Improving the Detection of Small Oriented Objects in Aerial Images

Chandler Timm C. Doloriel and Rhandley D. Cajote
Electrical and Electronics Engineering Institute
University of the Philippines, Diliman, Philippines
{chandler.timm.doloriel, rhandley.cajote}@eee.upd.edu.ph

Abstract

Small oriented objects that represent tiny pixel-area in large-scale aerial images are difficult to detect due to their size and orientation. Existing oriented aerial detectors have shown promising results but are mainly focused on orientation modeling with less regard to the size of the objects. In this work, we proposed a method to accurately detect small oriented objects in aerial images by enhancing the classification and regression tasks of the oriented object detection model. We designed the Attention-Points Network consisting of two losses: Guided-Attention Loss (GALoss) and Box-Points Loss (BPLoss). GALoss uses an instance segmentation mask as ground-truth to learn the attention features needed to improve the detection of small objects. These attention features are then used to predict box points for BPLoss, which determines the points' position relative to the target oriented bounding box. Experimental results show the effectiveness of our Attention-Points Network on a standard oriented aerial dataset with small object instances (DOTA-v1.5) and on a maritime-related dataset (HRSC2016). The code is publicly available[1].

1. Introduction

Object detection is a valuable technique for understanding objects in an image, describing both what and where these objects are with the goal of identifying the location enclosed in bounding boxes. The usual method is to use a rectangle bounding box with no angle orientation, also called a horizontal bounding box (HBB). To enclose an object inside an HBB, the model should be able to accurately locate the object and identify its class. There are many use-cases of using HBB, specific examples are in applications for vehicle tracking[3, 32], face recognition[19, 30], maritime detection [28, 29] etc., even the objects in aerial images can be detected using an HBB. However, it is very ineffective to detect oriented aerial objects using this method, objects cannot be precisely localized, more noise and background will be enclosed that can lead to misdetection. Therefore, an object detector that can produce an oriented bounding box (OBB) is needed to detect oriented aerial objects.

Oriented aerial detection is a popular research topic in computer vision in the past years [6, 7, 11, 12, 16, 21, 33–35, 37–40, 44]. Existing methods have designed effective OBB detectors that can accurately enclose oriented objects. These methods vary from refining features [11, 37, 39], proposal extraction [7, 12, 16, 21], orientation alignment [16, 35], and regression loss design [6, 38, 40]. However, despite being very effective in detecting oriented objects, more research is needed to detect small oriented objects in aerial images.

Objects in aerial images vary greatly in size, orientation, or surroundings. Existing methods have used DOTA-v1.0 [33] to benchmark their performance in oriented object detection. However, DOTA-v1.0 is not known to contain small and complex instances. To properly benchmark our method in dataset with small oriented objects, we used DOTA-v1.5 [33].

In this work, we proposed the Attention-Points Network to detect small oriented objects in aerial images. We used attention mechanism to gather the important features of an object, which increases the model's awareness especially on hard-to-identify objects such as small and complex instances. Furthermore, the attention features are used in our designed regression loss to predict box points and score them based on their relative position to the target OBB.

In Attention-Points Network, we designed two loss functions: Guided-Attention Loss (GALoss) and Box-Points Loss (BPLoss). GALoss compares the attention features to target features that can be obtained using the instance segmentation masks. However, these masks are not easy to annotate due to the irregular shapes of aerial objects. Instead, we used coarse-level masks that only need the bounding box coordinates for annotation. Meanwhile, BPLoss is calculated by scoring the box points based on their relative position to the target OBB. We measure the relative position of the box points to the target OBB using a kernel derived

[1] https://github.com/chandlerbing65nm/APDetection.

from the sigmoid function and compute for the IoU-based loss.

To verify our work, we conducted experiments on the standard oriented aerial dataset, DOTA [33]. We chose the version of this dataset that contains very small instances (less than 10 pixels), DOTA-v1.5. We also used Oriented RCNN [34] as the baseline, and R-50-FPN [14] as backbone. Results show the effectiveness of our Attention-Points Network on a standard oriented aerial dataset with small object instances (DOTA-v1.5) and on a maritime-related dataset (HRSC2016).

The contributions of this paper are summarized as follows:

1. We proposed the Attention-Points Network to improve the detection of small oriented objects in aerial images. This network uses two losses: Guided-Attention Loss (GALoss) and Box-Points Loss (BPLoss). GALoss uses attention features to improve the detection of small objects and BPLoss is used to score the predicted box-points based on their relative position to the target OBB.

2. We compared our method to other existing OBB detectors. Experimental results show the effectiveness of our Attention-Points Network on a standard oriented aerial dataset with small object instances (DOTA-v1.5) and on a maritime-related dataset (HRSC2016).

3. We conducted an ablation experiment to evaluate our designed loss functions, GALoss and BPLoss, and compared them to the baseline architecture. Our results showed that each loss function contributes to the overall performance without lagging behind the baseline.

2. Related Work

In this section, we discuss the different approaches to detecting objects using a bounding box along with the following classification: generating proposal and regression loss design. With these, we further describe the methods in oriented object detection and lastly, we look at the details of attention mechanism and its use for oriented object detection in aerial images.

2.1. Generating Region Proposals

Generating region proposal uses an additional network to predict the location and class of objects. In [27], a segmentation map is applied to an image to discriminate the objects with the background and rejects the overlapping proposals with low objectness scores. Then, the intersection-over-union (IoU) between the ground-truth and prediction is computed with different thresholds. Usually, IoU$\geq$0.5 will be considered as an object class while IoU$<$0.5 is background. Finally, a convolutional neural network (CNN) is used to classify and localize the objects.

Current multi-stage methods tackle the issue of proposal generation. A region proposal network from [25] is designed to share the convolution layers with the feature extractor to minimize the cost, that creates sets of RoIs by dictating the model where to look. It scans every location in the extracted features to assess whether further processing is needed in a certain region and uses k anchor boxes with two scores representing whether there's an object or not at each location.

2.1.1 Oriented Proposals

To represent the rotation of an object for detection is to use anchors that rely on an angle parameter [7, 11, 12, 21, 34, 35, 37]. Early method that generate proposals with fifty-four anchors with different scales, ratios, and angles [21] obtained good performance in detecting objects that are arbitrary-oriented, but a large number of anchors causes computational complexity and memory overhead. The transformation from horizontal to rotated RoI [7] was seen as a solution to reduce the number of generated anchors since the angle parameter is not introduced in generating proposals. However, the transformation network is also heavy and complex because it involves fully connected layers and alignment operations during the learning of RoI's. Another approach based on the transformation network [7] used a rotation equivariant feature extractor [12] to draw out rotation-invariant features for region proposals. It warps and aligns the rotated region-of-interests in its correct orientation dimension through feature interpolation. However, it did not reduce the computational cost of the transformation network and the rotation equivariant backbone is computationally complex.

The key to computational bottleneck is the design of a more efficient architecture [34], this is the motivation to improve the previous oriented object detectors. To realize this, two-stage detection frameworks should generate high-quality proposals while quickly detecting objects in a cost-efficient manner [34]. In this paper, we used the network from [34] because of its efficient architecture in the proposal generation and further improve its detection using loss function design.

2.2. Regression Loss Design

A bounding box is predicted with a regression loss that gives the error between the ground-truth and prediction. Regression losses can be divided into two categories: L1-type and IoU-based loss. An example of an L1-type loss is the

smooth L1 loss, also known as Huber loss, given as:

$$loss(x, y) = \begin{cases} 0.5(x-y)^2, if|x-y| < 1 \\ |x-y| - 0.5, \text{otherwise} \end{cases} \quad (1)$$

Smooth L1 loss is less sensitive to outliers and also prevents exploding gradients. If the absolute loss is greater than one, the loss function is not squared and avoids high-value losses hence preventing exploding gradients. However, this loss is uncorrelated with the metric used in object detection. A low loss value does not always correspond to a high metric. Thus, the IoU-based loss is designed to combine the regression loss and the metric, it is given as:

$$\mathcal{L}_{IoU} = 1 - IoU \quad (2)$$

Using (2) is simple but the IoU cannot be computed when there is no overlapping area between two bounding boxes and it cannot be used in oriented object detection because the resulting function becomes undifferentiable, meaning the gradients cannot be backpropagated to enable network training.

2.2.1 Oriented IoU-based Loss

There are three known IoU-based loss in oriented object detection in aerial images. The first calculates the polygon distance of the ground-truth and prediction. It partly circumvents the need for a differentiable IoU-based loss by combining with the smooth L1 loss [39]. However, since the IoU-based loss is undifferentiable, the gradient direction is still dominated by the smooth L1 loss so the metric cannot be regarded as consistent. The second converts the ground-truth and prediction box into a 2-D Gaussian distribution and calculates the loss function through Wasserstein distance [38] and Kullback-Leibler divergence [40]. It approximates the resulting IoU-based loss to obtain a differentiable function so that it can produce useful gradients. The issue is complexity, the conversion of a bounding box to a gaussian distribution and the distance calculation using Wasserstein and Kullback-Leibler divergence are complicated and adds significant overhead in the network. The third calculates the IoU-based loss directly by accumulating the contribution of the overlapping pixels of the ground-truth and prediction box [6]. The function used is the normal distance between the pixels and the OBB center which is simple to implement but it cannot accurately represent the target object's importance since each pixel has the same level of attention. In this paper, we designed an IoU-based loss by predicting box-points from attention features. These box-points are then scored based on their relative position to the target OBB.

2.3. Attention Mechanism

Convolutional neural network (CNN) is a type of attention mechanism in computer vision that uses a filter to process the input features and calculates the non-linearity using an activation function. An example of object detection model that used CNN as attention mechanism is from [23], the work highlights the occluded objects that are detected by the region proposal network, improving the detection of occluded objects.

The disadvantage of using CNN as an attention mechanism is that the filter sizes are limited, it can usually take 3x3 or 5x5 but the attention features become coarser when we increase the filter size. Thus, CNN's can only take the attention in the local space of a filter and farther features are ignored.

2.3.1 Self-Attention

Another type of attention mechanism is self-attention, used in natural language processing (NLP) [1] to solve the problem of recognizing long sentences. In machine translation, to predict the next word is to look at the previous words, but this can be a bottleneck if the sentence is too long because it will lose the information from the previous words. Thus, self-attention searches the whole sentence, both previous and succeeding words, and analyzes the context to predict the next word. It relates different positions of a word in a sentence in order to obtain richer information.

Self-attention has three elements: Queries (Q), Keys (K), and Values (V). At the start, the input sentence is transformed into a vector that represents the three elements, which calculates the attention scores that measures how much attention to put in a word from a certain location. To compute this, the Q and K of the word is multiplied using the dot product and normalized to make the gradients stable. Then, the result is compared with V to highlight the words to focus and disregard irrelevant words. The steps above are formulated by (3).

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3)$$

As self-attention had become effective in NLP, it also gained popularity in computer vision. In some cases, they completely replaced CNN's in image classification tasks. The architecture that used self-attention is the Vision Transformer (ViT) [45]. ViT transforms the input image into a series of patches and follows the same computation for Q, K, and V, then directly predicts the class label for the image. The self-attention in ViT makes it possible to embed information globally across an image.

The downside of self-attention is the computational cost because of its function to get the attention information glob-

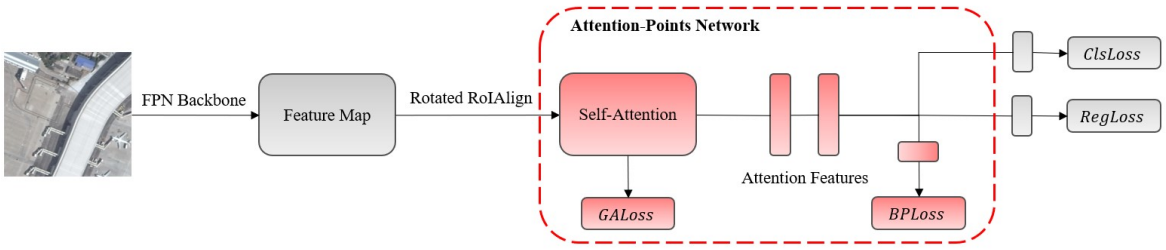

Figure 1: Architecture of Attention-Points Network.

ally, all features are used for computation compared to CNN that only used the local space. This can be mitigated by using small input image but that also limits the application of the model [31]. To reduce computational cost, [46] swapped the positions of Q and V because the dot-product of K and V would result in a smaller dimension compared to using Q. It is equivalent to the self-attention but the dot-product of vectors is used differently.

We used the concept of efficient self-attention to gather the global features and detect small oriented objects in aerial images. Furthermore, we designed a loss function that can refine the attention features by comparing them to the segmentation masks of objects.

3. Attention-Points Network

We present the details of our proposed small oriented object detector with Attention-Points Network (APN) with two new loss functions: Guided-Attention Loss (GALoss) and Box-Points Loss (BPLoss). The baseline is from Oriented RCNN [34] and we placed APN after the rotated RoIAlign, the architecture is shown in Fig. 1. It is a two-stage detector consisting of feature extraction in the first stage and prediction in the second stage. We used ResNet [14] as the backbone that produce five levels of features with each level going to the feature pyramid network (FPN) [17] for refined feature extraction. The features are inputs to the region proposal network (RPN) that generates proposals in various scales and ratios that tells the detector where the objects might be. These proposals are extracted and transformed into features by rotated RoIAlign operation and then used as input to the self-attention as region-of-interests (RoI).

We used feature size of 7x7 for each RoI to use as input to the self-attention and refine it using the ***GALoss***. The attention features are used to predict box-points that are scored based on their relative position to the target OBB using ***BPLoss***.

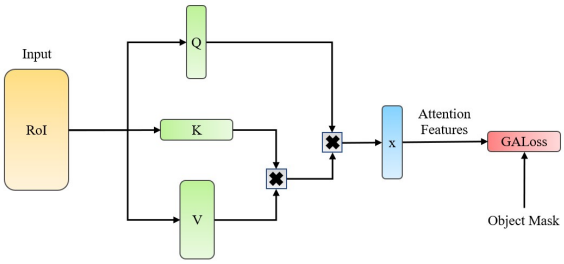

Figure 2: Illustration of Guided-Attention Loss. Input RoI is transformed into three vectors Queries (Q), Keys (K), and Values (V), then processed by a self-attention network to obtain attention features (x) that are compared to object masks using Guided-Attention Loss.

Algorithm 1 Guided-Attention Loss ($GALoss$)

Require: RoI and Mask
Ensure: $GALoss$ value
$x \leftarrow SA(RoI)$ ▷ SA is self-attention
$y \leftarrow Mask$
$GALoss \leftarrow -\frac{1}{N}\sum_{j=1}^{N}[y_j \log(x_j)+(1-y_j)\log(1-x_j)]$

3.1. Guided-Attention Loss (*GALoss*)

To make sure that we highlight the object in every RoI, we used a loss function that compares the attention features and object masks. These masks are obtained by converting the bounding box into instance segmentation of the object. We got this idea from [23], but instead of using CNN to produce attention features, we used self-attention to get the global context of objects from RoI's.

First, RoI's are processed by self-attention to obtain rich attention features, then we used binary cross-entropy to compare the similarity between the features and masks. Through training, the attention features will learn the object masks and start to focus on the foreground which will result into having more information than the input RoI. Using these features, it will improve the detection of small objects in aerial images and boost the performance on complex in-

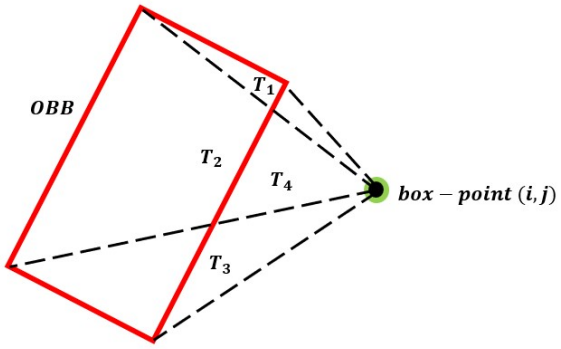

Figure 3: General idea of Box-Points Loss. T_1, T_2, T_3, T_4 are the triangles formed when the edges of the OBB are connected with the box-point at (i, j).

stances. The computation of GALoss is given in Algo. 1 and shown in Fig. 2.

3.2. Box-Points Loss (*BPLoss*)

The *BPLoss* is a function where we calculate the distance between an OBB and box-point located at ***(i,j)***, illustrated in Fig. 3. We compute the relative position of a box-point (inside or outside the box) as follows:

$$\delta(BP_{i,j}|OBB) = \begin{cases} 1, & \sum_{n=1}^{4} Area_{T_n} \leq Area_{OBB} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

As given in (4), the *BP* is the box-point and $Area_{OBB}$ is the area of the OBB. If the sum of the areas of the triangles is less than or equal to the $Area_{OBB}$, this means the *BP* is inside the OBB, otherwise it is outside. Equation (4) is a non-differentiable function, meaning we cannot have useful gradients during training, so we designed a kernel function that can approximate (4), given by:

$$\delta(BP_{i,j}|OBB) = \frac{2}{1 + e^{k\frac{\sum_{n=1}^{4} Area_{T_n} - Area_{OBB}}{Area_{OBB}}}} \tag{5}$$

Finally, to compute for the $BPLoss$, we subtract one by the sum of contributions of each kernel averaged by the total number of points. This is given by (6).

$$BPLoss = 1 - \frac{\sum_{n=1}^{N} \delta(BP_n|OBB)}{N} \tag{6}$$

The calculation of BPLoss is similar to PIoU Loss [6]. In PIoU Loss, the distance of pixels and OBB center is computed. They used OBB of both the target and prediction while in BPLoss we only used the target OBB and did not have to convert the coordinates into bounding box center format since we only need the vertices. Furthermore, our distance calculation between the box-points and OBB is done through the difference of areas which is different from the PIoU Loss that used the euclidean distance of pixels.

3.3. Evaluation

The parameter we used to compute the detection score on small oriented objects is the mean average precision (mAP). To calculate the mAP, we need to know the area of intersection over the area of union (also called as IoU) between the ground-truth and prediction boxes. We can set different IoU thresholds to get the true positives (TP) and false positives (FP) of our predictions. If the IoU is greater than the threshold, the prediction is TP, otherwise, it is FP. With TP and FP, we can calculate the precision score, which is the number of true positives (TP) over the sum of all positive predictions (TP + FP) and its average is the weighted mean at every threshold. Finally, mAP is the average precision (AP) of each class (i) averaged over the total number of classes (N) given by:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{7}$$

We used PASCAL VOC 2007 (VOC07) [9] and 2012 (VOC12) [10] mAP evaluation metrics in this paper. By default, we used IoU=0.5 and VOC07 in evaluating our model but we also employ (VOC12) and other IoU thresholds accepted as standards like IoU=0.7 and IoU=0.5:0.95.

4. Experiments

To evaluate our method, we used two datasets: DOTA-v1.5 [33] for standard aerial images with small object instances and HRSC2016 [20] for maritime-related images.

4.1. Datasets

The Dataset for Object Detection in Aerial Images version-1.5 (DOTA-v1.5) [33] is the largest dataset for object detection in aerial images with oriented bounding box annotations. It contains 2806 large-size images (1/2 train, 1/6 val, and 1/3 test splits) with 403,318 instances and 16 categories including 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), Roundabout (RA), Harbor (HA), Swimming pool (SP), Helicopter (HC), and Container Crane (CC). DOTA-v1.5 contains extremely small instances (less than 10 pixels) that vary greatly in scale, orientation, and aspect ratio, that increases the difficulty of object detection.

The High-Resolution Ship Collections 2016 (HRSC2016) [20] is a maritime-related dataset that contains ships from the sea and inshore. It contains 1061 images ranging from 300×300 to 1500×900 pixels for

Single Stage																	
Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	mAP_{50}
RetinaNet$^{\alpha}$[18]	0.753	0.754	0.318	0.619	0.321	0.692	0.790	0.896	0.716	0.588	0.430	0.661	0.501	0.603	0.390	1.5e-5	0.5649
FCOS$^{\alpha}$[26]	0.786	0.725	0.443	0.595	0.562	0.640	0.780	0.894	0.714	0.733	0.495	0.664	0.557	0.632	0.447	0.094	0.6104
RSDet++$^{\alpha}$[24]	0.793	0.740	0.449	0.609	0.564	0.605	0.780	0.894	0.708	0.735	0.512	0.683	0.562	0.682	0.522	0.102	0.6218
DCL$^{\alpha}$[36]	0.803	0.743	0.442	0.620	0.502	0.721	0.788	0.892	0.740	0.670	0.454	0.690	0.561	0.640	0.551	0.092	0.6198
GWD$^{\alpha}$[38]	0.802	0.742	0.465	0.638	0.566	0.736	0.823	0.899	0.742	0.704	0.475	0.688	0.598	0.650	0.46	0.113	0.6322
BCD$^{\alpha}$[41]	0.802	0.737	0.397	0.650	0.568	0.745	0.866	0.896	0.751	0.663	0.487	0.647	0.642	0.643	0.524	0.138	0.6353
KLDα [40]	0.802	0.727	0.473	0.601	0.632	0.751	0.860	0.895	0.735	0.729	0.503	0.662	0.645	0.691	0.578	0.137	0.6517
KFIoU$^{\alpha}$[43]	0.801	0.775	0.470	0.669	0.568	0.748	0.843	0.908	0.767	0.670	0.470	0.703	0.573	0.663	0.572	0.142	0.6469
R^{2}CNN$^{\alpha}$[15]	0.803	0.787	0.479	0.625	0.656	0.713	0.863	0.897	0.762	0.762	0.497	0.675	0.634	0.731	0.584	0.156	0.6644
Multi Stage																	
MR$^{\beta}$[13]	0.7684	0.7351	0.4990	0.5780	0.5131	0.7134	0.7975	0.9046	0.7421	0.6607	0.4621	0.7061	0.6307	0.6446	0.5781	0.0942	0.6267
CMR$^{\beta}$[2]	0.6777	0.7462	0.5109	0.6344	0.5164	0.7290	0.7999	0.9035	0.7400	0.6750	0.4954	0.7285	0.6419	0.6488	0.5587	0.0302	0.6341
HTC$^{\beta}$[4]	0.7780	0.7367	0.5140	0.6399	0.5154	0.7331	0.8031	0.9048	0.7512	0.6734	0.4851	0.7063	0.6484	0.6448	0.5587	0.0515	0.6340
FR$^{\beta}$[25]	0.7189	0.7447	0.4445	0.5987	0.5128	0.6880	0.7937	0.9078	0.7738	0.6750	0.4775	0.6972	0.6122	0.6528	0.6047	0.0154	0.6200
RT$^{\beta}$[7]	0.7192	0.7607	0.5187	0.6924	0.5205	0.7518	0.8072	0.9053	0.7858	0.6826	0.4918	0.7174	0.6751	0.6553	0.6216	0.0999	0.6503
ORCNN [34]	0.8098	0.8500	0.5992	0.7960	0.6775	0.8206	0.8978	0.9088	0.7893	0.7791	0.7097	0.7617	0.8173	0.7664	0.7354	0.4709	0.7619
OURS	0.8620	0.8563	0.5914	0.8015	0.6780	0.8180	0.8989	0.9080	0.7789	0.7843	0.6977	0.7618	0.8125	0.7654	0.7536	0.4234	**0.7620**
ORCNN* [34]	0.8011	0.6736	0.4590	0.6765	0.5912	0.7428	0.8750	0.9074	0.6927	0.7519	0.4814	0.6971	0.6854	0.6636	0.5956	0.3798	0.6672
OURS*	0.8534	0.8051	0.5473	0.7489	0.6567	0.7994	0.8864	0.9067	0.7584	0.7758	0.6319	0.7264	0.7235	0.7314	0.6206	0.3041	**0.7172**

Table 1: Comparison of results on DOTA-v1.5 trainval/test and train/test* splits (ORCNN [34] is the baseline). The colors red and blue indicate the highest value of trainval/test and train/test, respectively. The α and β denote that results are obtained from AlphaRotate [42] and AerialDetection [8] libraries, respectively.

Method	mAP_{50}	mAP_{75}	$mAP_{50:95}$
Baseline [34]	0.7619	0.5089	0.4795
Ours	0.7620 (+0.01%)	0.5204 (+2.24%)	0.4824 (+0.59%)
Baseline* [34]	0.6672	0.3886	0.3800
Ours*	0.7172 (+7.49%)	0.4425 (+13.87%)	0.4260 (+12.11%)

Table 2: Comparison of results on DOTA-v1.5 trainval/test and train/test* splits with different IoU thresholds. The color red indicate the relative difference between the methods.

which train, val, and test sets have 436, 181, and 444 images, respectively. We combined the train and val sets for training and the test set for testing.

4.2. Implementation

We used Quadro RTX 8000 for training the models and OBBDetection [34], a PyTorch library that contains different sets of oriented object detection models modified from MMdetection toolbox [5] to automatically check the performance. We also based the comparison of results published in AerialDetection [8] and AlphaRotate [42] libraries.

For DOTA-v1.5, we cropped a series of 1024x1024 patches from the original images with a stride of 524 and resized the images into multiple scales, 0.5x, 1.0x, and 1.5x with random rotation from 0-90 degrees. We optimized the network training using SGD algorithm with momentum of 0.9 and weight decay of 0.0001. We used two dataset splits for training and evaluation, trainval/test and train/test. The former is trained for 36 epochs and has an initial learning rate of 0.005 with learning rate scheduling that is divided by 10 at epochs 24 and 33, while the latter is trained for 20 epochs with no learning rate scheduling.

For HRSC2016, we randomly rotated the objects during training from 0-90 degrees, resized the images into 1333x800, and trained for 180 epochs with R-50-FPN as backbone.

Method	Backbone	mAP_{50}(07)	mAP_{50}(12)
PIoU [6]	DLA-34	0.8920	-
R3Det [37]	R-101-FPN	0.8926	0.9601
DAL [22]	R-101-FPN	0.8977	-
S2ANet [11]	R-101-FPN	0.9017	0.9501
Rotated RPN [21]	R-101	0.7908	0.8564
R2CNN [15]	R-101	0.7307	0.7973
RoI Transformer [7]	R-101-FPN	0.8620	-
Gliding Vertex [35]	R-101-FPN	0.8820	-
Oriented R-CNN [34]	R-50-FPN	0.9040	0.9650
Ours	R-50-FPN	**0.9059**	**0.9789**

Table 3: Comparison of results on HRSC2016.

4.3. Comparison with other Methods

Results on DOTA-v1.5: We compared our results with other methods. As shown in Table 1, our method has a marginal increase of mAP_{50} over the baseline at trainval/test split, but obtained a large improvement of 7.5% when using the train/test split. Furthermore, the classes with the smallest instances: small vehicles and ships, have increased by 11.01% and 1.3%, respectively. The reason why our method performed marginally at trainval/test and better in train/test is because of the distribution of data in validation set. The validation set contain images with more complex instances than the training set, hence the difference in performance. Moreover, the published results of related works only showed the mAP_{50} on trainval/test, so we implemented the baseline using train/test and compared with our method. Finally, the baseline has the second highest performance in DOTA-v1.5 trainval/test that is why we chose it for comparison in train/test split.

In Table 2, we showed that our method achieved a

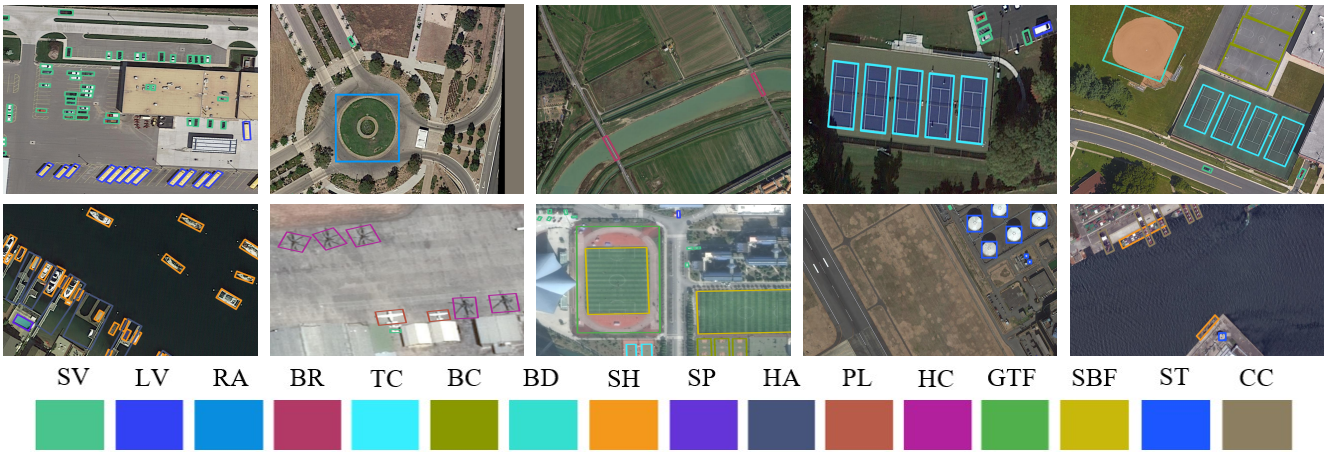

Figure 4: Visualization of detection results on DOTA-v1.5 dataset.

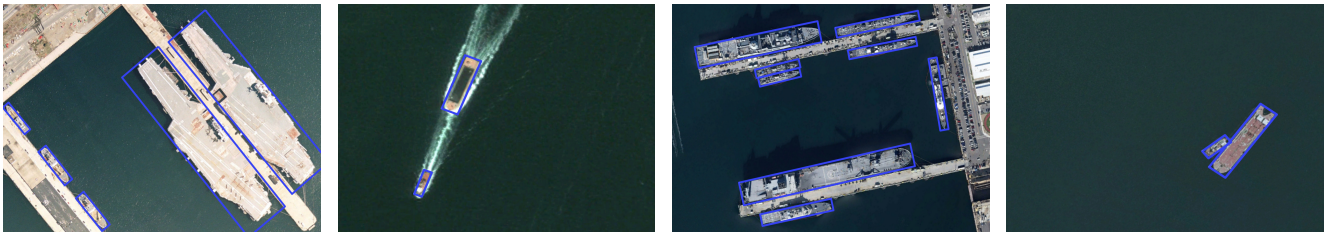

Figure 5: Visualization of detection results on HRSC2016 dataset. Ships are either in the sea or inshore.

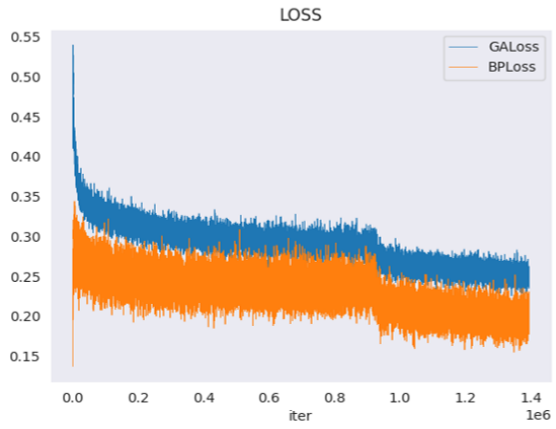

Figure 6: Learning curves of GALoss and BPLoss.

very high performance increase versus the baseline when evaluated both on DOTA-v1.5 trainval/test and train/test splits across mAP_{75} and $mAP_{50:95}$ evaluation metrics. We attributed the difference of performance of mAP_{75} and $mAP_{50:95}$ to mAP_{50} on the nature of small oriented aerial objects. These objects represents only a tiny pixel-area in an image so using IoU=0.5 is a coarse threshold and could miss objects with small instances. Thus, using finer thresholds of IoU=0.75 and IoU=0.5:0.95 are more appropriate in

Method	mAP_{75}	$mAP_{50:95}$
Case 0: Baseline [34]	0.5089	0.4795
Case 1: GALoss only	0.5144 (+1.08%)	0.4826 (+0.642%)
Case 2: BPLoss only	0.5171 (+1.61%)	0.4815 (+0.415%)
Case 3: GALoss and BPLoss	0.5204 (+2.26%)	0.4824 (+0.605%)

Table 4: Ablation experiment conducted on DOTA-v1.5 trainval/test.

metrics. Figure 4 shows sample detection results on DOTA-v1.5.

Results on HRSC2016: We also compared our method on this dataset to the baseline and other methods, shown in Table 3. We used mAP_{50} of PASCAL VOC 2007 and VOC 2012 metrics to compare the performance. As can be seen, our method achieved results in $mAP_{50}(07)$ and $mAP_{50}(12)$ that are better than the baseline. Visualization of results on HRSC is shown in Fig 5.

4.4. Loss Functions

To show that our loss functions learned during training, we plotted the learning curves over the number of iterations and showed that GALoss and BPLoss are decreasing, illustrated in Fig. 6. Although the plots are noisy, this is expected since the ground-truths used for loss calculation are coarse-level like the object masks and target OBB. Note

that the figure is not a comparison of which loss function contributed more to our method, but rather a visualization of how both loss functions learned during training.

4.5. Ablation Study

To evaluate the effectiveness of each loss functions, we conducted ablation experiment on DOTA-v1.5 dataset and compared the performance on the baseline using mAP_{75} and $mAP_{50:95}$ evaluation metrics, shown in Table 4. Cases 1 and 2 are evaluated separately and measured their performance. As can be seen in the table, both cases contributed to the overall performance without lagging behind the baseline. We can also notice that the performance of $mAP_{50:95}$ at Case 3 did not sum up when we add the results of Case 1 and Case 2. This is because $mAP_{50:95}$ is an average performance when IoU=0.5 to IoU=0.95 is calculated, IoU=0.5 is a coarse threshold not appropriate for small instances which affects the calculation of $mAP_{50:95}$. This is also the reason why we did not include it in the ablation experiment. Finally, if we look at the relative increase of Case 1 and Case 2 then take the mean, we can see that Case 3 is above the average. This shows that our designed loss functions are individually effective in the overall architecture of the Attention-Points Network.

5. Conclusion

We developed the Attention-Points Network and designed loss functions: Guided-Attention Loss (GALoss) and Box-Points Loss (BPLoss) for small oriented objects in aerial images. Results showed that our method was able to achieve better results against the baseline and other architectures on a standard oriented aerial dataset with small object instances (DOTA-v1.5) and on a maritime-related dataset (HRSC2016). Ablation experiment and learning curves of loss functions, GALoss and BPLoss, are also presented to verify the effectiveness of our method.

Acknowledgements

We would like to thank the Department of Science and Technology - Science Education Institute (DOST-SEI) for funding our research and providing graduate student scholarships through the Space Science and Technology Proliferation through University Partnerships (STeP-UP) project. A big thanks also to Dr. Rowel Atienza and the technical support of the people at the Computer Networks Laboratory (CNL) of Electrical and Electronics Engineering Institute (EEEI) of the University of the Philippines Diliman for allowing us to use their GPU's, servers and IT resources.

References

[1] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[3] Jingyuan Chen, Guanchen Ding, Yuchen Yang, Wenwei Han, Kangmin Xu, Tianyi Gao, Zhe Zhang, Wanping Ouyang, Hao Cai, and Zhenzhong Chen. Dual-modality vehicle anomaly detection via bilateral trajectory tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4016–4025, June 2021.

[4] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[6] Zhiming Chen, Kean Chen, Weiyao Lin, John See, Hui Yu, Yan Ke, and Cong Yang. PIoU Loss: Towards Accurate Oriented Object Detection in Complex Environments. In *Proceedings of the European Conference on Computer Vision*, volume 12350 LNCS, pages 195–211, 2020.

[7] Jian Ding, Nan Xue, Yang Long, Gui Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 2844–2853, 2019.

[8] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Micheal Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges, 2021.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. Pascal visual object classes challenge 2007 (voc2007) complete dataset.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[11] Jiaming Han, Jian Ding, Jie Li, Gui-song Xia, and Senior Member. Align Deep Features for Oriented Object Detection. In *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, pages 1–11, 2021.

[12] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. ReDet: A Rotation-equivariant Detector for Aerial Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2786–2795, 2021.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2017. cite arxiv:1703.06870Comment: open source; appendix on more results.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, June 2016.

[15] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: Rotational region cnn for orientation robust scene text detection. *ArXiv*, abs/1706.09579, 2017.

[16] W. Li, Y. Chen, K. Hu, and J. Zhu. Oriented reppoints for aerial object detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1819–1828, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.

[17] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

[18] Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.

[19] Yang Liu, Fei Wang, Jiankang Deng, Zhipeng Zhou, Baigui Sun, and Hao Li. Mogface: Towards a deeper appreciation on face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4093–4102, June 2022.

[20] Zikun Liu, Liu Yuan, Lubin Weng, and Yiping Yang. 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 - Volume 1: ICPRAM,*, pages 324–331. INSTICC, SciTePress, 2017.

[21] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. In *IEEE Transactions on Multimedia*, volume 20, pages 3111–3122, 2018.

[22] Qi Ming, Zhiqiang Zhou, Lingjuan Miao, Hongwei Zhang, and Linhao Li. Dynamic anchor learning for arbitrary-oriented object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2355–2363, 2021.

[23] Yanwei Pang, Jin Xie, Muhammad Haris Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Mask-guided attention network for occluded pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 4966–4974, 2019.

[24] W. Qian, Xue Yang, Silong Peng, Junchi Yan, and Xiujuan Zhang. Rsdet++: Point-based modulated loss for more accurate rotated object detection. *ArXiv*, abs/2109.11906, 2021.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 39, pages 1137–1149, 2017.

[26] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Proc. Int. Conf. Computer Vision (ICCV)*, 2019.

[27] J. R.R. Uijlings, K. E.A. Van De Sande, T. Gevers, and A. W.M. Smeulders. Selective search for object recognition. In *International Journal of Computer Vision*, volume 104, pages 154–171, 2013.

[28] Leon Amadeus Varga, Benjamin Kiefer, Martin Messmer, and Andreas Zell. Seadronessee: A maritime benchmark for detecting humans in open water. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2260–2270, January 2022.

[29] Lojze Žust and Matej Kristan. Learning maritime obstacle detection from weak annotations by scaffolding. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 955–964, January 2022.

[30] Wenjing Wang, Wenhan Yang, and Jiaying Liu. Hla-face: Joint high-low adaptation for low light face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[31] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local Neural Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[32] Minghu Wu, Yeqiang Qian, Chunxiang Wang, and Ming Yang. A multi-camera vehicle tracking system based on city-scale vehicle re-id and spatial-temporal information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4077–4086, June 2021.

[33] Gui-song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. DOTA : A Large-scale Dataset for Object Detection in Aerial Images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[34] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented R-CNN for Object Detection. In *Proceedings of 2021 International Conference on Computer Vision*, pages 3520–3529, 2021.

[35] Yongchao Xu, Mingtao Fu, Qimeng Wang, Yukang Wang, Kai Chen, Gui Song Xia, and Xiang Bai. Gliding Vertex on the Horizontal Bounding Box for Multi-Oriented Object Detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 43, pages 1452–1459. IEEE, 2021.

[36] Xue Yang, Liping Hou, Yue Zhou, Wentao Wang, and Junchi Yan. Dense label encoding for boundary discontinuity free rotation detection. *arXiv preprint arXiv:2011.09670*, 2020.

[37] Xue Yang, Junchi Yan, Ziming Feng, and Tao He. R3Det: Refined Single-Stage Detector with Feature Refinement for Rotating Object. In *Proceedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21) R3Det:*, 2021.

[38] Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking Rotated Object Detection

with Gaussian Wasserstein Distance Loss. In *Proceedings of the International Conference on Machine Learning*, 2021.

[39] Xue Yang, Jirui Yang, Junchi Yan, Yue Zhang, Tengfei Zhang, Zhi Guo, Xian Sun, and Kun Fu. SCRDet: Towards more robust detection for small, cluttered and rotated objects. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 8231–8240, 2019.

[40] Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning High-Precision Bounding Box for Rotated Object Detection via Kullback-Leibler Divergence. In *Proceedings of 2021 Conference on Neural Information Processing Systems*, pages 1–16, 2021.

[41] Xue Yang, Gefan Zhang, Xiaojiang Yang, Yue Zhou, Wentao Wang, Jin Tang, Tao He, and Junchi Yan. Detecting rotated objects as gaussian distributions and its 3-d generalization, 2022.

[42] Xue Yang, Yue Zhou, and Junchi Yan. Alpharotate: A rotation detection benchmark using tensorflow. *arXiv preprint arXiv:2111.06677*, 2021.

[43] Xue Yang, Yue Zhou, Gefan Zhang, Jirui Yang, Wentao Wang, Junchi Yan, Xiaopeng Zhang, and Qi Tian. The kfiou loss for rotated object detection, 2022.

[44] Jingru Yi, Pengxiang Wu, Bo Liu, Qiaoying Huang, Hui Qu, and Dimitris Metaxas. Oriented object detection in aerial images with box boundary-aware vectors. In *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pages 2149–2158, 2020.

[45] Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of International Conference on Learning Representations*, page s, 2021.

[46] Shen Zhuoran, Zhang Mingyuan, Zhao Haiyu, Yi Shuai, and Li Hongsheng. Efficient attention: Attention with linear complexities. In *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pages 3530–3538, 2021.