Supplementary Material for RFLA: Gaussian Receptive Field based Label Assignment for Tiny Object Detection

Chang Xu[1], Jinwang Wang[2], Wen Yang[1]*, Huai Yu[1], Lei Yu[1], and Gui-Song Xia[3]

[1] School of Electronic Information, Wuhan University
[2] Huawei Technologies Co., Ltd.
[3] School of Computer Science, Wuhan University
{xuchangeis,yangwen,yuhuai,ly.wd,guisong.xia}@whu.edu.cn
wangjinwang3@huawei.com

1 The approximation of effective receptive field

The size of Effective Receptive Field (ERF) varies dynamically during the training process [5], thus it requires much additional computation to get the precise ERF of each layer. As a compromise, in the design of Effective Receptive Field (ERF), we heuristically multiply a decay factor α_n to the Theoretical Receptive Field (TRF) and approximate the Effective Receptive Field (ERF). In this part, we will thoroughly analyze the effect of α_n on the performance of the proposed RFLA. According to our observations, the performance of RFLA is stable under the several proposed strategies of estimating ERF, thus the decay factor α_n can be regarded as a minor point of the whole design which will not greatly affect the performance of RFLA.

In this paper, we deploy two strategies to approximate the ERF, including setting α_n to a constant and setting α_n to a variable which shrinks linearly *w.r.t.* $1/\sqrt{n}$ [5], where n denotes the *n-th* layer of the convolution network. The results are shown in Tab. 1, it can be seen that the best performance of 21.1 AP on AI-TOD dataset can be achieved when setting α_n to the constant 1/2 or constant 1/3, while directly setting TRF (*i.e.* $\alpha_n = 1$) as prior information generates the sub-optimal performance. This observation implicitly echoes the claim that the ERF only takes up a fraction of TRF. Moreover, it further convinces that leveraging ERF as prior information for label assignment can achieve a promising result on TOD. Therefore, in all our experiments, we choose 1/2 as the default setting.

Moreover, although the best performance is achieved when setting α_n to 1/2 or 1/3, the gap between TRF or other strategies and the best performance is not significant (smaller than 1.0 AP points). Intuitively, it can be explained from the following two aspects. First, although the effective radius is not completely precise, we can assure that the centre point of Gaussian prior coincides with the

* Corresponding Author

Table 1. Performance of different decay factor α_n. Note that all models are trained on AI-TOD `train set` and tested on AI-TOD `test set`. The experiments are based on Faster R-CNN w/ RFLA. Note that the first five rows denote the constant-based strategy and the last row denotes the variable-based strategy.

α_n	AP	$AP_{0.5}$	$AP_{0.75}$	AP_{vt}	AP_t	AP_s	AP_m
1	20.8	50.8	13.2	7.8	21.1	26.6	**32.5**
1/2	**21.1**	51.6	13.1	**9.5**	21.2	26.1	31.5
1/3	**21.1**	**51.7**	**13.8**	7.5	21.5	**27.5**	31.8
1/4	20.5	50.8	12.7	8.0	21.4	24.9	30.3
1/5	20.4	49.6	13.2	9.6	21.7	24.4	29.0
$1/\sqrt{n}$	**21.1**	51.3	13.4	7.9	**21.9**	26.7	30.3

centre point of ERF. It is demonstrated that more attention is imposed on the centre region of ERF for a certain feature point [5]. The measurement of RFD gives a higher score to feature candidate which is closer to the centre region of *gt*, making the main area of the receptive field match the *gt*, which somewhat weakens the negative impact of the inaccurate effective radius of the receptive field. Second, RFD can model the relationship between any *gt* and any feature point in the whole image, when combined with HLA, at least k positive samples can be guaranteed for each instance, warranting a balanced training of objects in different scale ranges, making the performance insensitive to the chosen of α_n.

Further works are thus recommended to adaptively determine the effective radius of ERF, which makes the prior information better fits the precise ERF region.

2 Detailed derivation process of WD and KLD

2.1 Wasserstein distance

Let $P_p(M)$ denote the collection of all probability measures μ on metric space M with finite p^{th} moment. The p^{th} Wasserstein distance between two probability measure μ, ν in $P_p(M)$ is defined as:

$$W_p(\mu,\nu) := \left(\inf_{\gamma\in\Gamma(\mu,\nu)}\int_{M\times M} d(x,y)^p \, \mathrm{d}\gamma(x,y)\right)^{1/p}, \tag{s1}$$

where (M,d) is a metric space for every probability measure on M, inf denotes the infimum, $\Gamma(\mu,\nu)$ denotes the collection of all measures on $M\times M$ with marginals μ and ν on the first and second factors, respectively. Furthermore, the p^{th} Wasserstein metric can be equivalently defined as:

$$W_p(\mu,\nu) = \left(\inf \mathbb{E}\left[d(X,Y)^p\right]\right)^{1/p}, \tag{1}$$

where $\mathbb{E}[Z]$ denotes the expectation of variable Z, the infimum is taken over all joint distributions of the random variables X and Y with marginals μ and ν, respectively.

If the distance function d in Eq. 1 is Euclidean distance, 2^{nd} Wasserstein distance between μ and ν can be represented as:

$$W_2(\mu,\nu) = \left(\inf \mathbb{E}\left[\|X-Y\|_2^2\right]\right)^{1/2}. \tag{2}$$

For two Gaussian distributions in $\mathbb{R}^2$, $n_e = N_e(\boldsymbol{\mu_e}, \boldsymbol{\Sigma_e})$ and $n_g = N_g(\boldsymbol{\mu_g}, \boldsymbol{\Sigma_g})$, 2^{nd} Wasserstein distance between μ_1 and μ_2 is:

$$W_2^2(n_e,n_g) = \|\mu_e-\mu_g\|_2^2 + \mathbf{Tr}\left(\boldsymbol{\Sigma}_e+\boldsymbol{\Sigma}_g-2\left(\boldsymbol{\Sigma}_e^{1/2}\boldsymbol{\Sigma}_g\boldsymbol{\Sigma}_e^{1/2}\right)^{1/2}\right), \tag{3}$$

where $\boldsymbol{\mu}_e \in \mathbb{R}^n$ and $\boldsymbol{\mu}_g \in \mathbb{R}^n$ are the expectation, $\boldsymbol{\Sigma}_e \in \mathbb{R}^{n\times n}$ and $\boldsymbol{\Sigma}_g \in \mathbb{R}^{n\times n}$ are the covariance matrix of two Gaussian distributions. Note that for bounding box, we have $\boldsymbol{\Sigma}_e\boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}_g\boldsymbol{\Sigma}_e$, thus:

$$\mathrm{tr}\left(\left(\boldsymbol{\Sigma}_g^{1/2}\boldsymbol{\Sigma}_e\boldsymbol{\Sigma}_g^{1/2}\right)^{1/2}\right) = \mathrm{tr}\left(\left(\boldsymbol{\Sigma}_e^{1/2}\boldsymbol{\Sigma}_g\boldsymbol{\Sigma}_e^{1/2}\right)^{1/2}\right). \tag{4}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of the matrix. Furthermore, Eq. 3 can be simplified as:

$$\begin{aligned} W_2^2(n_e,n_g) &= \left\|\boldsymbol{\mu}_e-\boldsymbol{\mu}_g\right\|_2^2 + \left\|\boldsymbol{\Sigma}_e^{1/2}-\boldsymbol{\Sigma}_g^{1/2}\right\|_F^2 \\ &= \left\|\left(\left[x_n,y_n,er_n,er_n\right]^{\mathrm{T}},\left[x_g,y_g,\frac{w_g}{2},\frac{h_g}{2}\right]^{\mathrm{T}}\right)\right\|_2^2. \end{aligned} \tag{5}$$

where $\|\cdot\|_F$ is the Frobenius norm.

2.2 Kullback-Leibler divergence

Kullback-Leibler Divergence (KLD) is a classic statistical distance which measures how one probability distribution is different from another. For two continuous 2-D probability density function $p(x)$ and $q(x)$, their K-L divergence [10] is defined as:

$$D_{\mathrm{KL}}(P\|Q) = \int_{\mathbb{R}^2} p(x)\log\left(\frac{p(x)}{q(x)}\right)dx. \tag{6}$$

Similar to Wasserstein distance, KLD between two Gaussian distributions has a closed form solution. Concretely, the KLD between ERF $n_e = N_e(\boldsymbol{\mu_e}, \boldsymbol{\Sigma_e})$ and gt region $n_g = N_g(\boldsymbol{\mu_g}, \boldsymbol{\Sigma_g})$ is as follows:

$$D_{\mathrm{KL}}\left(n_e\|n_g\right) = \frac{1}{2}(\mathrm{tr}\left(\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Sigma}_e\right) + \left(\boldsymbol{\mu}_g-\boldsymbol{\mu}_e\right)^{\top}\boldsymbol{\Sigma}_g^{-1}\left(\boldsymbol{\mu}_g-\boldsymbol{\mu}_e\right) + \ln\frac{|\boldsymbol{\Sigma}_g|}{|\boldsymbol{\Sigma}_e|} - 2), \tag{7}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of the matrix. Given the 2-D Gaussian model of ERF and gt region, each item in Eq. 7 can be expressed as

$$\mathrm{tr}\left(\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Sigma}_e\right) = \frac{er_n^2}{4w_g^2} + \frac{er_n^2}{4h_g^2}, \tag{8}$$

$$\left(\boldsymbol{\mu}_g-\boldsymbol{\mu}_e\right)^{\top}\boldsymbol{\Sigma}_g^{-1}\left(\boldsymbol{\mu}_g-\boldsymbol{\mu}_e\right) = \frac{4(x_n-x_g)^2}{w_g^2} + \frac{4(y_n-y_g)^2}{h_g^2}, \tag{9}$$

$$\ln \frac{|\boldsymbol{\Sigma}_g|}{|\boldsymbol{\Sigma}_e|} = \ln \frac{4w_g^2}{er_n^2} + \ln \frac{4h_g^2}{er_n^2}, \tag{10}$$

Thus, Eq. 7 can be simplified in its closed form

$$D_{\mathrm{KL}}\left(n_e \| n_g\right) = \frac{er_n^2}{8w_g^2} + \frac{er_n^2}{8h_g^2} + \frac{2(x_n - x_g)^2}{w_g^2} + \frac{2(y_n - y_g)^2}{h_g^2} + \ln \frac{2w_g}{er_n} + \ln \frac{2h_g}{er_n} - 1. \tag{11}$$

3 Detailed information about datasets

Experiments are conducted on four datasets, including AI-TOD [8], TinyPerson [11], VisDrone2019 [2] and DOTA-v2.0 [1]. In this section, we report the detailed information about these datasets.

AI-TOD is a dataset designed for detecting tiny objects in aerial images. AI-TOD comes with 700,621 instances across 28,036 aerial images over 8 categories. The mean absolute object size in AI-TOD is only 12.8 pixels, 86% of the objects are smaller than 16×16 pixels, which makes it a representative dataset for tiny object detection.

TinyPerson is a dataset dedicated for detecting tiny person in large-scale images. TinyPerson contains 72,651 objects of one single class *person*, the images are mainly captured around seaside. The mean absolute size of TinyPerson is only 18.0 pixels.

VisDrone2019 is an UAV dataset for object detection. It is composed of 10,209 images with 10 categories. Captured in different places at different height, objects in VisDrone2019 have large scale variance and complex background, where part of them also exhibit extremely tiny scales.

DOTA-v2.0 is a large-scale dataset for object detection in aerial images. It is the follow-up version of DOTA-v1.0 [9], and it contains 1,793,658 instances over 18 categories. One of the main challenges in DOTA-v2.0 is raised by the considerable number of tiny scale object.

4 Comparison with more state-of-the-art methods

In this part, we compare the proposed RFLA with more state-of-the-art methods on VisDrone2019 and DOTA-v2.0 dataset, further verifying RFLA's capability of pushing forward detectors' TOD performance. Results are shown in Tab. 2 and Tab. 3. Notably, we change the backbone of Faster R-CNN from ResNet-50-FPN [3] to the HRNet [7] which holds a fine-grained resolution for tiny object detection. It can be observed that the overall AP of Faster R-CNN w/ RFLA is comparable to that of HRNet, and the $\mathrm{AP}_{\mathrm{vt}}$ and AP_{t} of a simple Faster R-CNN w/ RFLA surpass the strong HRNet-w32 by 4.5 and 5.2 points on VisDrone2019. Since our method is about the prior information and the definition of positive and negative samples in the training stage, no additional cost will be introduced in the inference stage. However, it is known that switching the backbone to the

Table 2. Results on VisDrone2019. The `train set` and `val set` is used for training and validation respectively. Note that HRNet-w32* denotes Faster R-CNN built on HRNet-w32 [7] backbone, we use ResNet-50-FPN as backbone for other detectors.

Method	AP	$AP_{0.5}$	$AP_{0.75}$	AP_{vt}	AP_t	AP_s	AP_m
FCOS	14.1	25.5	13.9	0.1	2.1	8.4	24.2
Faster R-CNN	22.3	38.0	23.3	0.1	6.2	20.0	33.0
Cascade R-CNN	22.5	38.5	23.1	0.5	6.8	21.4	33.6
HRNet-w32*	23.5	40.1	23.8	0.4	6.5	21.7	34.5
DetectoRS	25.7	41.7	27.0	0.5	7.6	23.4	**38.1**
FCOS w/ RFLA	$15.1^{+1.0}$	$27.3^{+1.8}$	$15.1^{+1.2}$	$0.4^{+0.3}$	$3.8^{+1.7}$	$11.4^{+3.0}$	$24.7^{+0.5}$
Faster R-CNN w/ RFLA	$23.4^{+1.1}$	$41.4^{+3.4}$	$22.7^{-0.6}$	$\mathbf{4.8}^{+4.7}$	$11.7^{+5.5}$	$20.1^{+0.1}$	$32.3^{-0.7}$
Cascade R-CNN w/ RFLA	$23.9^{+1.4}$	$40.4^{+1.9}$	$24.4^{+1.3}$	$2.9^{+2.4}$	$10.9^{+4.1}$	$20.6^{-0.8}$	$33.2^{-0.4}$
DetectoRS w/ RFLA	$\mathbf{27.4}^{+1.7}$	$\mathbf{45.3}^{+3.6}$	$\mathbf{28.1}^{+1.1}$	$4.5^{+4.0}$	$\mathbf{12.9}^{+5.3}$	$\mathbf{24.2}^{+0.8}$	$37.6^{-0.5}$

Table 3. Results on DOTA-v2.0. Models are trained on its `train set` and validated on its `val set`. Note that HRNet-w32* denotes Faster R-CNN built on HRNet-w32 backbone, we use ResNet-50-FPN as backbone for other detectors.

Method	AP	$AP_{0.5}$	$AP_{0.75}$	AP_{vt}	AP_t	AP_s	AP_m
FCOS	31.8	55.4	31.7	0.3	4.0	19.4	38.7
Faster R-CNN	35.6	59.5	37.2	0.0	7.1	28.9	42.1
HRNet-w32*	36.9	60.4	39.3	0.0	9.9	31.9	43.3
Cascade R-CNN	37.0	59.5	39.6	0.0	5.9	28.4	44.0
DetectoRS	40.8	62.6	44.2	0.0	7.0	29.9	**47.8**
FCOS w/ RFLA	$32.1^{+0.3}$	$55.6^{+0.2}$	$32.8^{+1.1}$	$0.7^{+0.7}$	$6.8^{+2.8}$	$23.5^{+4.1}$	$38.3^{-0.4}$
Faster R-CNN w/ RFLA	$36.3^{+0.7}$	$61.5^{+2.0}$	$37.5^{+0.3}$	$1.9^{+1.9}$	$\mathbf{11.7}^{+4.6}$	$31.0^{+2.1}$	$41.9^{-0.2}$
Cascade R-CNN w/ RFLA	$37.3^{+0.3}$	$60.9^{+1.4}$	$39.0^{-0.6}$	$1.7^{+1.7}$	$9.9^{+4.0}$	$29.4^{+1.0}$	$43.3^{-0.7}$
DetectoRS w/ RFLA	$\mathbf{41.3}^{+0.5}$	$\mathbf{64.2}^{+1.6}$	$\mathbf{44.4}^{+0.2}$	$\mathbf{2.1}^{+2.1}$	$10.8^{+3.8}$	$\mathbf{33.5}^{+3.6}$	$47.4^{-0.4}$

HRNet-w32 will introduce much additional computation cost, which indicates that RFLA holds both accuracy and efficiency utility.

Besides, we build the proposed RFLA on DetectoRS [6] and test its performance on VisDrone2019 and DOTA-v2.0, DetectoRS is a recently published work which holds the state-of-the-art performance on horizontal object detection tasks [4,6] among all convolution-based detectors. It is obvious that a significant improvement can be achieved when applying RFLA into the DetectoRS, particularly, the improvement in the *tiny* scale is remarkable, 5.3 points for VisDrone2019 and 3.8 points for DOTA-v2.0. The consistent and significant improvement on different detectors convinces the generality and robustness of the proposed RFLA. Nevertheless, the performance in *medium* scale suffers from a minor drop (round 0.5 points), resulting from the reduction of the number of positive samples assigned to large objects under the balanced HLA strategy. But compared to the significant improvement of tiny objects, the minor drop of large objects is trivial, which will not greatly detract from the whole method.

5 Visualization and failure cases

Here are more visualization results on VisDrone2019 and DOTA-v2.0 dataset. See Fig. 1 and Fig. 2, it can be observed that the false negative detections of

tiny objects are greatly eliminated compared with the original DetectoRS. The reduction in the number of objects missed to be detected mainly attributes to the sufficient training of tiny objects under the RFLA strategy. Surprisingly, we observe that RFLA can detect some tiny objects missed to be annotated, which means that some false positive detections (*i.e.* blue boxes) come from label noise problem.

Finally, here are two failure cases of the RFLA. First, the RFLA may generate false positive detections on the tiny object missed to be labeled, owing to the label noise. Second, the RFLA fails to make optimal label assignments and get precise predictions when objects are severely occluded. The above two failure cases can be seen from Fig. 1 and Fig. 2.

Fig. 1. Visualization results on VisDrone2019. The first row is the detection result generated by DetectoRS and the second row is the detection result of DetectoRS w/ RFLA. The green, blue and red boxes denote true positive (TP), false positive (FP) and false negative (FN) predictions respectively.

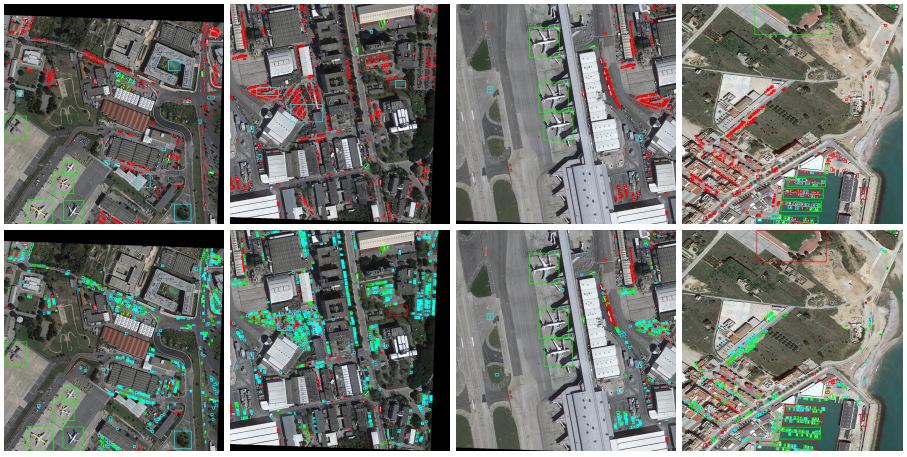

Fig. 2. Visualization results on DOTA-v2.0. The first row is the detection result generated by DetectoRS and the second row is the detection result of DetectoRS w/ RFLA. The green, blue and red boxes denote true positive (TP), false positive (FP) and false negative (FN) predictions respectively.

References

1. Ding, J., Xue, N., Xia, G.S., Bai, X., Yang, W., Yang, M.Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., et al.: Object detection in aerial images: A large-scale benchmark and challenges. IEEE Transactions on Pattern Analysis and Machine Intelligence p. in press (2021)
2. Du, D., Zhu, P., Wen, L., et al.: Visdrone-det2019: The vision meets drone object detection in image challenge results. In: IEEE International Conference on Computer Vision Workshops. pp. 213–226 (2019)
3. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125 (2017)
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)
5. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. Advances in Neural Information Processing Systems **29** (2016)
6. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)
7. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5693–5703 (June 2019)
8. Wang, J., Yang, W., Guo, H., Zhang, R., Xia, G.S.: Tiny object detection in aerial images. In: International Conference on Pattern Recognition. pp. 3791–3798 (2021)
9. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: DOTA: A large-scale dataset for object detection in aerial images. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3974–3983 (2018)
10. Yang, X., Yang, X., Yang, J., Ming, Q., Wang, W., Tian, Q., Yan, J.: Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. Advances in Neural Information Processing Systems **34** (2021)
11. Yu, X., Gong, Y., Jiang, N., Ye, Q., Han, Z.: Scale match for tiny person detection. In: IEEE Workshops on Applications of Computer Vision. pp. 1257–1265 (2020)