truth Counts | Precision | Recall | mAP  |
|------------|---------------------|-----------|--------|------|
| Sedan      | 14,924              | 0.978     | 0.975  | 0.981|
| Truck      | 2035                | 0.931     | 0.939  | 0.933|
| Bus        | 605                 | 0.958     | 0.976  | 0.974|
| Tractor    | 153                 | 0.938     | 0.891  | 0.879|
| Trailer    | 151                 | 0.833     | 0.849  | 0.803|
| Motorbike  | 26,314              | 0.940     | 0.761  | 0.800|
| All        | 44,182              | 0.930     | 0.898  | 0.895|

3.2. Training Details

CSPDarknet53 is the backbone of architecture, which was pretrained on COCO [33] and provided by YOLOv4. The stochastic gradient descent (SGD) was used as an optimizer in all datasets, and its weight decay and momentum were set to 0.0005 and 0.937, respectively. Training epochs and learning rates were set differently in each dataset. We trained 250 epochs on DOTA and 300 epochs on HRSC2016 and our dataset. The initial learning rates for DOTA, HRSC2016 and our dataset were 0.01, 0.0001 and 0.05, respectively. The mini-batch sizes of DOTA and HRSC2016 were set to eight and set to one for our dataset. The augmentation method is applied to all datasets, including mosaic, random flip, random color, random rotation and random scaling.

3.3. Result on DOTA1.0

Table 2 shows the comparison results on DOTA. The results here are obtained from the official DOTA evaluation server. Our proposed method attains a good trade-off between accuracy and speed. With test time augmentation (TTA), the mAP of our proposed method achieves 73.89% with 20.4 fps on average. Our results outperform all the other methods at the same speed level. Moreover, the three categories: small vehicle, large vehicle and ship obtain the highest mAP 79.37%, 83.34% and 88.65% in all methods. This may be due to the obvious orientation of these three types of objects and the fact that the bounding boxes of the objects are more rectangular in shape. Among the methods that are better than ours in the table, Gliding Vertex [23], Mask OBB [34], FFA [35], APE [36] and CenterMap OBB [37] are based on the slower two-stage detectors and RSDet [8], GCL [29] and CSL [27] are based on the single-stage methods while using deeper backbone (ResNet152). In terms of speed, our method is 5 fps faster than SAR [26], which is the fastest method on record in Table 2. The first reason is that our proposed approach is based on the YOLOv4 architecture, which is designed for high efficiency. Second, in some oriented object detectors, many anchors with different sizes and angles are proposed for better detection results. This leads to a significant increase in the number of detection candidates and increases the computation time when non-maximal suppression is applied. Our proposed six-parameter OBB model has the potential to maintain accuracy while keeping the minimum number of anchors.

3.4. Result on HRSC2016

HRSC2016 contains a large number of ships with arbitrary rotation and large aspect ratios, which is a great challenge to locate accurately. The comparison results are listed in Table 3. With TTA, the mAP of our proposed method achieves 93.7% under voc2012 evaluation protocol. It is worth noting that the inference speed for most methods is around 10 fps and that of the fastest method is 15.53 fps for 800 × 800 images. In comparison, our method can reach an inference speed of 19.23 fps and even 52.63 fps without TTA, which is more than five times faster than most SOTA methods.
Table 2. The comparison results on the DOTA1.0 dataset.

| Method | backbone | PL | SV | LV | SH | HC | BR | RD | DR | JB | FT | TC | ST | SW | RA | HA | SP | mAP | FPS |
|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| R2CNN [17] | ResNet101 | 90.94 | 99.22 | 50.91 | 55.81 | 49.22 | 65.67 | 35.34 | 67.44 | 90.67 | 64.92 | 72.39 | 55.06 | 52.53 | 53.14 | 50.67 |
| RRPN [16] | ResNet101 | 88.52 | 51.85 | 56.19 | 57.25 | 53.58 | 71.20 | 31.66 | 59.30 | 90.81 | 72.84 | 67.38 | 56.49 | 52.94 | 55.08 | 51.01 |
| RoI-Transformer [8] | ResNet101 | 88.64 | 68.81 | 76.68 | 83.59 | 47.67 | 78.52 | 43.44 | 75.92 | 90.74 | 77.27 | 81.46 | 58.39 | 53.54 | 62.83 | 58.95 | 69.56 | 5.90 |
| SCRDet [9] | ResNet101 | 89.98 | 68.36 | 60.32 | 72.41 | 65.21 | 80.65 | 52.09 | 68.36 | 90.85 | 87.94 | 86.86 | 65.02 | 66.68 | 66.25 | 68.24 | 72.61 |
| MFAIR-Net [13] | ResNet101 | 89.62 | 71.13 | 74.69 | 77.83 | 62.11 | 70.30 | 90.05 | 85.40 | 66.22 | 62.21 | 64.14 | 68.31 | 70.21 | 73.49 |
| Gliding Vertex [27] | ResNet101 | 89.64 | 70.71 | 73.14 | 66.82 | 57.32 | 85.00 | 52.46 | 77.34 | 90.74 | 70.81 | 69.37 | 59.55 | 70.81 | 70.94 | 70.92 | 10.00 |
| Mask OBB [26] | ResNetX101 [19] | 89.56 | 75.62 | 74.56 | 85.63 | 63.32 | 85.95 | 54.21 | 72.90 | 89.05 | 83.81 | 64.48 | 59.64 | 69.64 | 73.94 | 69.06 | 75.53 | 6.20 |
| FFA [41] | ResNet101 | 90.10 | 71.00 | 79.90 | 65.50 | 63.70 | 82.70 | 54.20 | 72.50 | 90.70 | 83.90 | 64.80 | 61.20 | 60.00 | 70.70 | 76.10 | 75.70 |
| R3Det [5] | ResNet101 | 89.96 | 74.01 | 77.16 | 79.45 | 65.53 | 85.62 | 53.42 | 76.03 | 90.03 | 87.15 | 64.72 | 67.72 | 63.35 | 78.45 | 71.94 | 79.75 |
| CenterMap OBB [27] | ResNet101 | 89.85 | 77.66 | 78.22 | 87.19 | 66.06 | 84.41 | 54.40 | 70.25 | 90.05 | 84.89 | 85.27 | 56.46 | 69.23 | 74.13 | 75.16 | 76.03 | 6.30 |
| Ours [10] | ResNet101 | 89.80 | 69.76 | 72.68 | 75.70 | 60.10 | 80.40 | 51.70 | 77.40 | 90.80 | 78.90 | 64.80 | 62.30 | 64.00 | 57.70 | 60.60 | 71.80 |

Table 3. The comparison results on HRSC2016.

| Method | Backbone | Image Size | mAP (07) | mAP (12) | FPS |
|--------|----------|------------|----------|----------|-----|
| R2CNN [17] | ResNet101 | 800 × 800 | 73.07 | 79.73 | 2.0 |
| RC1 & RC2 [47] | VGG16 | 800 × 800 | 64.00 | 51.70 | 1.0 |
| RRPN [16] | ResNet101 | 800 × 800 | 70.98 | 64.00 | 0.5 |
| R2-PN [48] | VGG16 | 800 × 800 | 60.64 | 51.70 | 1.0 |
| RetinaNet-H [28] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| R2Det [5] | VGG16 | 384 × 384 | 84.30 | slow |
| RetinaNet-H [28] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| RD [25] | VGG16 | 384 × 384 | 84.30 | 89.27 | 14.0 |
| Roll-Transformer [18] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| DS [5] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| Gliding Vertex [23] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| SAR [26] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| BBAVectors [41] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| GLS-Net [43] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |
| CSLP/FFN-based [27] | ResNet101 | 800 × 800 | 82.89 | 89.27 | 14.0 |

* indicates inference with test time augmentation.

3.5. Result on Our Dataset

Our dataset evaluation results and confusion matrix are shown in Table 1 and Figure 4. Vehicles are roughly divided into six categories: sedan, truck, bus, tractor, trailer and motorbike. Most of the large vehicles have an area of more than 4000 pixels, and the trailer may exceed 17,000 pixels. On the other hand, the area of a motorbike is less than 800. The mAP of a tractor and trailer only achieve 87.9% and 80.3% due to the lack of corresponding training samples. Nevertheless, the other three types of large objects can achieve over 93%, and sedans can even reach 98.1% mAP. In terms of small vehicles, the mAP of the motorbike is 80%. Low mAP is caused by two main reasons. The first reason is the small area of a motorbike and trailer, which may exceed 17,000 pixels. On the other hand, the area of a motorbike is less than 800. The mAP of a tractor and trailer only achieve 87.9% and 80.3% due to the lack of corresponding training samples. Nevertheless, the other three types of large objects can achieve over 93%, and sedans can even reach 98.1% mAP. In terms of small vehicles, the mAP of the motorbike is 80%. Low mAP is caused by two main reasons. The first reason is the small area of a motorbike and trailer, which may exceed 17,000 pixels. On the other hand, the area of a motorbike is less than 800.
could be easily mis-detected in a waiting zone or under the shade of trees. The second reason is that many incomplete motorbikes in the training images were mislabeled, which caused a high false positive rate, as shown in Figure 4.

**Figure 4.** Confusion matrix of our dataset. The horizontal axis is the ground truth and the vertical axis is the predicted result. Confidence threshold and IoU threshold are set to 0.01 and 0.1, respectively.

Figures 5–7 show some detection results on DOTA1.0, HRSC2016 and our dataset. The color of its OBB indicated the class of each detected object, and the white side denotes the front side of the object. As we can see, our model has outstanding detection results under various scales of aerial scenarios.

**Figure 5.** Cont.
Figure 5. Cont.
Figure 5. Detection results on DOTA1.0. The colors of OBBs are defined as follows: red: plane, yellow: baseball diamond, green: bridge, cyan: ground track field, blue: small vehicle, magenta: large vehicle, dark orange: ship, green yellow: tennis court, spring green: basketball court, deep sky blue: storage tank, dark violet: soccer ball field, deep pink: round about, orange: harbor, chartreuse: helicopter.

Figure 6. Detection results on HRSC2016.
Figure 7. Detection results on our dataset. Bounding box color is defined as follows: red: sedan, yellow: truck, green: bus, cyan: tractor, blue: trailer, magenta: motorbike.

4. Discussion

The evaluation from the experimental results shows that our method has a good trade-off between accuracy and speed, especially when the object has an explicit front-side appearance. However, the proposed method has weaknesses in the following cases:

(1) When an object does not have a visible front side, determining its front side could be ambiguous, as shown in Figure 8.

(2) It is still difficult to precisely locate objects with very large aspect ratio, as shown in Figure 9. In the future, we will try to add a lightweight attention module to locate the front point or use deformable convolution to detect objects with large aspect ratios.

Figure 8. The ambiguous problem of determining the front side of the tennis court. Since any opposite side of a tennis court is symmetrical, it is hard to define its front side clearly. The left detection result treats the longer side as the front side while the right one treats the shorter side as the front side. This makes the detection results inconsistent and leads to imprecise OBB positioning.
Figure 9. The OBB and front point of the object with very large aspect ratio is difficult to precisely locate.

5. Conclusions

In this paper, we proposed an arbitrary oriented object detector based on the YOLOv4 framework. We introduced the front point offset of an OBB to avoid the discontinuous boundary problem caused by the regression of angle or corners. The proposed method is especially suitable for vehicle localization in aerial images. It not only generates the OBB fit for the vehicle but also indicates the front end of it, which plays a vital role in traffic analysis. With the validation of the public datasets DOTA and HRSC2016, our detector was able to reach promising detection accuracy and achieve outstanding inference speed. The inference speed outperforms all the other methods and can accomplish real-time detection. Since our approach focuses on changing the head area’s structure, it is easier to further improve the accuracy and speed with future upgrades of the YOLO series.

Author Contributions: Conceptualization, T.-H.L. and C.-W.S.; methodology, T.-H.L. and C.-W.S.; software, T.-H.L.; validation, T.-H.L.; formal analysis, T.-H.L. and C.-W.S.; investigation, T.-H.L. and C.-W.S.; resources, C.-W.S.; data curation, T.-H.L.; writing—original draft preparation, T.-H.L.; writing—review and editing, T.-H.L. and C.-W.S.; visualization, T.-H.L.; supervision, C.-W.S.; project administration, T.-H.L.; funding acquisition, C.-W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology, Taiwan under Grants 110-2634-F-001-007.

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

References
1. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 1137–1149. [CrossRef]
2. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
3. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
4. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 5–10 December 2016; pp. 379–387.
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
6. 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, The Netherlands, 11–14 October 2016; Volume 9905, pp. 21–37. [CrossRef]
7. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Xian, S.; Fu, K. SCRDet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 8231–8240. [CrossRef]
8. Qian, W.; Yang, X.; Peng, S.; Guo, Y.; Yan, J. Learning Modulated Loss for Rotated Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Virtual, 2–9 February 2021; Volume 35, pp. 2458–2466.
9. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 40, 719–727. [CrossRef]
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, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125. [CrossRef]
11. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580. [CrossRef]
12. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767.
14. Bochkovskiy, A.; Wang, C.Y.; H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.
15. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-YOLOv4: Scaling Cross Stage Partial Network. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13024–13033. [CrossRef]
16. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans. Multimed. 2018, 20, 3111–3122. [CrossRef]
17. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection. arXiv 2017, arXiv:1706.09579.
18. 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 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2844–2853. [CrossRef]
19. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic Ship Detection in Remote Sensing Images from Google Earth of Complex Scenes Based on Multiscale Rotation Dense Feature Pyramid Networks. Remote Sens. 2018, 10, 132. [CrossRef]
20. Azimi, S.M.; Vig, E.; Bahmanyar, R.; Körner, M.; Reinartz, P. Towards Multi-class Object Detection in Unconstrained Remote Sensing Imagery. In Proceedings of the 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; pp. 150–165. [CrossRef]
21. 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]
22. Li, Y.; Huang, Q.; Pei, X.; Jiao, L.; Shang, R. RADet: Refine Feature Pyramid Network and Multi-Layer Attention Network for Arbitrary-Oriented Object Detection of Remote Sensing Images. Remote Sens. 2020, 12, 389. [CrossRef]
23. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Chen, K.; Xia, G.S.; Bai, X. Gliding Vertex on the Horizontal Bounding Box for Multi-Oriented Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. 2021, 43, 1452–1459. [CrossRef]
24. Liao, M.; Shi, B.; Bai, X. TextBoxes++: A Single-Shot Oriented Scene Text Detector. IEEE Trans. Image Process. 2018, 27, 3676–3690. [CrossRef]
25. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.S.; Bai, X. Rotation-Sensitive Regression for Oriented Scene Text Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5909–5918. [CrossRef]
26. Lu, J.; Li, T.; Ma, J.; Li, Z.; Jia, H. SAR: Single-stage anchor-free rotating object detection. IEEE Access 2020, 8, 205902–205912. [CrossRef]
27. Yang, X.; Yan, J. Arbitrary-Oriented Object Detection with Circular Smooth Label. In Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 677–694. [CrossRef]
28. Yang, X.; Liu, Q.; Yan, J.; Li, A.; Zhang, Z.; Yu, G. R2Det: Refined Single-Stage Detector with Feature Refinement for Rotating Object. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 2–9 February 2021; Volume 35, pp. 3163–3171.
29. Yang, X.; Yan, J.; He, T. On the Arbitrary-Oriented Object Detection: Classification based Approaches Revisited. Int. J. Comput. Vis. 2022, 130, 1340–1365. [CrossRef]
30. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
31. 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 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983. [CrossRef]
32. Yang, X.; Yan, J.; Sun, H.; Fu, K.; Zhang, T.; Zhang, Y.; Guo, Z.; Xian, S.; Xue, X. SCRDet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 8231–8240. [CrossRef]
33. Qian, W.; Yang, X.; Peng, S.; Guo, Y.; Yan, J. Learning Modulated Loss for Rotated Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Virtual, 2–9 February 2021; Volume 35, pp. 2458–2466.
34. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 42, 318–327. [CrossRef]
35. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125. [CrossRef]
32. Liu, Z.; Wang, H.; Weng, L.; 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. [CrossRef]

33. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common objects in context. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Volume 8693, pp. 740–755. [CrossRef]

34. Wang, J.; Ding, J.; Guo, H.; Cheng, W.; Pan, T.; Yang, W. Mask OBB: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images. *Remote Sens.* 2019, 11, 2930. [CrossRef]

35. Fu, K.; Chang, Z.; Zhang, Y.; Xu, G.; Zhang, K.; Sun, X. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS J. Photogramm. Remote Sens.* 2020, 161, 294–308. [CrossRef]

36. Zhu, Y.; Du, J.; Wu, X. Adaptive Period Embedding for Representing Oriented Objects in Aerial Images. *IEEE Trans. Geosci. Remote Sens.* 2020, 58, 7247–7257. [CrossRef]

37. Wang, J.; Yang, W.; Li, H.C.; Zhang, H.; Xia, G.S. Learning Center Probability Map for Detecting Objects in Aerial Images. *IEEE Trans. Geosci. Remote Sens.* 2019, 58, 7247–7257. [CrossRef]

38. Wang, J.; Yang, W.; Guo, H.; Cheng, W.; Pan, T.; Yang, W. Mask OBB: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images. *IEEE Geosci. Remote Sens. Lett.* 2016, 13, 1074–1078. [CrossRef]

39. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common objects in context. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Volume 8693, pp. 740–755. [CrossRef]

40. Hou, L.; Lu, K.; Xue, J.; Hao, L. Cascade detector with feature fusion for arbitrary-oriented objects in remote sensing images. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo, London, UK, 6–10 July 2020; pp. 1–6. [CrossRef]

41. Yi, J.; Wu, P.; Liu, B.; Huang, Q.; Qu, H.; Metaxas, D. Oriented object detection in aerial images with box boundary-aware vectors. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 2149–2158. [CrossRef]

42. Wang, Y.; Zhang, Y.; Zhang, Y.; Zhao, L.; Sun, X.; Guo, Z. SARD: Towards Scale-Aware Rotated Object Detection in Aerial Imagery. *IEEE Access* 2019, 7, 173855–173865. [CrossRef]

43. Li, C.; Luo, B.; Hong, H.; Su, X.; Wang, Y.; Liu, J.; Wang, C.; Zhang, J.; Wei, L. Object detection based on global-local saliency constraint in aerial images. *Remote Sens.* 2020, 12, 1435. [CrossRef]

44. Pan, X.; Ren, Y.; Sheng, K.; Dong, W.; Yuan, H.; Guo, X.; Ma, C.; Xu, C. Dynamic refinement network for oriented and densely packed object detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11204–11213. [CrossRef]

45. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the 2014 European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Volume 9912, pp. 483–499. [CrossRef]

46. Li, C.; Xu, C.; Cui, Z.; Wang, D.; Zhang, T.; Yang, J. Feature-Attentioned Object Detection in Remote Sensing Imagery. In Proceedings of the 2019 IEEE International Conference on Image Processing, Taipei, Taiwan, 22–25 September 2019; pp. 3886–3890. [CrossRef]

47. Liu, Z.; Yuan, L.; Weng, L.; Yang, Y. A High Resolution Optical Satellite Image Dataset for Ship Recognition and Some New Baselines. In Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 24–26 February 2017; pp. 324–331. [CrossRef]

48. 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. [CrossRef]

49. Liu, Y.; Zhang, S.; Jin, L.; Xie, L.; Wu, Y.; Wang, Z. Omnidirectional scene text detection with sequential-free box discretization. In Proceedings of the IJCAI Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 3052–3058. [CrossRef]