

Supplementary Information

FDMA-Net: Four-directional Mamba alignment for tiny object detection in visible and thermal imagery

Sungju An, Siwoo Kim, Jhonghyun An

Department of Artificial Intelligence and Software, Gachon University

Correspondence: jhonghyun@gachon.ac.kr

Supplementary Note

Directional configurations

Supplementary Table S1 compares single-direction, two-direction, four-direction simple averaging, and the final FDMA configuration.

Supplementary Table S1. Directional configuration comparison. All AP values are percentages and are reported as mean $\pm$ sample standard deviation over seeds 0, 1, and 2. Bold indicates the highest mean in each column.

Configuration	AP ₂₅ ^{all}	AP ₅₀ ^{tiny}	AP ₅₀ ^{tiny1}	AP ₅₀ ^{tiny2}	AP ₅₀ ^{tiny3}	AP ₅₀ ^{small}	AP ₅₀ ^{all}	FPS
1-dir ($\rightarrow$)	60.20 $\pm$ 0.53	49.05 $\pm$ 0.71	18.92 $\pm$ 5.84	37.41 $\pm$ 0.92	53.74 $\pm$ 0.68	30.52 $\pm$ 0.74	47.49 $\pm$ 0.55	38.63 $\pm$ 0.22
2-dir horizontal ($\rightarrow, \leftarrow$)	60.36 $\pm$ 1.18	49.06 $\pm$ 1.36	19.35 $\pm$ 7.12	37.56 $\pm$ 1.31	53.69 $\pm$ 1.24	30.47 $\pm$ 1.15	47.48 $\pm$ 1.24	35.82 $\pm$ 0.24
2-dir orthogonal ($\rightarrow, \downarrow$)	61.17 $\pm$ 0.40	49.94 $\pm$ 0.60	21.43 $\pm$ 5.96	38.21 $\pm$ 0.67	54.01 $\pm$ 0.56	30.96 $\pm$ 0.59	48.26 $\pm$ 0.45	36.26 $\pm$ 0.03
4-dir + simple mean	60.55 $\pm$ 0.98	49.38 $\pm$ 1.19	20.36 $\pm$ 6.85	37.81 $\pm$ 1.08	53.83 $\pm$ 1.06	30.71 $\pm$ 0.93	47.65 $\pm$ 1.09	32.95 $\pm$ 0.10
FDMA	62.07 $\pm$ 0.44	50.58 $\pm$ 1.03	22.77 $\pm$ 7.73	38.65 $\pm$ 0.80	54.42 $\pm$ 0.95	31.37 $\pm$ 0.68	49.08 $\pm$ 0.89	32.93 $\pm$ 0.09

The final FDMA configuration achieves the highest mean AP values across all reported detection metrics. Compared with four-direction simple averaging, its mean AP25 increases from 60.55% to 62.07%, and mean overall AP50 increases from 47.65% to 49.08%. Given the limited number of seeds, these differences are interpreted descriptively based on the reported mean and sample standard deviation.

Object-center alignment

Using boxes from both modalities, we quantify the behavior illustrated in Fig. 3 on 948 training-subset images with 3,620 automatically matched object pairs. Final-model P3 sampling reduces the mean center-reference error from 12.30 to 10.43 pixels, with a reported three-seed mean relative reduction of 15.16% (Supplementary Table S2).

Supplementary Table S2. Object-center alignment diagnostic. The relative reduction is reported as a three-seed mean. The last two rows are object proportions computed within each image and then averaged over images.

Measure	Result
Mean center-reference error before P3 sampling	12.30 px
Mean center-reference error after P3 sampling	10.43 px
Relative error reduction	15.16%
Displacement component toward the matched RGB center	87.37%
Reduced center-reference error	80.04%

In the main detection ablation, adding FDMA to paired fusion raises AP50 by 2.86 points (43.44% to 46.30%) and AP75 by 1.55 points (4.10% to 5.65%). Together, these observations support partial object-level feature alignment useful for detection, rather than complete pixelwise registration. This diagnostic uses training-subset images, not an independent test set.

Supplementary Methods

Implementation and evaluation

The implementation is built in MMDetection [1] using a GFL-based dense detection formulation [2] with dual YOLOv8m backbones [3]. The proposed FDMA, TDE, and CSDH modules are integrated into this framework. The object-region feature regularizer follows Peng et al. [4], as described in the main manuscript. Selected configurations are summarized in Supplementary Table S3.

Supplementary Table S3. Selected training and evaluation settings for the main experiments. Image dimensions are width $\times$ height.

Setting	RGBTDronePerson	VTUAV-det
Labels and boxes	Person, rider, crowd; thermal-coordinate boxes	Person only (original category ID 0); thermal-coordinate boxes
Data pairs	4,841 training; 1,225 validation	11,392 original training; 11,208 in the loader; all 5,378 original test pairs
Resize / padding	Fit within 640×512 , preserve aspect ratio; pad to multiples of 32	$1920 \times 1080 \rightarrow 640 \times 360$; pad to 640×384
Training	Two RTX A5000 GPUs, total batch size 16 pairs, 12 epochs; SGD, LR 0.01, momentum 0.9, weight decay 10^{-4} .	
Warm-up / LR	500 warm-up iterations; tenfold learning-rate drops at epochs 9 and 12	Linear warm-up: 500 iterations, ratio 0.001; tenfold learning-rate drops at epochs 8 and 11
Augmentation	Paired horizontal flips	Paired horizontal flips, $p = 0.5$; no evaluation flip or multi-scale augmentation
Checkpoint	Best validation AP50 per run	Best validation AP50 per run
Reporting	Seeds 0, 1, and 2 for main Tables 1, 3, and 4; Tables 3 and 4 report mean $\pm$ sample standard deviation.	Seeds 0, 1, 2; main Table 2 reports mean AP
Post-processing	Score 0.05, class-wise NMS IoU 0.3, at most 100 boxes/image	Score 0.05, NMS IoU 0.5, nms_pre=1000 , at most 100 boxes/image
Timing	One RTX A5000, batch size 1, FP32; 50 warm-ups followed by three repetitions of 300 measurements.	

We retain the dataset-specific post-processing settings used in each experimental configuration: NMS IoU 0.3 for RGBTDronePerson and 0.5 for VTUAV-det; these thresholds are fixed across all runs within each benchmark.

For RGBTDronePerson, checkpoint selection and reporting use the same validation split. For VTUAV-det, checkpoints are selected based on validation AP50, and the selected checkpoints are evaluated on the original test split for reporting.

For VTUAV-det, person-empty training images are filtered, without merging other categories into person.

Training uses two loader workers per GPU, L2 gradient clipping at 35, no automatic LR scaling, and **deterministic=False**. The learnable TDE scale is excluded from weight decay. The two datasets use dataset-specific input normalization with **to_rgb=True**.

For RGBTDronePerson,

$$\begin{aligned}\mu_{\text{RGB}} &= (115.37, 121.82, 122.63), & \sigma_{\text{RGB}} &= (85.13, 89.01, 88.27), \\ \mu_{\text{T}} &= (93.10, 93.10, 93.10), & \sigma_{\text{T}} &= (50.24, 50.24, 50.24).\end{aligned}$$

For VTUAV-det,

$$\begin{aligned}\mu_{\text{RGB}} &= (83.20, 92.24, 97.70), & \sigma_{\text{RGB}} &= (57.77, 57.41, 57.69), \\ \mu_{\text{T}} &= (134.84, 134.84, 134.84), & \sigma_{\text{T}} &= (81.58, 81.58, 81.58).\end{aligned}$$

Inference time for both datasets includes decoding and NMS but excludes data loading.

Sampling and loss conventions

Bilinear sampling uses zero padding and `align_corners=True` in PyTorch `grid_sample` [5]. For $W_l, H_l > 1$, the reference grid in the main manuscript corresponds to

$$x_0 = \frac{(j + 0.5)(W_l - 1)}{W_l}, \quad y_0 = \frac{(i + 0.5)(H_l - 1)}{H_l}. \quad (\text{S1})$$

The visible sampling position is $(x_0 + \Delta x_l, y_0 + \Delta y_l)$. Zero learned displacement therefore gives reference-sampled features, not generally the original unsampled tensors. The base range $R = 16$ is specified before the learnable level scale $\alpha_l \in (0, 2)$.

For ordered support values b_k and a target $z \in [b_k, b_{k+1}]$, the adjacent-bin distributional loss [2] is

$$\ell_{\text{dist}}(z, p) = -\frac{b_{k+1} - z}{b_{k+1} - b_k} \log p_k - \frac{z - b_k}{b_{k+1} - b_k} \log p_{k+1}. \quad (\text{S2})$$

At an endpoint, all target mass is placed on that endpoint. CSDH decodes continuous expectations from nine center bins over $[-2, 2]$ and 33 size bins over $[0, 32]$, in stride units.

In the main objective, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{dist}}$ already include weights 2 and 0.25; these are not applied twice. Feature regularization has outer weight 0.1 and compares fused and unsampled thermal channel distributions within expanded object regions, with gradients on both branches. Directional-consistency weights are 0.01 in epochs 1–8, 0.005 in epochs 9–10, and zero in epochs 11–12.

References

- [1] Chen, K. *et al.* MMDetection: Open MMLab detection toolbox and benchmark. Preprint at <https://arxiv.org/abs/1906.07155> (2019).
- [2] Li, X. *et al.* Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *Advances in Neural Information Processing Systems* **33** (2020). <https://arxiv.org/abs/2006.04388>.
- [3] Jocher, G., Chaurasia, A. & Qiu, J. Ultralytics YOLOv8. Software (2023). <https://github.com/ultralytics/ultralytics>.
- [4] Peng, P. *et al.* COXNet: Cross-layer fusion with adaptive alignment and scale integration for RGBT tiny object detection. *IEEE Trans. Circuits Syst. Video Technol.* **36**, 596–608 (2026). <https://doi.org/10.1109/TCSVT.2025.3595147>.
- [5] PyTorch. `torch.nn.functional.grid_sample` documentation. https://docs.pytorch.org/docs/main/generated/torch.nn.functional.grid_sample.html.