THE KFIOU LOSS FOR ROTATED OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

As a fundamental building block for visual analysis across aerial images, scene text etc., rotated object detection has established itself an emerging area, which is more general than classic horizontal object detection. Differing from the horizontal detection case whereby the alignment between final detection performance and regression loss is well kept thanks to the differentiable IoU loss, rotation detection involves the so-called SkewIoU that is undifferentiable. In this paper, we design a novel approximate SkewIoU loss based on Kalman filter, namely KFIoU loss. To avoid the standing and well-known boundary discontinuity and square-like problems, we convert the rotating bounding box into a Gaussian distribution, in line with recent Gaussian-based rotation detection works. Then we use the center loss to narrow the distance between the center of the two Gaussian distributions, followed by calculating the overlap area under the new position through Kalman filter. We qualitatively show the value consistency between KFIoU loss and the SkewIoU loss for rotation detection in different cases. We further extend our technique to the 3-D case which also suffers from the same issues as 2-D object detection. Extensive experimental results on various public datasets (2-D/3-D, aerial/text images) with different base detectors show the effectiveness of our approach. The source code will be made public available.

1 INTRODUCTION

Rotated object detection is challenging due to the difficulties of locating the arbitrary-oriented objects and separating them effectively from the background, such aerial images (Yang et al., 2018a; Ding et al., 2019; Yang et al., 2018b; 2019; 2020a; Ming et al., 2021b), scene text (Jiang et al., 2017; Zhou et al., 2017; Ma et al., 2018; Liao et al., 2018b). Though considerable progress has been made, for practical settings, there still exist challenges for rotating objects with large aspect ratio, dense distribution.

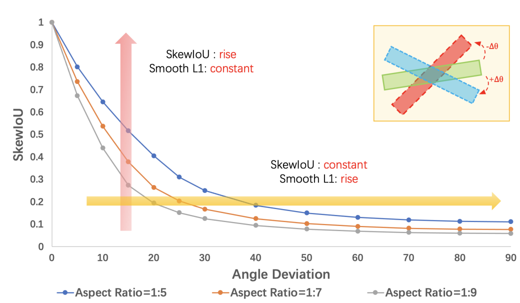


Figure 1: Inconsistency between metric and regression loss in rotated object detection.

As sketched in Fig. 1, the Skew Intersection over Union (SkewIoU) score between large aspect ratio objects is sensitive to the deviations of the object positions. This causes the negative impact of the inconsistency between metric (dominated by SkewIoU) and regression loss (e.g. l_n-norms), which is common in horizontal detection, to be further amplified in rotation detection. The red and orange arrows in Fig. 1 show the inconsistency between SkewIoU and Smooth L1 Loss. Specifically, when the angle deviation is fixed (red arrow), SkewIoU will decrease sharply as the aspect ratio increases, while the Smooth L1 loss is unchanged (mainly from the angle difference). Similarly, when SkewIoU does not change (orange arrow), Smooth L1 loss increases as the angle deviation increases. Solution for inconsistency between the metric and regression loss has been extensively discussed in horizontal detection by using IoU loss and related variants, such as GIoU (Rezatofighi et al., 2019) and DIoU (Zheng et al., 2020b). However, these solutions cannot be directly migrated to rotated object detection due to the undifferentiable of the SkewIoU. Therefore, developing a differentiable SkewIoU loss approximate calculation method is an effective alternative.

In this paper, we design a novel approximate SkewIoU loss based on Kalman filter, named KFIoU loss. Specifically, we use Gaussian modeling to convert the rotating bounding box into a Gaussian distribution, which can avoid the standing and well-known boundary discontinuity and square-like problems (Yang et al., 2019; Yang & Yan, 2020; Song et al., 2020; Qian et al., 2021; Ming et al., 2021c; Yang et al., 2021c) in rotation detection. Then we use a center loss to narrow the distance between the center of the two Gaussian distributions, fellow by calculating the overlap area under the new position through Kalman filter. **The highlights of this paper are as follows:**

1) We elaborate on the inconsistency between the final detection performance and regression loss in rotated object detection, especially for objects with large aspect ratios.

2) We propose a novel KFIoU loss for rotation detection, based on Gaussian modeling and Kalman filter, leading to direct approximate computing of the SkewIoU. Compared to the recent Gaussian based technique (Yang et al., 2021c;d) that approximate SkewIoU by learning ad-hoc nonlinear transformations, our model is simple and can be more physically coherent.

3) We also extend the Gaussian modeling and KFIoU loss from 2-D (aerial images, scene texts) to 3-D (KITTI) object detection, with notable improvement obtained. To our best knowledge, this is the first 3-D rotation detector based on Gaussian modeling in contrast to the peer works (Yang et al., 2021c;d) only focusing on 2-D rotated object detection.

4) Results on public datasets show the effectiveness of our approach. In particular, our method outperforms the recent GWD-based loss (Yang et al., 2021c) which relies on non-linear transforms to approximate the IoU loss while our method provides a more direct computational model which is scale-invariant, leading to better performance on small objects as verified in our experiments.

2 Related Work

Rotated Object Detection. Rotated object detection is an emerging direction, which attempts to extend classical horizontal detectors (Girshick, 2015; Ren et al., 2015; Lin et al., 2017a;b) to the rotation case by adopting the rotated bounding boxes. Aerial images and scene text are popular application scenarios of rotation detector. For aerial images, objects are often arbitrary-oriented and dense-distributed with large aspect ratios. To this end, ICN (Azimi et al., 2018), ROI-Transformer (Ding et al., 2019), SCRDet (Yang et al., 2019), Mask OBB (Wang et al., 2019), Gliding Vertex (Xu et al., 2020), ReDet (Han et al., 2021b) are two-stage mainstreamed approaches whose pipeline is inherited from Faster RCNN (Ren et al., 2015), while DRN (Pan et al., 2020), DAL (Ming et al., 2021d), R^{3}Det (Yang et al., 2021b), RSDet (Qian et al., 2021) and S^{2}A-Net (Han et al., 2021a) are based on single-stage methods for faster detection speed. For scene text detection, RRPN (Ma et al., 2018) employs rotated RPN to generate rotated proposals and further perform rotated bounding box regression. TextBoxes++ (Liao et al., 2018a) adopts vertex regression on SSD (Liu et al., 2016). RRD (Liao et al., 2018b) further improves TextBoxes++ by decoupling classification and bounding box regression on rotation-invariant and rotation sensitive features, respectively. The regression loss of the above algorithms are all bounding box or point based or mask-based representation, and they are rarely SkewIoU loss due to its undifferentiability.

Variants of IoU-based Loss. The inconsistency between metric and regression loss is a common problem for both horizontal detection and rotation detection. Solution for this inconsistency has been extensively discussed in horizontal detection by using IoU loss and related variants. For instance, Unitbox (Yu et al., 2016) proposes an IoU loss which regresses the four bounds of a predicted box as a whole unit. More works (Rezatofighi et al., 2019; Zheng et al., 2020b) extend the idea of Unitbox by introducing GIoU loss anf DIoU loss for bounding box regression. However, due to the undifferentiable of the SkewIoU, none of the above methods can be directly applied to rotation detection. Recently, some approximate methods for SkewIoU loss have been proposed. **Box/Polygon based:** SCRDet (Yang et al., 2019) propose IoU-Smooth L1, which partly circumvents the need for differentiable SkewIoU loss by combining IoU and Smooth L1 loss. To tackle the uncertainty of convex caused by rotation, Zheng et al. (Zheng et al., 2020a) proposes a projection operation to estimate the intersection area for both 2-D/3-D object detection. PolarMask (Xie et al., 2020) proposes Polar IoU loss that can largely ease the optimization and considerably improve the accuracy. **Pixel based:** PIoU (Chen et al., 2020) calculates the SkewIoU directly by accumulating the contribution

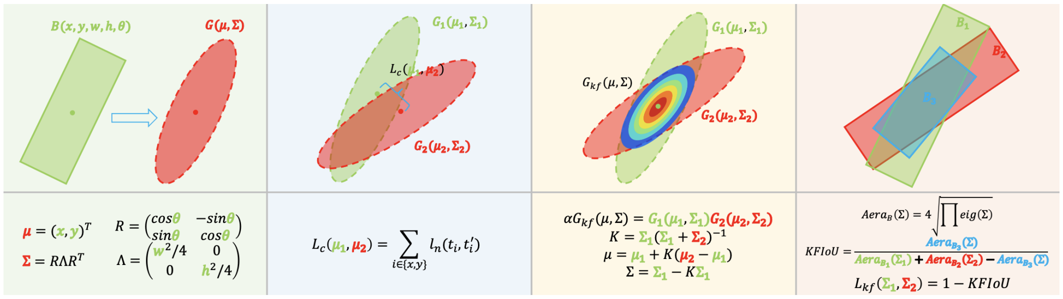


Figure 2: SkewIoU approximation process in two-dimensional space based on Kalman filter.

of interior overlapping pixels. **Gaussian based:** GWD (Yang et al., 2021c) and KLD (Yang et al., 2021d) simulate SkewIoU through Gaussian distance measurement and nonlinear transformation.

In this paper, we propose a novel regression loss based on Gaussian distribution representation, which also completes the approximation of the SkewIoU loss through Kalman filtering.

3 Background on Gaussian Distribution Modeling

In this section, we present the preliminary according to (Yang et al., 2021c), for how to convert an arbitrary-oriented 2-D/3-D bounding box to a Gaussian distribution $\mathcal{G}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

$$\boldsymbol{\Sigma} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^\top, \ \boldsymbol{\mu} = (x, y, (z))^\top \tag{1}$$

where $\mathbf{R}$ represents the rotation matrix, and $\boldsymbol{\Lambda}$ represents the diagonal matrix of eigenvalues.

For 2-D object $\mathcal{B}_{2d}(x, y, h, w, \theta)$,

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \ \boldsymbol{\Lambda} = \begin{pmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{pmatrix} \tag{2}$$

and for 3-D object $\mathcal{B}_{3d}(x, y, z, h, w, l, \theta)$,

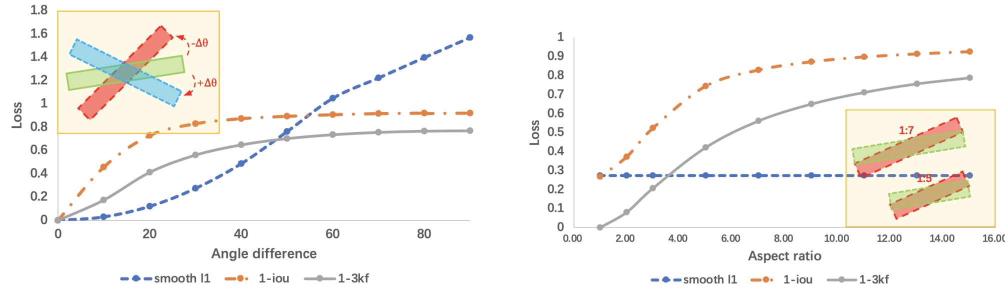
$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ \boldsymbol{\Lambda} = \begin{pmatrix} \frac{w^2}{4} & 0 & 0 \\ 0 & \frac{h^2}{4} & 0 \\ 0 & 0 & \frac{l^2}{4} \end{pmatrix} \tag{3}$$

and l, w, h represent the length, width, and height of the 3-D bounding box, respectively.

It is worth noting that the recent work GWD (Yang et al., 2021c) also belongs to the design of regression loss based on Gaussian modeling. Compared with our work, their difference is that GWD directly uses the distance metric between distributions as the final loss. Since GWD does not have scale invariance (the first term of Gaussian Wasserstein Distance is the Euclidean distance between the center points), it needs to be normalized with nonlinear transformation to ensure the normal convergence of model training. However, such an operation cannot truly achieve the consistency of metric and regression losses. In this paper, we will take another perspective to approximate the SkewIoU loss to better train the detector, which can be more physically coherent.

4 Proposed Method

In this section, we present our main approach. Fig. 2 shows the approximate process of SkewIoU loss in two-dimensional space based on Kalman filtering. Briefly, we first convert the bounding box to a Gaussian distribution as discussed in Sec. 3, and move the center points of the two Gaussian distributions to make them coincide. Then, the distribution function of the overlapping area is obtained by Kalman filtering. Finally, the obtained distribution function is inverted into a rotating bounding box to calculate the overlapping area and the IoU.



(a) Angle difference case (b) Aspect ratio case

Figure 3: Behavior comparison of different loss in different cases. Zoom in for better view.

4.1 SkewIoU Computing based on Kalman Filtering

First of all, we can easily calculate the volume of the corresponding rotating box based on its covariance, when we obtain a new Gaussian distribution, where n denotes the number of dimensions.

$$\mathcal{V}_{\mathcal{B}}(\mathbf{\Sigma}) = 2^n \sqrt{\prod eig(\mathbf{\Sigma})} = 2^n \cdot |\mathbf{\Sigma}^{\frac{1}{2}}| = 2^n \cdot |\mathbf{\Sigma}|^{\frac{1}{2}} \tag{4}$$

To obtain the final SkewIoU, calculating the area of overlap is critical. For two Gaussian distributions, we can use Kalman filter to get the distribution function of the overlapping area. Specifically:

$$\alpha \mathcal{G}_{kf}(\boldsymbol{\mu}, \mathbf{\Sigma}) = \mathcal{G}_1(\boldsymbol{\mu}_1, \mathbf{\Sigma}_1)\mathcal{G}_2(\boldsymbol{\mu}_2, \mathbf{\Sigma}_2) \tag{5}$$

Note here α is written by:

$$\alpha = \mathcal{G}_{\alpha}(\boldsymbol{\mu}_2, \mathbf{\Sigma}_1 + \mathbf{\Sigma}_2) = \frac{1}{\sqrt{\det(2\pi(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2))}} e^{-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)} \tag{6}$$

where $\boldsymbol{\mu} = \boldsymbol{\mu}_1 + \boldsymbol{K}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$, $\mathbf{\Sigma} = \mathbf{\Sigma}_1 - \boldsymbol{K}\mathbf{\Sigma}_1$, $\boldsymbol{K} = \mathbf{\Sigma}_1(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1}$.

We observe that $\mathbf{\Sigma}$ is only related to the covariance ($\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$) of the given two Gaussian distributions, which means that no matter how the two Gaussian distributions move, as long as the covariance is fixed, the area calculated by Eq. 14 will not change. This is obviously not in line with intuitive feeling: the overlapping area should be reduced when the two Gaussian distributions are far away. The main reason is $\alpha \mathcal{G}_{kf}(\boldsymbol{\mu}, \mathbf{\Sigma})$ is not a standard Gaussian distribution (the sum of probability is not 1), we cannot directly use $\mathbf{\Sigma}$ to calculate the area of the current overlap by Eq. 14 without considering α. It can be found from Eq. 6 that α is related to the distance between the center points ($\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$) of the two Gaussian distributions. Based on the above findings, we can first use a center loss $L_c(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ to narrow the distance between the center of the two Gaussian distributions, and then calculate the overlap area under the new position by Eq. 14. According to Fig. 2, we can easily calculate the KFIoU loss $L_{kf}(\mathbf{\Sigma}_1, \mathbf{\Sigma}_2)$ when we get the overlap area.

$$\text{KFIoU} = \frac{\mathcal{V}_{\mathcal{B}_3}(\mathbf{\Sigma})}{\mathcal{V}_{\mathcal{B}_1}(\mathbf{\Sigma}_1) + \mathcal{V}_{\mathcal{B}_2}(\mathbf{\Sigma}_2) - \mathcal{V}_{\mathcal{B}_3}(\mathbf{\Sigma})} \tag{7}$$

In the appendix, we prove that the upper bounds of KFIoU in n-dimensional space is $\frac{1}{2^{\frac{n}{2}}-1}$. For 2-D/3-D detection, the upper bounds are $\frac{1}{3}$ and $\frac{1}{\sqrt{32}-1}$ respectively when $n=2$ and $n=3$. We can easily stretch the range of KFIoU to $[0,1]$ by linear transformation according to the upper bound, and then compare it with IoU for consistency. It should be noted that this linear transformation is not necessary and will improve the final performance, because we pay more attention to whether the changing trends of KFIoU and IoU are consistent rather than specific values.

Fig. 3 shows the curves of three loss functions for two bounding boxes with the same center in different cases. It should be noted that we have expanded KFIoU by 3 times so that its value range is $[0,1]$ like SkewIoU. Case 1 (left) depicts the relation between angle difference and loss functions. Though they all bear monotonicity, only smooth L1 curve is convex while the others are not. Case 2 (right) shows the changes of the three loss functions under different aspect ratio conditions. It can be seen that the smooth L1 loss of the two bounding boxes are constant (mainly from the angle difference), but the IoU loss and KFIoU loss will change drastically as the aspect ratio varies. Regardless of the case, KFIoU loss can maintain a similar trend to IoU loss.

4.2 The Proposed KFIoU Loss

We take 2-D object detection as the main example for notation brevity. We use the one-stage detector RetinaNet (Lin et al., 2017b) as the baseline. Rotated rectangle is represented by five parameters (x, y, w, h, θ). First, we need to clarify that the network has not changed the output of the original regression branch, that is, it is not directly predicting the parameters of the Gaussian distribution. The whole training process of detector is as follows: i) predict offset $(t_x^*, t_y^*, t_w^*, t_h^*, t_\theta^*)$; ii) decode prediction box; iii) convert prediction box and target ground-truth into Gaussian distribution; iv) calculate L_c and L_{kf} of two Gaussian distributions. Therefore, the inference time remains unchanged.

The regression equation of (x, y, w, h) is as follows:

$$\begin{aligned} t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a \\ t_w &= \log(w/w_a), t_h = \log(h/h_a) \\ t_x^* &= (x^* - x_a)/w_a, t_y^* = (y^* - y_a)/h_a \\ t_w^* &= \log(w^*/w_a), t_h^* = \log(h^*/h_a) \end{aligned} \tag{8}$$

where x, y, w, h denote the box's center coordinates, width, height and angle, respectively. Variables x, x_a, x^* are for the ground-truth box, anchor box, and predicted box (likewise for y, w, h).

As for the regression equation of θ, we use two forms as the baseline to be compared:

i) Direct regression, marked as **Reg.** **(**$\Delta\theta$**)**. The model directly predicts the angle offset t_θ^*:

$$\begin{aligned} t_\theta =& (\theta - \theta_a) \cdot \pi/180 \\ t_\theta^* =& (\theta^* - \theta_a) \cdot \pi/180 \end{aligned} \tag{9}$$

ii) Indirect regression, marked as **Reg.*** **(**$\sin\theta$**,** $\cos\theta$**)**. The model predicts two vectors ($t_{\sin\theta}^*$ and $t_{\cos\theta}^*$) to match the two targets from the ground truth ($t_{\sin\theta}$ and $t_{\cos\theta}$):

$$\begin{aligned} t_{\sin\theta} =& \sin(\theta \cdot \pi/180), t_{\cos\theta} = \cos(\theta \cdot \pi/180) \\ t_{\sin\theta}^* =& \sin(\theta^* \cdot \pi/180), t_{\cos\theta}^* = \cos(\theta^* \cdot \pi/180) \end{aligned} \tag{10}$$

To ensure that $t_{\sin\theta}^{*2} + t_{\cos\theta}^{*2} = 1$ is satisfied, we will perform the following normalization processing:

$$t_{\sin\theta}^* = \frac{t_{\sin\theta}^*}{\sqrt{t_{\sin\theta}^{*2} + t_{\cos\theta}^{*2}}}, \quad t_{\cos\theta}^* = \frac{t_{\cos\theta}^*}{\sqrt{t_{\sin\theta}^{*2} + t_{\cos\theta}^{*2}}} \tag{11}$$

Indirect regression is a simpler way to avoid boundary discontinuity problem (Yang et al., 2019; Yang & Yan, 2020; Song et al., 2020; Ming et al., 2021c; Yang et al., 2021c). The multi-task loss is:

$$L_{total} = \lambda_1 \sum_{n=1}^{N_{pos}} L_{reg}(b_n, gt_n) + \frac{\lambda_2}{N} \sum_{n=1}^{N} L_{cls}(p_n, t_n) \tag{12}$$

where N and N_{pos} indicates the number of all anchors and the number of positive anchors. b_n denotes the n-th predicted bounding box, gt_n is the n-th target ground-truth. t_n represents the label of n-th object, p_n is the n-th probability distribution of various classes calculated by sigmoid function. λ_1, λ_2 control the trade-off and are set to $\{0.01, 1\}$ by default. The classification loss L_{cls} is set as the focal loss (Lin et al., 2017b). The regression loss $L_{reg} = L_c + L_{kf}$, where

$$L_c(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \sum_{i \in (x,y)} l_n(t_i, t_i^{'}), \quad L_{kf}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \quad f(\text{KFIoU}) \tag{13}$$

and $f(\cdot)$ represents the loss concerning KFIoU, such as $-\ln(\text{KFIoU} + \epsilon)$, $1 - \text{KFIoU}$, $e^{1-\text{KFIoU}} - 1$.

5 Experiments

5.1 2-D Datasets and Implementation Details

Aerial image dataset: DOTA (Xia et al., 2018) is one of the largest dataset for oriented object detection in aerial images with three released versions: DOTA-v1.0, DOTA-v1.5 and DOTA-v2.0.

Table 1: Ablation study of different KFIoU loss forms with different detectors on DOTA-v1.0.

Method	Smooth L1	$-\ln(\textbf{KFIoU}+\epsilon)$	$1-\textbf{KFIoU}$	$e^{1-\textbf{KFIoU}}-1$	$e^{1-3\textbf{KFIoU}}-1$	$-\ln(3\textbf{KFIoU}+\epsilon)$
RetinaNet	65.73	69.80 **(+4.07)**	70.19 **(+4.46)**	**70.64** **(+4.91)**	69.64 **(+3.91)**	–
R^{3}Det	70.66	**72.28** **(+1.62)**	71.09 **(+0.43)**	71.58 **(+0.92)**	–	71.77 **(+1.11)**

Table 2: Ablation experiments on five datasets and two detectors.

Method	Reg. Loss	MLT	UCAS-AOD	DOTA-v1.0	DOTA-v1.5	DOTA-v2.0
RetinaNet	Smooth L1	48.42	94.56	65.73	58.87	44.16
	GWD	54.58 **(+6.16)**	95.44 **(+0.88)**	68.93 **(+3.20)**	60.03 **(+1.16)**	46.65 **(+2.49)**
	KFIoU	**55.96** **(+7.54)**	**96.13** **(+1.57)**	**70.64** **(+4.91)**	**62.71** **(+3.84)**	**48.04** **(+3.88)**
R^{3}Det	Smooth L1	–	–	70.66	62.91	48.43
	GWD	–	–	71.56 **(+0.90)**	63.22 **(+0.31)**	49.25 **(+0.82)**
	KFIoU	–	–	**72.28** **(+1.62)**	**64.69** **(+1.78)**	**50.41** **(+1.98)**

DOTA-v1.0 contains 15 common categories, 2,806 images and 188,282 instances. The proportions of the training set, validation set, and testing set in DOTA-v1.0 are 1/2, 1/6, and 1/3, respectively. In contrast, DOTA-v1.5 uses the same images as DOTA-v1.0, but extremely small instances (less than 10 pixels) are also annotated. Moreover, a new category (CC-container crane), containing 402,089 instances in total is added in this version. While DOTA-v2.0 contains 18 common categories (two new categories: AP-airport and HP-helipad), 11,268 images and 1,793,658 instances. Compared to DOTA-v1.5, it further includes the new categories. The 11,268 images in DOTA-v2.0 are split into training, validation, test-dev, and test-challenge sets. We divide the images into 600×600 subimages with an overlap of 150 pixels and scale it to 800×800, in line with the cropping protocol in literature. **UCAS-AOD** (Zhu et al., 2015) contains 1,510 aerial images of approximately $659 \times 1,280$ pixels, with two categories of 14,596 instances in total. In line with (Azimi et al., 2018; Xia et al., 2018), we randomly select 1,110 for training and 400 for testing. **HRSC2016** (Liu et al., 2017) contains images from two scenarios including ships on sea and ships close inshore. The training, validation and test set include 436, 181 and 444 images.

Scene text dataset: ICDAR2015 (Karatzas et al., 2015), **MLT** (Nayef et al., 2017) and **MSRA-TD500** (Yao et al., 2012) are commonly used for oriented scene text detection and spotting. ICDAR2015 includes 1,000 training images and 500 testing images. ICDAR2017 MLT is a multi-lingual text dataset, which includes 7,200 training images, 1,800 validation images and 9,000 testing images. MSRA-TD500 consists of 300 training images and 200 testing images.

We use Tensorflow (Abadi et al., 2016) for implementation, and all experiments are performed on a server with GeForce RTX 3090 Ti and 24G memory. Experiments are initialized by ResNet50 (He et al., 2016) by default unless otherwise specified. We perform experiments on three aerial benchmarks and two scene text benchmarks to verify the generality of our techniques. Weight decay and momentum are set 0.0001 and 0.9, respectively. We employ MomentumOptimizer over 4 GPUs with a total of 4 images per mini-batch (1 image per GPU). All the used datasets are trained by 20 epochs in total, and learning rate is reduced tenfold at 12 epochs and 16 epochs, respectively. The initial learning rates for RetinaNet is 1e-3. The number of image iterations per epoch for DOTA-v1.0, DOTA-v1.5, DOTA-v1.0, UCAS-AOD, HRSC2016, ICDAR2015, MLT and MSRA-TD500 are 54k, 64k, 80k, 5k, 10k, 10k, 10k, 10k and 5k respectively, and increase exponentially if data augmentation (i.e. random graying, flipping and rotation) and multi-scale training are enabled.

5.2 3-D Datasets and Implementation Details

KITTI (Geiger et al., 2012) contains 7,481 training and 7,518 testing samples for 3-D object detection. The training samples are generally divided into the train split (3,712 samples) and the val split (3,769 samples). The evaluation is classified into Easy, Moderate or Hard according to the object size, occlusion and truncation. All results are evaluated by the mean average precision with a rotated IoU threshold 0.7 for cars and 0.5 for pedestrian and cyclists. To evaluate the model's performance on KITTI val split, we train our model on the train set and report the results on the val set.

We use third-party tools, MMDetection3D (Chen et al., 2019), for experiments and use PointPillar (Lang et al., 2019) as the baseline, and the training schedule inherited from SECOND (Yan et al., 2018): ADAM optimizer with a cosine-shaped cyclic learning rate scheduler that spans 160 epochs.

Table 3: Results on the KITTI val split 3D detection. † indicates our own implementation.

Method	mAP	Car - 3D Detection			Ped. - 3D Detection			Cyc. - 3D Detection		
	Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars	59.50	85.90	73.88	67.98	50.17	45.11	41.09	78.66	59.51	56.02
PointPillars†	61.34	85.66	75.48	68.39	55.46	48.69	43.71	79.37	59.84	55.92
+ KFIoU	**64.98**	**86.45**	**76.49**	**74.41**	**58.11**	**54.22**	**49.53**	**82.68**	**64.23**	**60.07**

Table 4: Results on the KITTI val BEV Detection. † indicates our own implementation.

Method	mAP	Car - BEV Detection			Ped. - BEV Detection			Cyc. - BEV Detection		
	Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars	66.97	**90.14**	85.27	79.81	57.80	52.53	47.50	79.96	63.10	59.35
PointPillars†	68.16	89.89	**86.97**	79.64	61.04	54.94	49.26	81.76	62.56	60.54
+ KFIoU	**70.91**	89.59	86.81	**83.21**	**63.34**	**58.43**	**54.80**	**84.61**	**67.50**	**64.52**

Table 5: High-precision detection experiment under different regression loss. 'R', 'F' and 'G' indicate random rotation, flipping, and graying, respectively. The resolution of HRSC2016, MSRA-TD500 and ICDAR2015 are 500×500, $800 \times 1,000$ and $800 \times 1,000$.

Method	Dataset	Data Aug.	Reg. Loss	Hmean/AP$_{50}$	Hmean/AP$_{60}$	Hmean/AP$_{75}$	Hmean/AP$_{85}$	Hmean/AP$_{50:95}$
RetinaNet	HRSC2016	R+F+G	Smooth L1	84.28	74.74	48.42	12.56	47.76
			KFIoU	**84.41** (+0.13)	**82.23** (+7.49)	**58.32** (+9.90)	**18.34** (+5.78)	**51.29** (+3.53)
	MSRA-TD500	R+F	Smooth L1	70.98	62.42	36.73	12.56	37.89
			KFIoU	**76.30** (+5.32)	**69.84** (+7.42)	**47.58** (+10.85)	**19.21** (+6.65)	**44.96** (+7.07)
	ICDAR2015	F	Smooth L1	69.78	64.15	36.97	8.71	37.73
			KFIoU	**75.90** (+6.12)	**69.28** (+5.13)	**40.03** (+3.06)	**9.18** (+0.47)	**41.17** (+3.44)

The learning rate starts from 1e-4 and reaches its peak value 1e-3 at the 60 epochs, and then goes down gradually to 1e-7 in the end. In the development phase, the experiments are conducted with a single model for 3-class joint detection.

5.3 Ablation Study and Further Comparison

Ablation study of three forms of KFIoU loss on two detectors. We use two different detectors and three different KFIoU based loss functions to verify its effectiveness, as shown in Table 1. RetinaNet-based detector will have a large number of low-SkewIoU prediction bounding box in the early stage of training, and will produce very large loss after the $\log$ function, which weakens the improvement of the model. Compared with the linear function, the derivative of the $\exp$-based function will pay more attention to the training of difficult samples, so it has a higher performance, at **70.64%**. In contrast, R^{3}Det-based detector can generate high-quality prediction box at the beginning of training by adding refinement stages, so it will not suffer the same troubles as RetinaNet. Due to the same mechanism of focusing on difficult samples, $\log$ and $\exp$-based functions are both better than linear functions, and the best performance is achieved on the $\log$-based function, about **72.28%**. We also expanded KFIoU by 3 times to make its range truly consistent with the IoU loss, at $[0, 1]$. However, this consistency do not bring any additional gains, so the following experiments are all use the KFIoU before non-expansion.

Ablation study of KFIoU loss on five datasets and two detectors. Table 2 shows the performance comparison of three different regression losses on five datasets. Smooth L1 is a regression loss commonly used by detectors based on bounding box representation. In contrast, both GWD and KFIoU loss are regression losses based on Gaussian modeling, but the core of the former is the Gaussian distribution distance metric, and the latter is a SkewIoU approximation based on Kalman filtering. The Gaussian representation based regression loss is significantly better than the bounding box representation based. This is mainly due to the inherent advantages of the Gaussian distribution representation described in (Yang et al., 2021c), including immunity to boundary discontinuity, and square-like detection problem. The disadvantage of GWD is that it is not scale invariant, being not conducive to small object detection. By contrast, our KFIoU loss is scale-invariant and shows better performance on datasets containing a large number of small objects e.g. DOTA-v1.5/v2.0.

Ablation study of KFIoU loss on 3-D object detection. We generalize the KFIoU loss from 2-D to 3-D object detection, with results in Table 3 and Table 4. It shows the performance comparison

Table 6: Accuracy (%) comparison on DOTA. † and ‡ represents the large aspect ratio object and the square-like object, respectively. The bold **red** and **blue** indicate the top two performances. D_{oc} and D_{le} denotes OpenCV Definition ($\theta \in [-90^\circ, 0^\circ)$) and Long Edge Definition ($\theta \in [-90^\circ, 90^\circ)$) of RBox. 'H' and 'R' denotes the horizontal and rotating anchors, respectively.

Method	Box Def.	v1.0 tranval/test									v1.5	v2.0
		BR†	SV†	LV†	SH†	HA†	ST‡	RA‡	7-AP_{50}	AP_{50}	AP_{50}	AP_{50}
RetinaNet-H (Reg.) (2017b)	D_{oc}	42.17	65.93	51.11	72.61	53.24	78.38	62.00	60.78	65.73	58.87	44.16
RetinaNet-H (Reg.) (2017b)	D_{le}	38.31	60.48	49.77	68.29	51.28	78.60	60.02	58.11	64.17	56.10	43.06
RetinaNet-H (Reg.*) (2017b)	D_{le}	41.52	63.94	44.95	71.18	53.22	78.11	60.54	59.07	65.78	57.17	43.92
RetinaNet-R (Reg.) (2017b)	D_{oc}	34.86	**73.58**	**73.33**	**82.95**	51.03	79.08	59.57	64.91	67.25	56.50	42.04
IoU-Smooth L1 (2019)	D_{oc}	**44.32**	63.03	51.25	72.78	56.21	77.98	63.22	61.26	66.99	59.16	46.31
Modulated Loss (2021)	D_{oc}	42.92	67.92	52.91	72.67	53.64	**80.22**	58.21	61.21	66.05	57.75	45.17
Modulated Loss (2021)	Quad.	43.21	70.78	54.70	72.68	**60.99**	**79.72**	62.08	63.45	67.20	**61.42**	**46.71**
RIL (2021c)	Quad.	40.81	67.63	55.45	72.42	55.49	78.09	**64.75**	62.09	66.06	58.91	45.35
CSL (2020)	D_{le}	42.25	68.28	54.51	72.85	53.10	75.59	58.99	60.80	67.38	58.55	43.34
DCL (BCL) (2021a)	D_{le}	41.40	65.82	56.27	73.80	54.30	79.02	60.25	61.55	67.39	59.38	45.46
GWD (2021c)	D_{oc}	44.07	71.92	62.56	77.94	60.25	79.64	63.52	**65.70**	**68.93**	60.03	46.65
KFIoU (Ours)	D_{oc}	**46.30**	**72.56**	**65.61**	**78.16**	**63.68**	78.15	**66.34**	**67.26**	**70.64**	**62.71**	**48.04**

Table 7: AP of different objects on DOTA-v1.0. R-101 denotes ResNet-101 (likewise for R-50, R-152). RX-101 and H-104 denotes ResNeXt101 (Xie et al., 2017) and Hourglass-104 (Newell et al., 2016). MS indicates using multi-scale training/testing. **Red** and **blue**: top two performances.

	Method	Backbone	MS	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	AP_{50}
Single-stage	PIoU (2020)	DLA-34		80.90	69.70	24.10	60.20	38.30	64.40	64.80	**90.90**	77.20	70.40	46.50	37.10	57.10	61.90	64.00	60.50
	O^2-DNet (2020)	H-104	✓	89.31	82.14	47.33	61.21	71.32	74.03	78.62	90.76	82.23	81.36	60.93	60.17	58.21	66.98	61.03	71.04
	DAL (2021d)	R-101	✓	88.61	79.69	46.27	70.37	65.89	76.10	78.53	90.84	79.98	78.41	58.71	62.02	69.23	71.32	60.65	71.78
	P-RSDet (2020)	R-101	✓	88.58	77.83	50.44	69.29	71.10	75.79	78.66	90.88	80.10	81.71	57.92	63.03	66.30	69.77	63.13	72.30
	BBAVectors (2020)	R-101	✓	88.35	79.96	50.69	62.18	**78.43**	**78.98**	**87.94**	90.85	83.58	84.35	54.13	60.24	65.22	64.28	55.70	72.32
	DRN (2020)	H-104	✓	**89.71**	82.34	47.22	64.10	76.22	74.43	85.84	90.57	86.18	84.89	57.65	61.93	69.30	69.63	58.48	73.23
	DCL (2021a)	R-152	✓	89.10	84.13	50.15	73.57	71.48	58.13	78.00	**90.89**	86.64	**86.78**	67.97	67.25	65.63	**74.06**	67.05	74.06
	PolarDet (2021)	R-101	✓	**89.65**	**87.07**	48.14	70.97	**78.53**	**80.34**	**87.45**	90.76	85.63	**86.87**	61.64	**70.32**	71.92	73.09	67.15	76.64
	GWD (2021c)	R-152	✓	86.96	83.88	**54.36**	**77.53**	74.41	68.48	80.34	86.62	83.41	85.55	**73.47**	67.77	72.57	**75.76**	**73.40**	**76.30**
	KFIoU (Ours)	R-152		89.33	85.03	52.91	70.92	77.22	70.00	82.22	90.84	**87.74**	84.77	62.88	63.39	**75.07**	70.98	**70.14**	75.56
			✓	89.46	**85.72**	**54.94**	**80.37**	77.16	69.23	80.90	90.79	**87.79**	86.13	**73.32**	**68.11**	**75.23**	71.61	69.49	**77.35**
Two/Refine-stage	ICN (2018)	R-101	✓	81.40	74.30	47.70	70.30	64.90	67.80	70.00	90.80	79.10	78.20	53.60	62.90	67.00	64.20	50.20	68.20
	RoI-Trans. (2019)	R-101	✓	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
	SCRDet (2019)	R-101	✓	89.98	80.65	52.09	68.36	68.36	60.32	72.41	90.85	**87.94**	86.86	65.02	66.68	66.25	68.24	65.21	72.61
	CFC-Net (2021a)	R-101	✓	89.08	80.41	52.41	70.02	76.28	78.11	87.21	**90.89**	84.47	85.64	60.51	61.52	67.82	68.02	50.09	73.50
	S^2A-Net (2021a)	R-50		89.11	82.84	48.37	71.11	78.11	78.39	87.25	90.83	84.90	85.64	60.36	62.60	65.26	69.13	57.94	74.12
	Gliding Vertex (2020)	R-101		89.64	**85.00**	52.26	77.34	73.01	73.14	86.82	90.74	79.02	86.81	59.55	**70.91**	72.94	70.86	57.32	75.02
	Mask OBB (2019)	RX-101	✓	89.56	**85.95**	54.21	72.90	76.52	74.16	85.63	89.85	83.81	86.48	54.89	**69.64**	73.94	69.06	63.32	75.33
	CenterMap (2020)	R-101	✓	89.83	84.41	54.60	70.25	77.66	78.32	87.19	90.66	84.89	85.27	56.46	69.23	74.13	71.56	66.06	76.03
	FPN-CSL (2020)	R-152	✓	**90.25**	85.53	54.64	75.31	70.44	73.51	77.62	90.84	86.15	86.69	69.60	68.04	73.83	71.10	**68.93**	76.17
	ReDet (2021b)	ReR-50		88.79	82.64	53.97	74.00	78.13	**84.06**	88.04	90.89	**87.78**	85.75	61.76	60.39	75.96	68.07	63.59	76.25
	RSDet-II (2021)	R-152	✓	89.93	84.45	53.77	74.35	71.52	78.31	78.12	**91.14**	87.35	86.93	65.64	65.17	75.35	**79.74**	63.31	76.34
	R^3Det (2021b)	R-152	✓	89.80	83.77	48.11	66.77	78.76	83.27	87.84	90.82	85.38	85.51	65.67	62.68	67.53	78.56	72.62	76.47
	SCRDet++ (2020b)	R-101	✓	**90.05**	84.39	**55.44**	73.99	77.54	71.11	86.05	90.67	87.32	87.08	69.62	68.90	73.74	71.29	65.08	76.81
	DAL (2021d)	R-50	✓	89.69	83.11	55.03	71.00	78.30	81.90	**88.46**	90.89	84.97	**87.46**	64.41	65.65	76.86	72.09	64.35	76.95
	DCL (2021a)	R-152	✓	89.26	83.60	53.54	72.76	79.04	82.56	87.31	90.67	86.59	86.98	67.49	66.88	73.29	70.56	**69.99**	77.37
	GWD (2021c)	R-50	✓	88.89	83.58	**55.54**	**80.46**	76.86	83.07	86.85	89.09	83.09	86.17	**71.38**	64.93	76.21	73.23	64.39	77.58
	RIDet (2021c)	R-50	✓	89.31	80.77	54.07	76.38	**79.81**	81.99	**89.13**	90.72	83.58	**87.22**	64.42	67.56	**78.08**	**79.17**	62.07	**77.62**
	R^3Det-KFIoU (Ours)	R-50		89.04	84.04	52.98	73.00	78.69	83.60	87.61	90.79	85.97	85.47	64.77	63.29	69.18	76.38	65.63	76.70
		R-50	✓	89.60	84.76	55.31	**82.39**	**79.40**	**84.29**	88.13	90.86	87.61	87.03	**71.28**	64.74	**77.48**	79.11	66.52	**79.23**

in 3-D detection and BEV detection on KITTI val split, and significant performance improvements are also achieved. On the moderate level of 3-D detection, KFIoU loss improves PointPillars† by **3.64%**. On the moderate level of BEV detection, KFIoU loss achieves gains of **2.75%**, at **70.91%**. Fig. 4 visualizes the detection results of Smooth L1 loss-based and KFIoU loss-based detectors.

High-precision detection experiment. We compare the performance of Smooth L1 loss and KFIoU loss in high-precision detection indicators, as shown in Table 5. For HRSC2016 containing a large number of ship with large aspect ratios, KFIoU loss has a **9.90%** improvement over Smooth L1 on AP_{75}. For the scene text datasets MSRA-TD500 and ICDAR2015, KFIoU loss achieves **7.07%** and **3.44%** improvements on $\text{Hmean}_{50:95}$, reaching **44.96%** and **41.17%** respectively.

Comparison with peer methods. Methods in Table 6 are all based on the same baseline RetinaNet, and initialized by ResNet50 (He et al., 2016) without using data augmentation and multi-scale training/testing. They are trained/tested under the same environment and hyperparameters. We detail the accuracy of the seven categories, with large aspect ratio (BR, SV, LV, SH, HA) and square-like object (ST, RD), to better reflect the real-world challenges and the effectiveness of our method.

First, we conduct ablation experiments on anchor form (horizontal and rotating anchors), rotating bounding box definition form (OpenCV definition and Long Edge definition), and angle regression form (direct regression and indirect regression) based on RetinaNet. Rotating anchors provides accurate prior, which makes the model show strong performance in large aspect ratio objects (e.g.

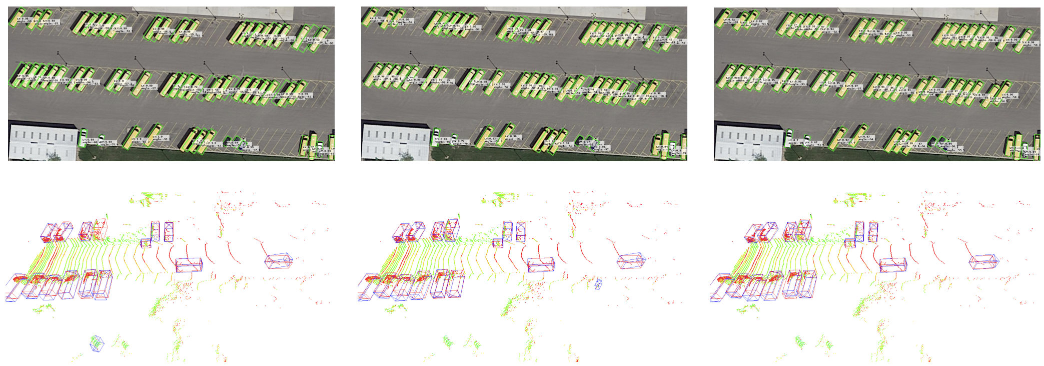


Figure 4: Comparison of the detection results between Smooth L1 loss-based (**left**), GWD-based (**middle**) and the KFIoU loss-based (**right**) detectors on DOTA (2-D) and KITTI (3-D). For 3-D object detection, red and blue box denotes ground-truth and predict bounding box, respectively.

SV, LV, SH). However, the large number of anchors makes it time-consuming. Therefore, we use horizontal anchors by default to balance accuracy and speed. OpenCV definition (D_{oc}) (Yang et al., 2019; Qian et al., 2021; Yang et al., 2021b) and Long Edge definition (D_{le}) (Yang & Yan, 2020; Yang et al., 2021a) are two popular methods for defining bounding boxes with different angles. Experiments show that D_{oc} is slightly better than D_{le} on the three versions of DOTA=v1.0/v1.5/v2.0. Angle direct regression (Reg.) always suffers from the standing boundary discontinuity problem as widely stuied recently (Yang et al., 2019; Yang & Yan, 2020; Song et al., 2020; Qian et al., 2021; Ming et al., 2021c; Yang et al., 2021c). In contrast, angle indirect regression (Reg*.) is a simpler way to avoid above problems and has an advantage in most indicators according to Table 6.

IoU-Smooth L1 partly circumvents the need for differentiable SkewIoU loss by combining IoU and Smooth L1 loss. Although IoU-Smooth L1 has achieved an improvement of **1.26%/0.29%/2.15%** from **65.73%/58.87%/44.16%** to **66.99%/59.16%/46.31%** on DOTA-v1.0/v1.5/v2.0, the gradient is still dominated by Smooth L1. Modulated Loss and RIL implement ordered and disordered quadrilateral detection respectively, and the more accurate representation makes them both have a considerable performance improvement. In particular, Modulated Loss achieves the second highest performance on DOTA-v1.5 and DOTA-v2.0. CSL and DCL convert the angle prediction from regression to classification, cleverly eliminating the boundary discontinuity problem caused by the angle periodicity. GWD and KFIoU loss are two different regression losses based on Gaussian distribution. In contrast, KFIoU loss has a more obvious performance increase due to its scale invariance and a more consistent calculation process with SkewIoU loss. Finally, KFIoU loss ranks among the top two of all methods in Table 6 on most indicators.

5.4 Comparison with the State-of-the-Art

Table 7 compares state-of-the-art detectors on DOTA-v1.0, as categorized by single-stage, two-stage, and refine-stage based methods. Data augmentation and multi-scale training/testing are used. For single-stage method, our single scale model RetinaNet-KFIoU achieves **75.56%** and outperforms most multi-scale models. For multi-scale testing, it achieves state-of-the-art accuracy **77.35%**. For two/refine-stage methods, R^{3}Det-KFIoU achieves the best AP$_{50}$: **79.23%**.

6 Conclusion

This paper first elaborates on the inconsistency between the final detection performance and regression loss in rotated object detection. To address this issue, we propose a novel approximate SkewIoU loss for rotation detection, called KFIoU loss, specifically based on the techniques of Gaussian modeling and Kalman filter. The loss is essentially scale-invariant and more physically coherent while the Gaussian distribution based loss (Yang et al., 2021c) is not. We extend our approach from 2-D to the 3-D case, leading to the first Gaussian distribution based 3-D detector. Extensive experimental results on multiple public datasets (2-D/3-D, aerial/text images) with different base detectors show the effectiveness of our approach.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.

Seyed Majid Azimi, Eleonora Vig, Reza Bahmanyar, Marco Körner, and Peter Reinartz. Towards multi-class object detection in unconstrained remote sensing imagery. In *Asian Conference on Computer Vision*, pp. 150–165. Springer, 2018.

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.

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 *European Conference on Computer Vision*, pp. 195–211. Springer, 2020.

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 Conference on Computer Vision and Pattern Recognition*, pp. 2849–2858, 2019.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. IEEE, 2012.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2021a.

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*, pp. 2786–2795, 2021b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

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 preprint arXiv:1706.09579*, 2017.

Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition*, pp. 1156–1160. IEEE, 2015.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, 2018a.

Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5909–5918, 2018b.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017b.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pp. 21–37. Springer, 2016.

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 International Conference on Pattern Recognition Applications and Methods*, volume 2, pp. 324–331, 2017.

Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

Qi Ming, Lingjuan Miao, Zhiqiang Zhou, and Yunpeng Dong. Cfc-net: A critical feature capturing network for arbitrary-oriented object detection in remote sensing images. *arXiv preprint arXiv:2101.06849*, 2021a.

Qi Ming, Lingjuan Miao, Zhiqiang Zhou, Junjie Song, and Xue Yang. Sparse label assignment for oriented object detection in aerial images. *Remote Sensing*, 13(14):2664, 2021b.

Qi Ming, Zhiqiang Zhou, Lingjuan Miao, Xue Yang, and Yunpeng Dong. Optimization for oriented object detection via representation invariance loss. *arXiv preprint arXiv:2103.11636*, 2021c.

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*, 2021d.

Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR International Conference on Document Analysis and Recognition*, volume 1, pp. 1454–1459. IEEE, 2017.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pp. 483–499. Springer, 2016.

Xingjia Pan, Yuqiang Ren, Kekai Sheng, Weiming Dong, Haolei Yuan, Xiaowei Guo, Chongyang Ma, and Changsheng Xu. Dynamic refinement network for oriented and densely packed object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11207–11216, 2020.

Wen Qian, Xue Yang, Silong Peng, Junchi Yan, and Yue Guo. Learning modulated loss for rotated object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2458–2466, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.

Qing Song, Fan Yang, Lu Yang, Chun Liu, Mengjie Hu, and Lurui Xia. Learning point-guided localization for detection in remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020.

Jinwang Wang, Jian Ding, Haowen Guo, Wensheng Cheng, Ting Pan, and Wen Yang. Mask obb: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images. *Remote Sensing*, 11(24):2930, 2019.

Jinwang Wang, Wen Yang, Heng-Chao Li, Haijian Zhang, and Gui-Song Xia. Learning center probability map for detecting objects in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

Haoran Wei, Yue Zhang, Zhonghan Chang, Hao Li, Hongqi Wang, and Xian Sun. Oriented objects as pairs of middle lines. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:268–279, 2020.

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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983, 2018.

Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 12193–12202, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500, 2017.

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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. In *European Conference on Computer Vision*, pp. 677–694. Springer, 2020.

Xue Yang, Hao Sun, Kun Fu, Jirui Yang, Xian Sun, Menglong Yan, and Zhi Guo. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1):132, 2018a.

Xue Yang, Hao Sun, Xian Sun, Menglong Yan, Zhi Guo, and Kun Fu. Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network. *IEEE Access*, 6:50839–50849, 2018b.

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*, pp. 8232–8241, 2019.

Xue Yang, Junchi Yan, and Tao He. On the arbitrary-oriented object detection: Classification based approaches revisited. *arXiv preprint arXiv:2003.05597*, 2020a.

Xue Yang, Junchi Yan, Xiaokang Yang, Jin Tang, Wenlong Liao, and Tao He. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *arXiv preprint arXiv:2004.13316*, 2020b.

Xue Yang, Liping Hou, Yue Zhou, Wentao Wang, and Junchi Yan. Dense label encoding for boundary discontinuity free rotation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 15819–15829, 2021a.

Xue Yang, Junchi Yan, Ziming Feng, and Tao He. R3det: Refined single-stage detector with feature refinement for rotating object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3163–3171, 2021b.

Xue Yang, Junchi Yan, Ming Qi, Wentao Wang, Zhang Xiaopeng, and Tian Qi. Rethinking rotated object detection with gaussian wasserstein distance loss. In *International Conference on Machine Learning*, 2021c.

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. *arXiv preprint arXiv:2106.01883*, 2021d.

Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090. IEEE, 2012.

Jingru Yi, Pengxiang Wu, Bo Liu, Qiaoying Huang, Hui Qu, and Dimitris Metaxas. Oriented object detection in aerial images with box boundary-aware vectors. *arXiv preprint arXiv:2008.07043*, 2020.

Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pp. 516–520, 2016.

Pengbo Zhao, Zhenshen Qu, Yingjia Bu, Wenming Tan, and Qiuyu Guan. Polardet: A fast, more precise detector for rotated target in aerial images. *International Journal of Remote Sensing*, 42 (15):5821–5851, 2021.

Yu Zheng, Danyang Zhang, Sinan Xie, Jiwen Lu, and Jie Zhou. Rotation-robust intersection over union for 3d object detection. In *European Conference on Computer Vision*, pp. 464–480. Springer, 2020a.

Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 12993–13000, 2020b.

Lin Zhou, Haoran Wei, Hao Li, Wenzhe Zhao, Yi Zhang, and Yue Zhang. Arbitrary-oriented object detection in remote sensing images based on polar coordinates. *IEEE Access*, 8:223373–223384, 2020.

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5551–5560, 2017.

Haigang Zhu, Xiaogang Chen, Weiqun Dai, Kun Fu, Qixiang Ye, and Jianbin Jiao. Orientation robust object detection in aerial images using deep convolutional neural network. In *2015 IEEE International Conference on Image Processing*, pp. 3735–3739. IEEE, 2015.

A PROOF OF KFIOU UPPER BOUND

For an n-dimensional Gaussian distribution, its volume is:

$$\mathcal{V} = 2^n \cdot |\mathbf{\Sigma}^{\frac{1}{2}}| = 2^n \cdot |\mathbf{\Sigma}|^{\frac{1}{2}} \tag{14}$$

For $\mathbf{\Sigma}_{kf}$, we have

$$|\mathbf{\Sigma}_{kf}| = |\mathbf{\Sigma}_1 - \mathbf{\Sigma}_1(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1}\mathbf{\Sigma}_1| = |\mathbf{\Sigma}_1(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1}\mathbf{\Sigma}_2| = \frac{|\mathbf{\Sigma}_1| \cdot |\mathbf{\Sigma}_1|}{|\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2|} \tag{15}$$

According to Minkowski's inequality:

$$|\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2|^{\frac{1}{n}} \geq |\mathbf{\Sigma}_1|^{\frac{1}{n}} + |\mathbf{\Sigma}_2|^{\frac{1}{n}} \tag{16}$$

Simultaneous mean inequalities:

$$|\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2|^{\frac{1}{n}} \geq |\mathbf{\Sigma}_1|^{\frac{1}{n}}| + |\mathbf{\Sigma}_2|^{\frac{1}{n}} \geq 2 \cdot |\mathbf{\Sigma}_1|^{\frac{1}{2n}} \cdot |\mathbf{\Sigma}_2|^{\frac{1}{2n}} \tag{17}$$

Thus:

$$\begin{aligned} \frac{|\mathbf{\Sigma}_1|^{\frac{1}{2n}} \cdot |\mathbf{\Sigma}_2|^{\frac{1}{2n}}}{|\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2|^{\frac{1}{n}}} &\leq \frac{1}{2} \\ \frac{|\mathbf{\Sigma}_1|^{\frac{1}{2}} \cdot |\mathbf{\Sigma}_2|^{\frac{1}{2}}}{|\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2|} &\leq \frac{1}{2^n} \end{aligned} \tag{18}$$

and

$$\begin{aligned} |\mathbf{\Sigma}_{kf}| =& \frac{|\mathbf{\Sigma}_1| \cdot |\mathbf{\Sigma}_1|}{|\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2|} \leq \frac{|\mathbf{\Sigma}_1|^{\frac{1}{2}} \cdot |\mathbf{\Sigma}_2|^{\frac{1}{2}}}{2^n} \\ |\mathbf{\Sigma}_{kf}|^{\frac{1}{2}} \leq& \frac{|\mathbf{\Sigma}_1|^{\frac{1}{4}} \cdot |\mathbf{\Sigma}_2|^{\frac{1}{4}}}{2^{\frac{n}{2}}} \end{aligned} \tag{19}$$

Combine the mean inequalities again:

$$|\mathbf{\Sigma}_{kf}|^{\frac{1}{2}} \leq \frac{|\mathbf{\Sigma}_1|^{\frac{1}{4}} \cdot |\mathbf{\Sigma}_2|^{\frac{1}{4}}}{2^{\frac{n}{2}}} \leq \frac{|\mathbf{\Sigma}_1|^{\frac{1}{2}} + |\mathbf{\Sigma}_2|^{\frac{1}{2}}}{2^{\frac{n}{2}+1}} \tag{20}$$

According to Eq. 14, we have

$$\mathcal{V}_{kf} \leq \frac{\mathcal{V}_1 + \mathcal{V}_2}{2^{\frac{n}{2}+1}} \tag{21}$$

Therefore, the upper bound of KFIoU is

$$\text{KFIoU} = \frac{\mathcal{V}_{kf}}{\mathcal{V}_1 + \mathcal{V}_2 - \mathcal{V}_{kf}} \leq \frac{1}{2^{\frac{n}{2}+1} - 1} \tag{22}$$

When $n = 2$ and $n = 3$, the upper bounds are $\frac{1}{3}$ and $\frac{1}{\sqrt{32}-1}$ respectively.