SPAR-Det: Supplementary Materials

Seungchan Kwon Gyuil Lim Youngjoon Han
Soongsil University, South Korea
{kwonsc36, gyuilLim, young}@soongsil.ac.kr

Abstract

In the supplementary materials, we provide implementation details used in our experiments, ablation studies, and qualitative visualizations of object detection predictions. The supplementary content is organized as follows:

- ***Implementation details**: Supplementary information on implementation settings not covered in the main paper, along with a detailed description of the Gating Network algorithm used in SPAR-Det.*
- ***Ablation studies & Discussion**: Analysis of the effects of varying hyperparameters in our model, demonstrating how segmentation guidance enhances small object detection while revealing its dependency on the quality of pretrained features and the compatibility between segmentation and detection backbones. We further discuss limitations of external backbone reliance and propose a unified architecture as a future direction.*
- ***Experimental results**: Quantitative evaluation of SPAR-Det on the VisDrone [2] and AI-TOD [5] benchmarks.*

1. Implementation details

1.1. Gating network

Our MoE-Head performs expert selection at the RoI level. To facilitate this, we employ a lightweight *Gating Network* that operates on each RoI feature $x \in \mathbb{R}^{B\times C\times H\times W}$, where B is the number of RoIs in the batch. The network computes routing weights $\mathbf{g} \in \mathbb{R}^{B\times E}$ for E experts by extracting compact statistical descriptors from each RoI's internal attention map.

The complete computation pipeline of this gating network is summarized in **Algorithm 1**. This implementation directly corresponds to the MoE-Head routing mechanism introduced in Section 4.3 of the main paper, where the gating scores $\mathbf{g}$ are used to weight the outputs of selected experts for each RoI.

Step 1: Attention map computation

For each RoI feature x, we first compute a spatial attention map $A \in \mathbb{R}^{B\times(H\times W)}$ by averaging over the channel dimension and applying softmax:

$$A = \mathrm{Softmax}(\mathrm{Mean}_c(x)).$$

Step 2: Statistical descriptor extraction

From the attention map, we compute eight scalar statistics that capture the saliency and geometric properties of the RoI:

- **Entropy:** $-\sum A \log A$
  Measures the uncertainty of the attention distribution. Low entropy indicates sharp focus on a few pixels, while high entropy suggests that attention is spread across the region.
- **Maximum:** $\max(A)$
  Captures the strongest saliency response, reflecting the most confident location within the RoI.
- **Variance:** $\mathrm{Var}(A)$
  Represents the variability of attention values. A larger variance indicates stronger contrast between salient and non-salient pixels.
- **Region mean:** Mean of the top 10% attention values
  Provides a robust measure of the intensity of the most salient sub-region, less sensitive to noise than the global mean.
- **Centroid:** Attention-weighted coordinates (μ_x, μ_y) in normalized $[-1, 1]$ space
  Summarizes the spatial location of saliency, reflecting position-dependent priors such as the center bias commonly observed in Small object images.
- **Contrast:** Mean absolute deviation from the spatial mean of x
  Estimates the degree of foreground–background separation, where higher contrast indicates stronger objectness.
- **Area ratio:** Fraction of pixels in A greater than its mean
  Indicates the relative size and compactness of the salient object, helping distinguish small sparse objects from large diffuse ones.

These statistics are concatenated into an 8D vector $s \in \mathbb{R}^8$ for each RoI. Each of them is grounded in classical saliency priors validated in previous studies: entropy for distribution sharpness, maximum and variance for discriminability, centroid for center bias, contrast for figure–ground

Algorithm 1: RoI-Level Small-Object-Aware Gating (SAG)

```
Input: RoI feature $x \in \mathbb{R}^{B \times C \times H \times W}$
Output: Routing weights $\mathbf{g} \in \mathbb{R}^{B \times E}$, selected
        indices $k$
$A \leftarrow \mathrm{Softmax}(\mathrm{Mean}_c(x))$;          // [B, HW]
  attention per RoI
Compute per-RoI statistics:;
    entropy $\leftarrow -\sum A \log A$ ;
    max $\leftarrow \max(A)$ ;
    variance $\leftarrow \mathrm{Var}(A)$ ;
    regionMean $\leftarrow \mathrm{Mean}(\mathrm{Top}_{10\%}(A))$ ;
    $(\mu_x, \mu_y) \leftarrow \sum A \cdot (x, y)$ over normalized
      coordinates;
    contrast $\leftarrow \mathrm{Mean}|x - \overline{x}|$ ;
    areaRatio $\leftarrow \#(A > \mathrm{Mean}(A))/(HW)$ ;
    $s \leftarrow$ concat$([\cdot]) \in \mathbb{R}^8$ ;
$\ell \leftarrow$ MLP(LayerNorm$(s)$) ;            // expert
  logits
$k \leftarrow$ Top-$k(\ell)$ ;
$\mathbf{g} \leftarrow \mathrm{Softmax}(\ell \odot \mathbf{1}_k)$ ;     // top-k gating
return $\mathbf{g}, k$
```

separation, and area ratio for object size and compactness. Thus, they provide interpretable and task-relevant cues for routing decisions in the MoE-Head.

Step 3: Gating weight computation

Each vector s is normalized and passed through a two-layer MLP to produce expert logits:

$$\ell = \text{MLP}(\text{LayerNorm}(s)) \in \mathbb{R}^E.$$

We retain only the top-k logits and zero out the rest using a binary mask $\mathbf{1}_k$, then apply softmax:

$$\mathbf{g} = \mathrm{Softmax}(\ell \odot \mathbf{1}_k).$$

2. Ablation Studies

We conduct comprehensive ablation experiments to analyze the individual contributions of core components and design choices in our SPAR-Det framework. Specifically, we investigate (1) the number of selected experts (k) in the MoE-Head, (2) the effect of using fused feature maps with segmentation backbones, and (3) the sensitivity of the model to the weighting factor α in our geometric prior loss, (4) the threshold τ for binarizing the supervision map, and (5) the analysis of segmentation backbones.

2.1. Effect of Top-K in MoE-Head

The number of experts selected by the gating network (*i.e.*, the value of k in Top-k routing) is a critical hyperparameter that balances routing flexibility and expert specialization. To analyze this trade-off, we evaluate SPAR-Det with varying values of $k \in \{1, 2, 4, 6, 8\}$, while keeping the total number of experts fixed at 8.

Table 1. Effect of Top-k expert selection on VisDrone. The best performance for each metric is highlighted in **bold**.

Top-k	AP$_s$	AP$_m$	AP$_l$	AP	mAP
1	22.1	39.7	41.7	29.8	51.7
2	22.3	40.3	**43.4**	30.3	52.3
4	**23.2**	**40.6**	42.3	**30.8**	**53.4**
6	22.7	40.3	41.2	30.4	52.8
8	22.3	40.6	38.2	30.0	51.9

As shown in Table 1, the model achieves the best overall performance when $k = 4$. This suggests that moderate sparsity in expert selection leads to better expert diversity and more effective specialization. In contrast, using fewer experts (e.g., $k = 1$ or $k = 2$) limits the model's flexibility in routing, while selecting all experts ($k = 8$) results in oversmoothing and reduced discrimination, likely due to the lack of specialization pressure.

2.2. Effectiveness of Fused Feature Maps

To examine how segmentation features can be effectively integrated into the detection pipeline, we compare four different fusion strategies. Recalling the original formulation of the CAHF module in Equation 1:

$$\mathbf{F}^l_{\text{CAHF}} = \mathbf{C}^l(\mathbf{F}^l_f) \odot \mathbf{S}^l(\mathbf{F}^l_f) \odot \mathbf{F}^l_{\text{obj}} \quad (1)$$

we evaluate the following variants. Note that all fusion variants from (ii) to (V) incorporate both channel and spatial attention modules (i.e., $\mathbf{C}^l(\cdot)$ and $\mathbf{S}^l(\cdot)$ in Eq. 1) when applied to the fused feature. In contrast, the simple baseline (i) employs feature concatenation *without any attention module*, serving as a control to isolate the effect of attention-based fusion.

- **(i) Fuse-only**: This variant removes both spatial and channel attention modules in CAHF and uses only the fused feature $\mathbf{F}^l_f$ as the output. This experiment aims to isolate the effect of feature fusion without modulation by learned attention weights.
- **(ii) Seg-only (Segmentation-as-detection-weight)**: Instead of constructing a fused feature $\mathbf{F}^l_f$, we directly replace the object detection feature $\mathbf{F}^l_{\text{obj}}$ with the segmentation feature $\mathbf{F}^l_{\text{seg}}$ during training. The segmentation backbone is initialized with weights pretrained on the object detection task, enabling semantic representations to guide the detection stream. The resulting feature is then used in place of $\mathbf{F}^l_f$ as input to the CAHF module.

- **(iii) Fuse-input (Concat+Conv fusion)**: The object detection feature $\mathbf{F}^l_{\text{obj}}$ and segmentation feature $\mathbf{F}^l_{\text{seg}}$, both in $\mathbb{R}^{H^l \times W^l \times C^l}$, are concatenated along the channel axis. 1×1 convolution is applied to compress the channel dimension back to C^l, producing the fused feature $\mathbf{F}^l_f$. This fused feature is used as input to the attention modules, while the original $\mathbf{F}^l_{\text{obj}}$ remains in Equation 1.
- **(iv) Fuse-replace**: The fused feature $\mathbf{F}^l_f$ is used to replace $\mathbf{F}^l_{\text{obj}}$ entirely in Equation 1, such that the final CAHF output is fully computed from the fused representation.
- **(v) SPAR-Det (Full-fusion)**: SPAR-Det forms a fused feature $\mathbf{F}^l_f$ by reducing both $\mathbf{F}^l_{\text{seg}}$ and $\mathbf{F}^l_{\text{obj}}$ to $C^l/2$ channels and concatenating them. The fused map is then used to compute attention weights, which modulate $\mathbf{F}^l_{\text{obj}}$ in the CAHF module as defined in Equation 1.

Table 2. Comparison of segmentation fusion strategies on VisDrone. Only *Fuse-only* disables CAHF attention modules, isolating the effect of feature merging alone. All other strategies incorporate CAHF-based attention mechanisms.

Strategy	AP_s	AP_m	AP_l	AP	AP_{50}
Without CAHF attention module					
Fuse-only	19.6	35.7	37.8	26.7	46.9
With CAHF attention module					
Seg-only	22.5	39.8	40.3	30.0	51.7
Fuse-input	23.3	40.5	39.1	30.3	51.9
Fuse-replace	22.6	**40.6**	**45.3**	30.3	52.3
Full-fusion	**23.2**	**40.6**	42.3	**30.8**	**53.4**

As shown in Table 2, the strategies that incorporate CAHF attention modules generally yield better performance than Fuse-only baseline, suggesting that attention-based modulation may provide a more effective way to integrate segmentation features into the detection pipeline. Among the ablated variants, *Fuse-replace* achieves the best performance, particularly on large objects, suggesting that directly using the fused representation in the final output leads to better recognition of semantically rich, large-scale objects. In contrast, *Fuse-only*, which disables all attention mechanisms and directly uses the fused feature as output, results in the poorest performance. This demonstrates that naive feature fusion alone is insufficient, and validates the importance of attention-guided modulation, as implemented in SPAR-Det.

In contrast, SPAR-Det, which employs full fusion by modulating $\mathbf{F}^l_{\text{obj}}$ with attention weights derived from the fused feature, achieves superior performance on small and medium objects.

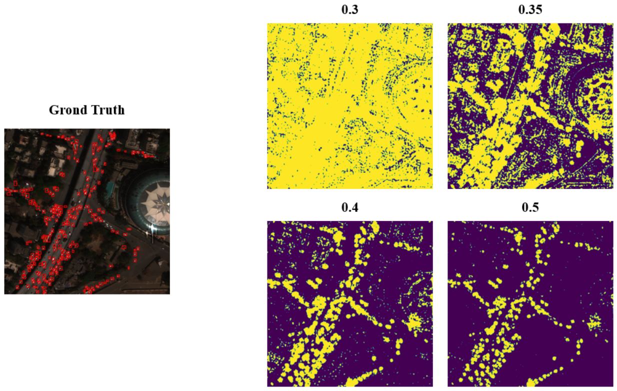

Figure 1. Visual comparison of binarized supervision maps $\hat{\mathbf{G}}_{\text{target}}$ generated with different threshold values

2.3. Effect of the Balance Weight in Geometric Prior Loss

The hyperparameter α controls the balance between geometric prior supervision and the standard detection loss. We conduct experiments varying $\alpha \in \{0.6, 0.7, 0.8, 0.9\}$ to identify its impact on performance.

Specifically, α determines the contribution of semantic versus geometric signals when constructing the supervision target $\mathbf{G}_{\text{target}}$ for geometric prior learning. Increasing α amplifies the influence of the segmentation-derived semantic map, while decreasing it gives greater weight to the Gaussian-based geometric prior $\text{H}_{\text{gaussian}}$.

Table 3. Effect of α on AI-TOD.

α	AP_s	AP_m	AP_l	AP	AP_{50}
0.6	22.3	39.6	41.5	29.7	51.5
0.7	22.0	40.4	39.3	29.7	51.8
0.8	**23.2**	**40.6**	**45.3**	**30.3**	**52.3**
0.9	22.4	39.9	40.1	30.0	52.2

As shown in Table 3, setting $\alpha = 0.8$ achieves the best performance across all metrics, suggesting that this value provides the most effective trade-off between semantic and geometric cues. When α is too small, the supervision target $\mathbf{G}_{\text{target}}$ is dominated by the geometric prior $\text{H}_{\text{gaussian}}$, potentially neglecting high-level semantic context. Conversely, overly large α values lead to excessive reliance on segmentation features, which may not always provide precise localization cues. Therefore, $\alpha = 0.8$ is empirically optimal for balancing both types of information in SPAR-Det.

2.4. Analysis of Threshold Sensitivity.

Figure 1 and Table 4 show visual and quantitative analyses of how varying the threshold τ, which is used to binarize

Table 4. Impact of the threshold factor τ on the binarized supervision map $\hat{\mathbf{G}}_{\text{target}}$ for the AI-TOD dataset.

τ	AP	mAP	AP_{vt}	AP_{t}	AP_{s}	AP_{m}
0.30	30.7	60.6	15.8	30.8	34.6	40.5
0.35	31.2	61.2	16.0	31.5	35.7	41.0
0.40	30.1	59.7	14.1	30.7	34.6	40.3
0.45	30.7	60.5	16.7	30.8	35.6	41.5

the supervision map $\hat{\mathbf{G}}_{\text{target}}$ derived from H_{gaussian}, affects performance. The threshold is varied from 0.30 to 0.45 in increments of 0.05.

As shown in Figure 1, setting τ too low causes the binary map to over-expand, including background regions along with small objects. In contrast, overly high thresholds result in insufficient coverage, missing essential object cues. Table 4 shows that the best performance is achieved at $\tau = 0.35$, with 31.2 AP and 61.2 mAP. This setting provides a good trade-off between semantic and geometric priors, indicating that $\tau = 0.35$ is optimal for binarizing $\hat{\mathbf{G}}_{\text{target}}$ in SPAR-Det.

2.5. Analysis of of Segmentation

In this section, we analyze the impact of different segmentation backbones on the performance of SPAR-Det. For a fair comparison, the object detection backbone is fixed to ResNet-50, and SPAR-Det employs UNet++ [8] as its default segmentation backbone. We further evaluate the effect of replacing UNet++ with alternative segmentation architectures, including UNet [4], DeepLabV3+ [1], and SegFormer [6].

The results are summarized in Table 5. Although DeepLabV3+ achieves the highest IoU, its detection performance remains relatively poor. This can be attributed to its architecture, which preserves the stage-3 and stage-4 feature maps of ResNet-50 without modification. As a result, the features learned by the detection backbone are not compatible with segmentation features, limiting the ability to extract effective representations for small object detection. Segformer achieves the lowest IoU among all segmentation backbones, and despite using the same detection backbone, the drop in IoU is accompanied by a proportional decrease in overall detection performance. In contrast, UNet++ and UNet, which directly utilize ResNet-50, achieve the most stable and superior performance when combined with SPAR-Det under the Cascade R-CNN framework.

In summary, the results indicate that alignment between segmentation and detection backbones is critical, followed by the inherent capability of the segmentation model itself, to achieve optimal detection performance. Among the tested backbones, applying UNet++ as the segmentation backbone while keeping ResNet-50 as the detection backbone achieves the highest AP and mAP, demonstrating that this combination is the most effective for boosting the performance of SPAR-Det.

Table 5. Impact of segmentation backbones on AI-TOD under the SPAR-Det framework.

Segmentation Backbone	IoU	AP_{vt}	AP	AP_{50}
UNet [4]	22.42	15.7	29.6	61.1
DeepLabV3+ [1]	38.15	15.2	27.6	55.0
Segformer [6]	13.65	14.2	29.6	58.9
UNet++ [8]	26.39	**16.0**	**31.2**	**61.2**

2.6. Analysis of Gating Network

In this study, we analyze the impact of the type and number of statistics used in the gating network on detection performance. As summarized in Table 6, the data-driven approach corresponds to the conventional MoE [3], while the statistics-based approach begins with Stats-4, which consists of Haralick-type descriptors (*entropy*, *contrast*, *region_attn_mean*, *attn_variance*). We then progressively augment this set by adding *max_attn*, *object_area_ratio*, *spatial_x*, and *spatial_y*, resulting in Stats-6 (including *spatial_x* and *spatial_y*) and Stats-8 (combining all descriptors).

Experimental results show that Stats-4 already outperforms the conventional MoE, and performance further improves as the set is expanded to Stats-6 and Stats-8. In particular, Stats-8 achieves 31.2 AP and 61.2 AP_{50}, demonstrating the best overall performance. These findings indicate that integrating traditional statistical descriptors with spatial and size-specific cues tailored for small object detection effectively stabilizes expert routing and enhances detection accuracy, while incurring only a negligible increase in the number of parameters.

Table 6. Effect of Statistical Configurations on Model Performance

Strategy	AP	AP_{50}	Params (M)
Data-driven	26.9	56.4	190.312
Stats-4	29.2	59.2	139.061
Stats-6	29.8	58.9	139.061
Stats-8	31.2	61.2	139.061

2.7. Discussion

Integrating a segmentation backbone significantly improves small object detection, but its effectiveness depends on the quality of the segmentation outputs. This underscores the importance of evaluating the reliability of segmentation features prior to deployment.

Two key factors affect this dependency: (1) the architectural compatibility between the segmentation and detection backbones, and (2) the relevance of the segmentation model's pretraining dataset to the target detection domain If the pretrained data are misaligned with the detection domain or there is structural incompatibility between backbones, this can introduce noise rather than useful guidance, which ultimately diminishes the benefits of feature-level fusion.

3. Experimental results

3.1. Quantitative evaluation on Visdrone

We evaluate SPAR-Det on the VisDrone dataset and compare it with the baseline RFLA [7]. The supplementary analysis focuses on visualizing detection differences between the two models.

As shown in Figure 2, SPAR-Det produces more true positives (TP) while reducing false negatives (FN), especially in challenging cases such as small or partially occluded objects. These observations suggest that the segmentation-guided fused features and expert routing mechanisms in SPAR-Det improve its ability to detect subtle object instances that are often missed by the baseline. The examples demonstrate improved recall without a significant increase in false positives.

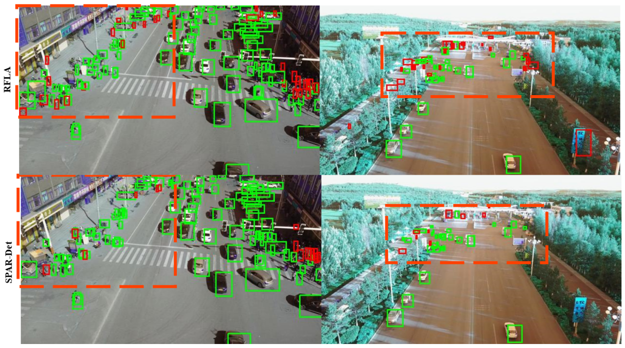

Figure 2. Qualitative comparison between SPAR-Det and the baseline RFLA [7] on the VisDrone dataset. SPAR-Det detects more true positives (TP) and reduces false negatives (FN), particularly for small or occluded objects, demonstrating improved recall through segmentation-guided fusion and expert routing.

3.2. Quantitative evaluation on AI-TOD

A class-wise analysis of SPAR-Det on the AI-TOD dataset is presented in Figure 3. This figure illustrates the detection performance across all object categories, highlighting the model's robustness in handling challenging classes such as windmill, *storage tank*, and *bridge*, which often exhibit high density and low contrast. The consistent performance across categories demonstrates the effectiveness of our segmentation-guided feature enhancement and expert routing design in dense satellite imagery scenarios.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. 4

[2] Dawei Du et al. Visdrone: A large-scale benchmark and challenge for vision meets drones. *arXiv preprint arXiv:1804.07437*, 2018. 1

[3] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023. 4

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[5] Jian Wu et al. Ai-tod: A benchmark dataset for tiny object detection in aerial images. *Remote Sensing*, 12(10):1605, 2020. 1

[6] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021. 4

[7] Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. RFLA: Gaussian receptive field based label assignment for tiny object detection. *arXiv preprint arXiv:2208.08738*, 2022. 5

[8] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *International workshop on deep learning in medical image analysis*, pages 3–11. Springer, 2018. 4

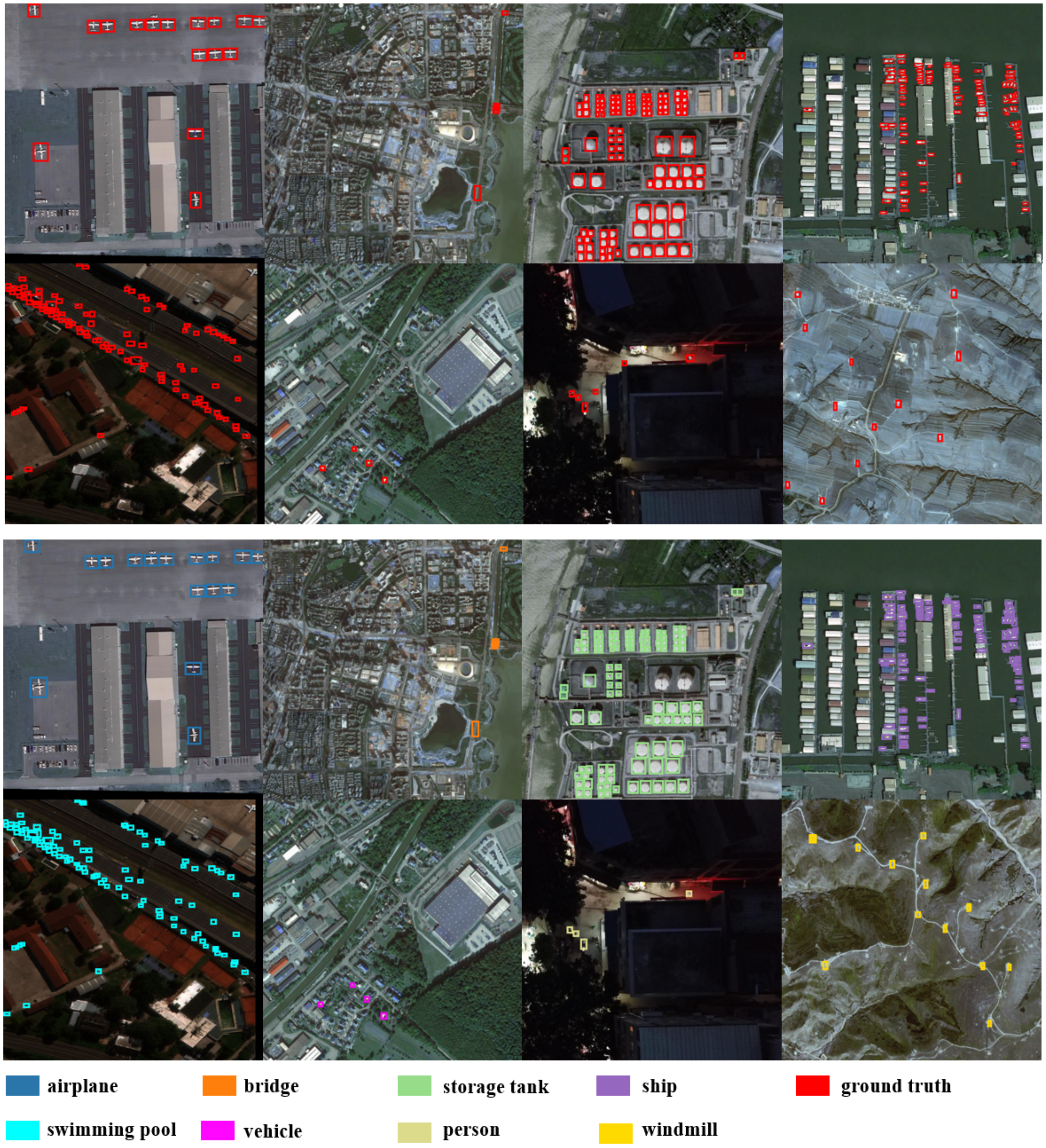

Figure 3. Qualitative examples of true positive (TP) cases on AI-TOD and VisDrone. SPAR-Det accurately localizes small objects under challenging conditions, such as low contrast, occlusion, and dense scenes, demonstrating its effectiveness in real-world scenarios.