# Supplementary Materials for “Visible and Clear: Finding Tiny Objects in Difference Map”

Bing Cao, Haiyu Yao, Pengfei Zhu*, and Qinghua Hu

College of Intelligence and Computing, Tianjin University, Tianjin, China
Tianjin Key Lab of Machine Learning, Tianjin, China
{caobing,yaohaiyu,zhupengfei,huqinghua}@tju.edu.cn

## 1 More Experiments

### 1.1 Competing Methods

In this part, we present the competing methods compared in the experiments.
**Faster R-CNN.** Faster R-CNN [11] is one of the most classic two-stage detectors. It firstly feeds the extracted feature map into the Region Proposal Network (RPN) to extract proposals, which is used to perform classification and regression in the second stage.
**RetinaNet.** RetinaNet [8] is a one-stage detector that proposes Focal Loss to increase the weight of hard samples, alleviating the foreground-background samples imbalance problem.
**Cascade R-CNN.** Cascade R-CNN [1] significantly improves the quality of proposals through multi-stage regression with different IoU thresholds.
**Cascade RPN.** Cascade RPN [12] introduces the cascade idea into the RPN network, alleviating the misalignment issue between anchors and features.
**TridentNet.** TridentNet [7] uses dilated convolutions for multi-scale perception.
**ATSS.** ATSS [18] proposes an adaptive training sample selection method that automatically divides positive and negative training samples based on the statistical characteristics of objects.
**DyHead.** DyHead [3] applies attention mechanisms to three aspects: scale perception, spatial perception, and task perception.
**DetectoRS.** DetectoRS [10] introduces Recursive Feature Pyramid (RFP) and Switchable Atrous Convolution (SAC), significantly improving detection performance.
**RFLA.** RFLA [15] introduces a simple but effective receptive field-based label assignment strategy, enhancing the detection performance of tiny objects.
**CZ Det.** CZ Det [9] introduces a cascaded zoom-in detector that performs cropping and re-detection on regions identified as density.
**CFINet.** CFINet [16] introduces a coarse-to-fine RPN to ensure sufficient and high-quality proposals for tiny objects, and equips the conventional detection head with a feature imitation branch to facilitate the region representations of size-limited instances.

* Corresponding author

**HANet.** HANet [6] proposes a hierarchical activation method to obtain scale-specific feature subspaces by activating object features at different scales hierarchically.

**Deformable-DETR.** Deformable-DETR [19] speeds up the training convergence of transformer-based detectors and improves the detection of tiny objects by introducing a multi-scale deformable attention module.

**DINO.** DINO [17] improves over previous transformer-based detectors in performance and efficiency by using a contrastive way for denoising training, a mixed query selection method for anchor initialization, and a look forward twice scheme for box prediction.

### 1.2 Effectiveness of Reweighting

In the Difference Map Guided Feature Enhancement (DGFE) module, we perform *reweighting* along the channel dimension of difference maps for feature enhancement. To validate the effectiveness of *reweighting*, we conduct experiments with Cascade R-CNN w/ SR-TOD on the VisDrone2019 [4] dataset. Specifically, we conduct experiments with channel-wise reweighting solely on the feature maps and keep the difference maps weighted equally across the channel dimension. The results are shown in Tab. 1. The performance of solely utilizing the difference map surpasses that of using only the reweighting method by 0.5 AP. This observation underscores the substantial impact of the prior information in the difference map on enhancing the efficacy of detecting tiny objects. Furthermore, by reweighting the difference maps along the channel dimension, the performance achieves 27.3 AP, with a significant improvement of 0.4 points in $\mathrm{AP}_{0.75}$. This demonstrates that directly utilizing difference maps with spatial information alone across all channels with equal weights is not qualified for element-wise feature enhancement. In contrast, reweighting the difference maps leads to a remarkable improvement in the regression accuracy of tiny objects.

### 1.3 More Details of Difference Map

In Tab. 2, we keep all other parameters fixed and empirically show that 40/255 is the best initial value of the learnable threshold $T_{init}$. However, the overall impact of the variation in initial values is minimal, demonstrating the robustness of our method to the selection of initial threshold values. Additionally, for datasets with more complex backgrounds, the model is more sensitive to $T_{init}$, as shown in Tab. 3. The setting of the $T_{init}$ is associated with different datasets. For complex ground backgrounds, we set lower $T_{init}$, such as 4/255 for VisDrone2019 and 3/255 for AI-TOD. For scenarios like DroneSwarms where most drone instances are in the sky, we set the $T_{init}$ to 40/255.

### 1.4 More Results on DOTA.

We have conducted experiments on the DOTA dataset [14]. Tab. 4 shows our method achieves considerable performance (*e.g.*, RFLA, $\mathrm{AP}_t$: 5.6 → 6.0, $\mathrm{AP}_s$:

**Table 1:** Effectiveness of reweighting. RW denotes only using reweighting in DGFE and DM denotes only using difference map.

| RW | DM | AP | $AP_{0.5}$ | $AP_{0.75}$ | $AP_{vt}$ | $AP_t$ | $AP_s$ |
|---|---|---|---|---|---|---|---|
| ✓ | | 26.7 | 46.3 | 26.6 | 2.4 | 10.9 | 23.9 |
| | ✓ | 27.2 | **47.0** | 27.1 | **2.8** | 11.3 | 24.5 |
| ✓ | ✓ | **27.3** | 46.9 | **27.5** | 2.3 | **11.5** | **24.7** |

**Table 2:** Performance of different $T_{init}$. $T_{init}$ denotes the initial threshold. Results are on DroneSwarms.

| $T_{init}$ | AP | $AP_{0.5}$ | $AP_{0.75}$ | $AP_{vt}$ | $AP_t$ | $AP_s$ |
|---|---|---|---|---|---|---|
| 20/255 | 38.0 | 87.3 | 25.0 | 30.7 | 46.9 | 59.3 |
| 40/255 | **38.3** | 87.4 | 25.4 | 30.8 | **47.4** | **59.4** |
| 60/255 | **38.3** | **87.6** | 25.3 | 31.1 | **47.4** | 59.2 |
| 80/255 | 38.1 | 87.4 | 25.3 | 30.9 | 47.1 | 59.2 |
| 100/255 | **38.3** | 87.5 | **25.6** | 30.8 | **47.4** | 59.2 |

26.5 → 26.8). This demonstrates our robustness even in remote sensing scenes with significant variations in object scales and extreme class imbalances.

### 1.5 More Comparisons on AI-TOD.

**Additional computation.** We have evaluated the additional computation and performance of integrating our method into DetectoRS on the AI-TOD dataset in Tab. 5. Our method increases acceptable computational costs (FPS: 12.5 → 8.9) while delivering considerable improvements (AP: 14.6 → 24.0, $AP_{0.5}$: 31.8 → 54.6, $AP_{vt}$: 0 → 10.1).
**Predicted difference map.** We have extended our self-reconstructed difference map to a predictive version by using ground truth (GT). In Tab. 5, although the predicted difference map (PDM) reduced computation costs, it also affected detection performance when compared to ours and even the baseline (*e.g.*, AP: 24.0 → 13.3, FPS: 8.9 → 11.4). This further validates that our reconstructed difference map is more sensitive to tiny objects, sufficient to construct prior knowledge for more robust detection.

### 1.6 More Comparisons with FPN Variants.

We have reported more comparisons with FPN variants in Tab. 6. Please kindly note that we impose an auxiliary reconstruction head on the FPN module without altering its structure. Our method exhibits remarkable flexibility, enabling seamless integration with different FPN variants to enhance their performance effectively (*e.g.*, Recursive-FPN, AP: 26.3 → 27.2, $AP_t$: 7.5 → 11.7).

**Table 3:** Different $T_{init}$ on VisDrone2019.

| $T_{init}$ | AP | $AP_{0.5}$ | $AP_{0.75}$ | $AP_{vt}$ | $AP_t$ | $AP_s$ |
|---|---|---|---|---|---|---|
| 2/255 | **27.3** | **46.9** | **27.5** | **2.6** | 11.1 | 24.3 |
| 4/255 | **27.3** | **46.9** | **27.5** | 2.3 | **11.5** | 24.7 |
| 6/255 | 27.2 | 47.0 | 27.3 | 2.2 | 10.9 | **24.8** |
| 8/255 | 27.0 | 46.8 | 27.1 | 1.9 | 10.9 | 24.4 |
| 10/255 | 26.8 | 46.4 | 27.0 | **2.6** | 10.5 | 24.1 |

**Table 4:** Results on DOTA. The best and second results are highlighted.

| Method | AP | $AP_{0.5}$ | $AP_{0.75}$ | $AP_{vt}$ | $AP_t$ | $AP_s$ |
|---|---|---|---|---|---|---|
| CZ Det [9] | 34.6 | 56.9 | 36.2 | - | - | - |
| Cascade R-CNN [1] | 43.9 | 68.9 | **47.9** | 0.0 | 4.2 | 26.2 |
| RFLA [15] | 44.0 | 69.1 | **47.9** | **0.3** | 5.6 | 26.5 |
| Ours | **44.1** | **69.6** | 47.7 | **0.3** | **6.0** | **26.8** |

### 1.7 More Visualizations of Difference Maps on Other Datasets

We select specific images from the VisDrone2019 [4] and AI-TOD [13] datasets to visualize difference maps for images with simple and complex backgrounds, as shown in Fig. 1a and Fig. 1b. For the images with simple backgrounds, the difference maps prominently depict tiny objects such as ships and vehicles, with a significant portion of the background remaining inactive. In addition, although the outlines of structures such as houses may be discernible in images with complex backgrounds, their activation levels are subdued, while tiny objects like vehicles and pedestrians are distinctly emphasized. This accentuates the notable utility of difference maps even amidst complex backgrounds.

### 1.8 Overall Flow of SR-TOD

The overall flow of SR-TOD is shown in Alg. 1. $I_r$ denotes the reconstructed image that reconstruction head outputs. $Mean$ denotes computing the mean value along the channel dimension. $Resize$ denotes resizing $D_b$ to the same size as $P2$. $D$ denotes the difference map resulting from the subtraction of the original image $I_o$ and the reconstructed image $I_r$. $D_b$ denotes the binary difference map generated by threshold filtering. $M$ denotes the element-wise attention matrix. In summary, we construct difference map $D$ and calculate element-wise attention matrix $M$ by threshold filtering and reweighting to enhance the feature map $P2$.

**Table 5:** More comparisons on AI-TOD. PDM means predicted difference map.

| Method | AP | $AP_{0.5}$ | $AP_{0.75}$ | $AP_{vt}$ | $AP_t$ | $AP_s$ | FPS |
|---|---|---|---|---|---|---|---|
| DetectoRS [10] | 14.6 | 31.8 | 11.5 | 0.0 | 11.0 | 27.4 | 12.5 |
| DetectoRS w/ PDM | 13.3 | 28.3 | 10.6 | 0.1 | 8.9 | 25.7 | 11.4 |
| DetectoRS w/ SR-TOD | **24.0** | **54.6** | **17.1** | **10.1** | **24.8** | **29.3** | 8.9 |

**Table 6:** Comparison with FPN variants on VisDrone2019.

| FPN variants | AP | $AP_{0.5}$ | $AP_{0.75}$ | $AP_{vt}$ | $AP_t$ | $AP_s$ |
|---|---|---|---|---|---|---|
| NAS-FPN [5] | 20.4 | 35.9 | 20.1 | 0.8 | 3.6 | 14.8 |
| Recursive-FPN [10] | 26.3 | 43.9 | 26.9 | 0.1 | 7.5 | 23.3 |
| Recursive-FPN w/ SR-TOD | 27.2 | **47.1** | 27.2 | **2.4** | **11.7** | 24.2 |
| FPN w/ SR-TOD | **27.3** | 46.9 | **27.5** | 2.3 | 11.5 | **24.7** |

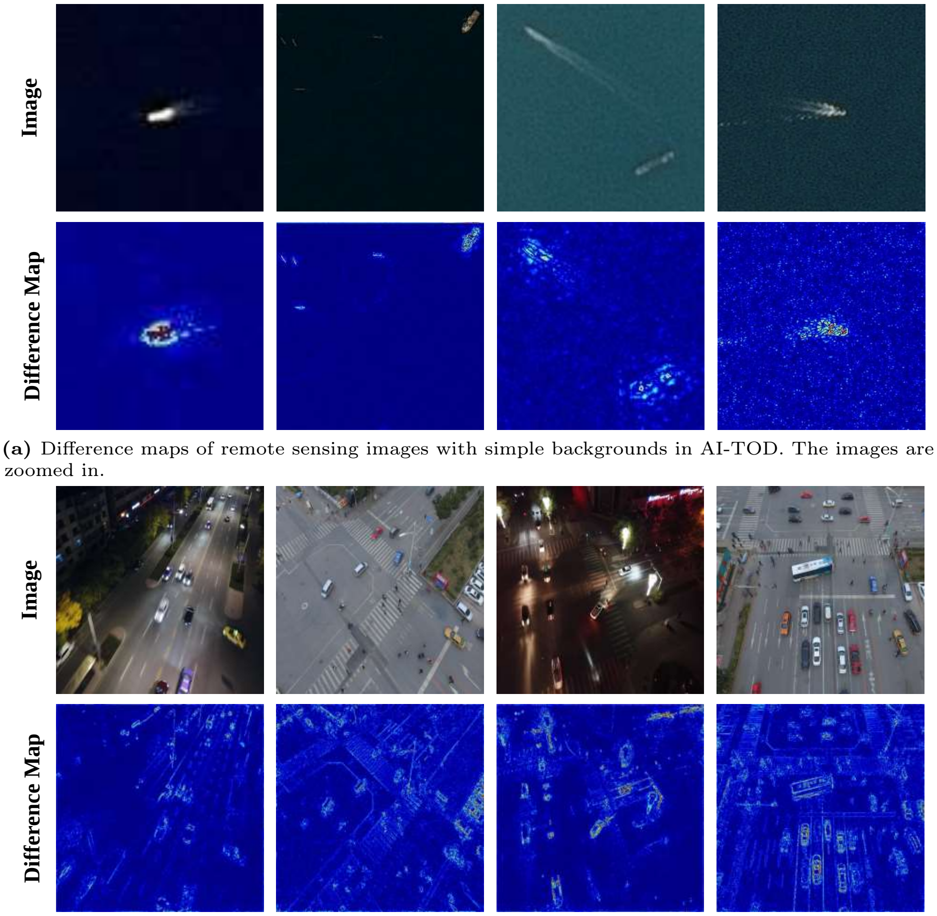
**Algorithm 1** Algorithm of SR-TOD

```
Input:
    The feature map $P2$;
    The original image input $I_o$;
    The threshold $t$;
    The Sign function $Sign$;
    The resize function $Resize$;
    The mean function $Mean$;
    The reconstruction head $RH$;
    The reweighting operation $Reweighting$;
Output:
    The enhanced feature map $P2'$
$I_r \leftarrow RH(I_o)$
$D \leftarrow Mean(|I_r - I_o|)$
$D_b \leftarrow (Sign(D - t) + 1) \times 0.5$
$M \leftarrow Reweighting(P2) \otimes (Resize(D_b) + 1)$
$P2' \leftarrow M \otimes P2$
return $P2'$
```

## 2 More Details of DroneSwarms

### 2.1 Overview of DroneSwarms

Typically, drones operate far from the surveillance apparatus, situated at considerable distances and altitudes, resulting in drone objects that are very tiny and lack clarity. Therefore, the anti-UAV scenario is an important application scenario suitable for tiny object detection. Furthermore, current tiny object detection datasets commonly include many medium and large objects, with average object sizes all above 12.8 pixels [2,15]. In order to construct a dataset consisting almost entirely of a large number of tiny objects, we propose a object detection dataset

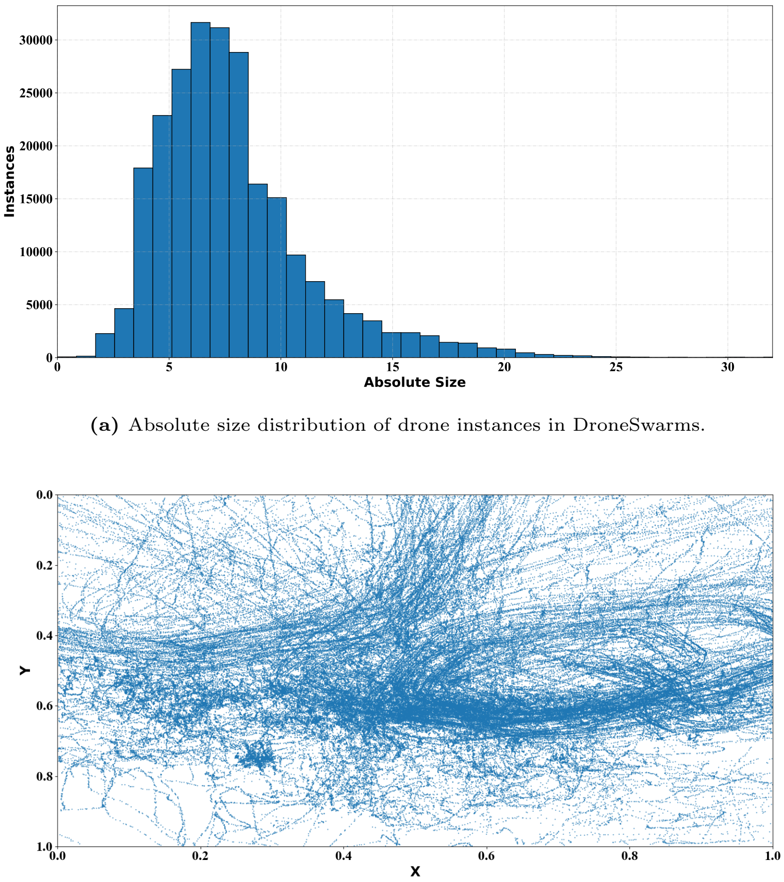


**(a)** Difference maps of remote sensing images with simple backgrounds in AI-TOD. The images are zoomed in.

**(b)** Difference maps of drone aerial images with complex backgrounds in VisDrone2019.

**Fig. 1:** Visualizations of difference maps on VisDrone2019 and AI-TOD.

with the smallest average size currently for anti-UAV, named DroneSwarms. DroneSwarms consists of 9,109 images and 242,218 annotated UAV instances, with 2,532 used for testing and 6,577 used for training. On average, each image contains 26.59 drone instances. The images are in the size of 1920 × 1080, manually labeled with high precision. DroneSwarms encompasses a variety of outdoor settings such as urban environments, mountainous terrain, and skies, among others. The drones are tiny and dispersed across the entirety of the image. Therefore, DroneSwarms can be used to comprehensively evaluate methods for tiny object detection.



**(a)** Absolute size distribution of drone instances in DroneSwarms.

**(b)** Spatial distribution of drone instances in DroneSwarms.

**Fig. 2:** Visualization of the statistical characteristics of drone instances in DroneSwarms.

## 2.2 Absolute Size Distribution

As shown in Fig. 2a, almost all instances in DroneSwarms are tiny objects. In DroneSwarms, there are 241,249 tiny objects with a pixel area below $32 \times 32$, accounting for approximately 99.60%. The average absolute size of objects in DroneSwarms is only 7.9 pixels. The information content contained in these tiny drones is minimal, and imaging blur often occurs, making feature extraction very challenging. Differentiating from the background is difficult when these tiny drones appear in front of ground and building backgrounds. Additionally, due to lighting angles, tiny drones under cloud cover are also hard to distinguish from the background in terms of color. The relative pixel areas of these tiny drones compared to the entire image are extremely small, resulting in a severe imbalance between foreground and background. Therefore, we propose the DroneSwarms as a challenging dataset for tiny object detection.

### 2.3 Spatial Distribution

In the DroneSwarms dataset, the spatial distribution of drone objects is represented using scatter plots, as illustrated in Fig. 2b. Significantly, the drone objects do not exhibit concentration around the image center; instead, their positions are extensively scattered. On average, each image in the DroneSwarms dataset contains 26.59 drone objects. Detecting densely packed tiny drones in neighboring regions poses a significant challenge within this context.

### 2.4 Image Background

DroneSwarms encompasses a variety of outdoor settings such as urban environments, mountainous terrain, and skies, among others. Furthermore, DroneSwarms also encompasses different lighting conditions based on various times and weather conditions, such as clear skies, overcast weather, dusk, and backlighting. Moreover, the drones showcase a variety of postures like takeoff, hovering, cruising, and landing, allowing them to appear in different sizes and angles within the same background.

## References

1. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: CVPR. pp. 6154–6162 (2018)
2. Cheng, G., Yuan, X., Yao, X., Yan, K., Zeng, Q., Xie, X., Han, J.: Towards large-scale small object detection: Survey and benchmarks. IEEE TPAMI **45**(11), 13467–13488 (2023)
3. Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L.: Dynamic head: Unifying object detection heads with attentions. In: CVPR. pp. 7373–7382 (2021)
4. Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., et al.: Visdrone-det2019: The vision meets drone object detection in image challenge results. In: Proceedings of the IEEE/CVF international conference on computer vision workshops. pp. 213–226 (2019)
5. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: CVPR. pp. 7036–7045 (2019)
6. Guo, G., Chen, P., Yu, X., Han, Z., Ye, Q., Gao, S.: Save the tiny, save the all: hierarchical activation network for tiny object detection. IEEE TCSVT (2023)
7. Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. In: ICCV. pp. 6054–6063 (2019)
8. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
9. Meethal, A., Granger, E., Pedersoli, M.: Cascaded zoom-in detector for high resolution aerial images. In: CVPRW. pp. 2045–2054 (2023)
10. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: CVPR. pp. 10213–10224 (2021)

11. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. NeurIPS **28**, 91–99 (2015)
12. Vu, T., Jang, H., Pham, T.X., Yoo, C.: Cascade rpn: Delving into high-quality region proposal network with adaptive convolution. NeurIPS **32**, 1432–1442 (2019)
13. Wang, J., Yang, W., Guo, H., Zhang, R., Xia, G.S.: Tiny object detection in aerial images. In: ICPR. pp. 3791–3798 (2021)
14. 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: CVPR. pp. 3974–3983 (2018)
15. Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S.: Rfla: Gaussian receptive field based label assignment for tiny object detection. In: ECCV. pp. 526–543 (2022)
16. Yuan, X., Cheng, G., Yan, K., Zeng, Q., Han, J.: Small object detection via coarse-to-fine proposal generation and imitation learning. In: ICCV. pp. 6317–6327 (2023)
17. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In: ICLR (2022)
18. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR. pp. 9759–9768 (2020)
19. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. ICLR (2021)