Supplementary material for "Polymer Banknote Recognition and Tear Detection under Adverse Illumination: A Dataset and Local-Contrast Gated Attention"

This document accompanies the manuscript. Tables and figures are numbered S1, S2, ... to match the cross-references in the main paper; references to "Table n" and "Fig. n" without the prefix S point to the main paper. As there, AP is average precision, mAP is the mean AP over classes, IoU is intersection over union, mAP@50 is mAP at an IoU threshold of 0.5, and mAP@50–95 averages over IoU thresholds from 0.50 to 0.95. Δ is the drop in tear AP@50–95 from neutral light to glare. LCGA is the Local-Contrast Gated Attention module of IR-YOLOv8n, $D(x)$ is its local gradient-energy map and w_s its signed spatial gate. The full comparison of all models (Table 2 and Table 3), the operator, window and insertion ablations (Table 4), the gate statistics (Table 5), the efficiency and tear-robustness plots (Figs. 3 and 2) and the loss, augmentation, CLAHE and latency settings (Sec. 5) are in the main paper.

1. Full table for the generic attention modules

Table S1 adds parameters, FLOPs, overall mAP@50–95 and denomination mAP to the generic-attention rows of the ablation in the main paper (Table 4).

Table S1: Generic attention modules inserted at $P_3/8$ and $P_4/16$: squeeze-and-excitation (SE) [1], efficient channel attention (ECA) [2], convolutional block attention module (CBAM) [3], coordinate attention (CA) [4] and the parameter-free attention module SimAM [5]. Denom.: denomination mAP@50–95; Δ: neutral minus glare tear AP@50–95. Test split, 50-epoch schedule, single run.

	Params	FLOPs	mAP@50	mAP@50–95	Denom.	Tear AP@50–95		Δ
Variant	(M)	(G)	(%)	(%)	(%)	Neutral	Glare	
YOLOv8n	3.01	8.09	97.24	91.42	95.54	71.37	62.47	8.90
+ SE	3.01	8.09	97.38	91.55	95.68	71.55	62.84	8.71
+ ECA	3.01	8.09	97.35	91.51	95.65	71.48	62.79	8.69
+ CBAM	3.02	8.10	97.45	91.68	95.72	71.85	63.12	8.73
+ CA	3.02	8.10	97.68	91.95	95.95	72.25	63.85	8.40
+ SimAM	3.01	8.10	97.60	91.88	95.88	72.10	63.70	8.40
+ LCGA (full)	3.02	8.10	98.76	93.14	96.83	74.39	67.96	6.43

Parameter additivity. Parameter counts are additive. Relative to the 3.012 M-parameter YOLOv8n baseline, the LCGA module adds 2,198 parameters at $P_3/8$ (64 channels), 8,342 at $P_4/16$ (128 channels) and 32,918 at $P_5/32$ (256 channels), yielding 3.023 M for $P_3 + P_4$ and 3.056 M for $P_3 + P_4 + P_5$ (the insertion rows of Table 4). These counts equal $2C^2/r + 147 + 2 + 1 = C^2/2 + 150$ for $r = 4$, that is, two bias-free MLP layers of the channel gate, a bias-free 7×7 convolution on three input descriptors (147 weights), the two affine parameters of batch normalisation [6] and the scalar γ. For example, $C = 64$ gives 2,048 + 150 = 2,198. The latency of the $P_3 + P_4$ model is 7.24 ms and that of the $P_3 + P_4 + P_5$ model is 8.05 ms (batch size 1, FP32, Tesla T4).

2. Optimisation and initialisation details

Optimisation. Training used the Ultralytics framework (version 8.3) with `optimizer: auto`. For 50 epochs on 1,260 images at batch size 16 (approximately 3,950 total iterations), the framework automatically selects AdamW rather than SGD (which is triggered only above 10,000 iterations), setting an initial learning rate $\mathrm{lr}_0 = 0.002 \times 5/(4 + nc) \approx 0.00091$ for $nc = 7$ classes, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 5×10^{-4} (0.0 on normalisation and bias parameters), and a 3-epoch linear warmup. The learning rate follows a cosine annealing schedule decaying to $0.01 \times \mathrm{lr}_0$ (`cos_lr: true`, `lrf: 0.01`) across all models. The loss weights, augmentation, CLAHE settings and latency protocol are given in Sec. 5 of the main paper.

Pretraining and initialisation. Official COCO-pretrained [7] weights provided by Ultralytics were used for standard baselines: `yolov8n.pt`, `yolov8s.pt`, `yolo11n.pt`, `yolov5nu.pt`, `yolov10n.pt` and `yolov9t.pt`. In contrast, Ultralytics does not provide official pretrained weights for Ghost-YOLOv8n, which was defined from an architecture YAML file; transferring weights from `yolov8n.pt` copied 277 of 397 tensors (matching standard C2f, SPPF and head layers), leaving all GhostConv downsampling and transition convolutions to be trained from scratch with random initialisation. Within the 50-epoch fine-tuning budget, these randomly initialised layers probably account for much of the lower performance of Ghost-YOLOv8n (85.97% overall mAP@50–95 and 47.07% overall tear AP@50–95), although we did not test this.

Initialisation of IR-YOLOv8n. IR-YOLOv8n transferred 319 of 373 tensors from `yolov8n.pt`. The newly introduced LCGA modules were initialised with $\gamma_0 = 0.05$ as described in the main paper, so that each module is a near-identity mapping (a perturbation of at most 5%) at epoch 0. The 54 non-transferred tensors comprise the 18 new LCGA parameters (2×9 across layers 15 and 18) and the 36 classification head parameters in `Detect.cv3` (which mismatch COCO's 80-class hidden dimension, identically to baseline YOLOv8n, which also transfers 319 of 355 tensors); all C2f blocks at layers 15 and 18 are 100% transferred from `yolov8n.pt`.

3. Feature-level statistics of the gate

Activations at layers 15 and 18 of the trained IR-YOLOv8n were recorded on all 276 test images. The feature space was partitioned into three regions: (i) *highlight cells*, which have value $V \geq 225$ in the HSV colour space of the input, downsampled to the feature-map grid by area interpolation and thresholded at coverage > 0.4 (194,303 cells at layer 15, 48,623 at layer 18); (ii) *tear-box cells*, which lie inside ground-truth tear bounding boxes, downsampled and thresholded at coverage > 0.3 (7,610 and 2,032 cells); and (iii) *remaining cells*, which correspond to the banknote foreground substrate ($V \geq 25$, excluding background) with both highlight and tear regions subtracted, thresholded at coverage > 0.5 (1,526,470 and 382,661 cells). Overlaps between highlights and tear boxes occur when specular glare directly strikes a tear; these cells are kept in both respective masks, whereas the remaining-substrate region is disjoint from both. Highlight cells are defined by brightness alone and are therefore a proxy for specular highlights.

Table 5 of the main paper lists the mean and standard deviation of $D(x)$ and w_s in each region. The standard deviations are several times larger than the differences between the highlight and remaining regions, so the two distributions overlap almost completely and the highlight region is only marginally flatter than the rest. Cells are spatially correlated and pooled over 276 images, so no significance tests are attached to these statistics. The summed weights of the 7×7 spatial kernel for the channel-maximum descriptor are -0.333 (layer 15) and -0.979 (layer 18), and for $D(x)$ they are $+1.492$ and $+1.932$. These sums should be read with care: the three descriptors have different scales ($D(x)$ is about 0.1 on average) and the convolution is followed by batch normalisation, so the raw weights are not directly comparable and do not by themselves explain the sign of w_s in a region. Fig. S1 shows one example of the feature activation and Fig. S2 summarises these statistics.

4. Per-regime profile

Fig. S3 plots the per-regime profile of the compared models summarised in Table 2 of the main paper.

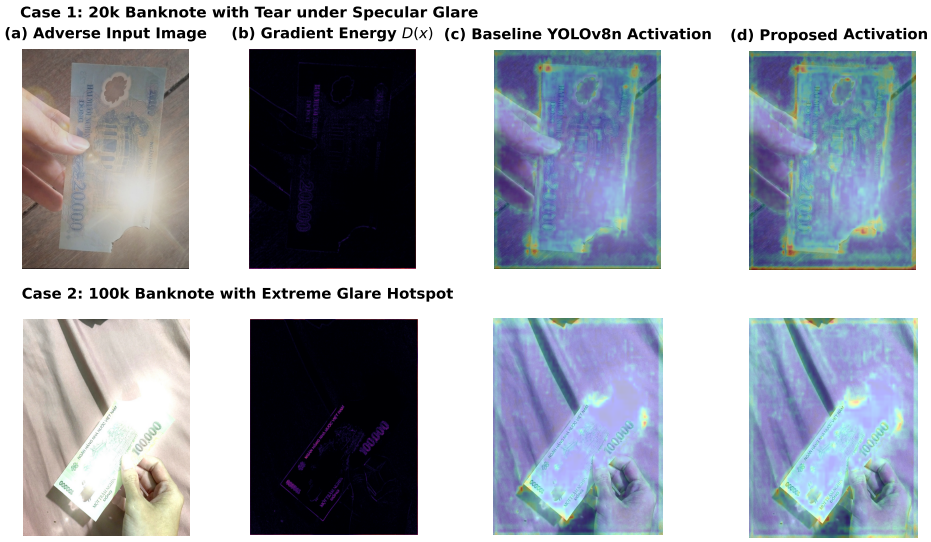


Figure S1: Feature activation at layer 15 under glare: (a) input, (b) $D(x)$, (c) YOLOv8n activation, (d) IR-YOLOv8n activation. Single examples, not statistical evidence.

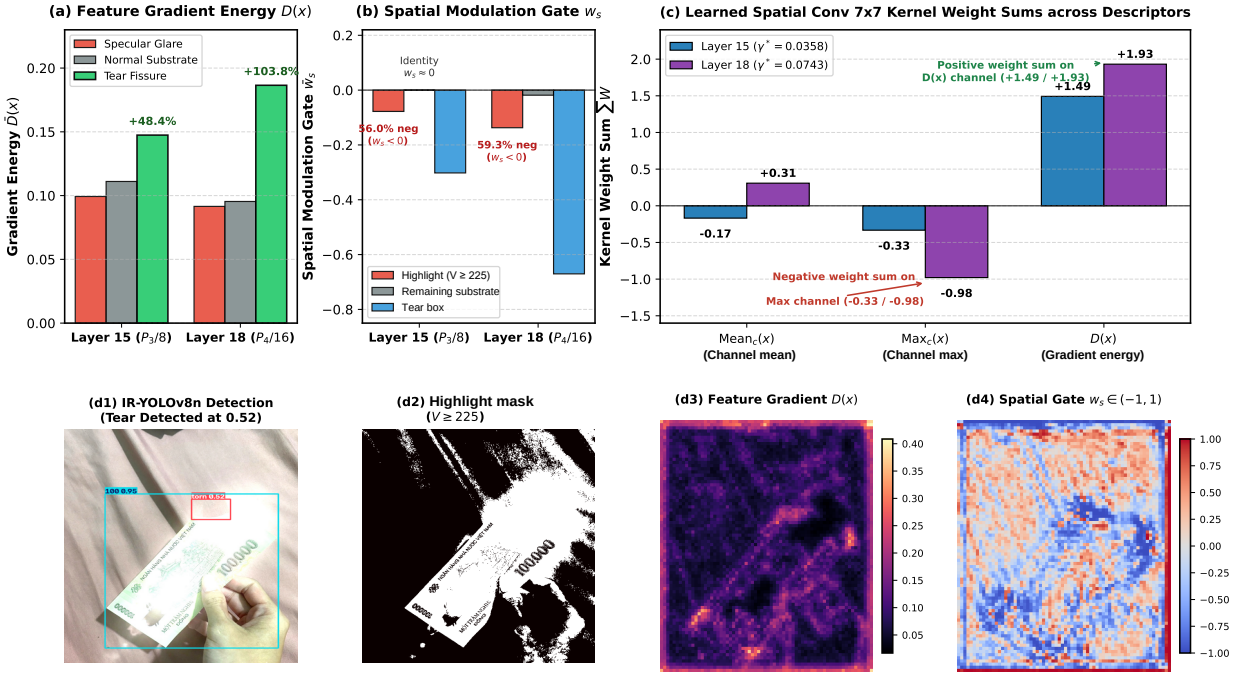


Figure S2: Quantitative deep-feature analysis across the test split: (a) mean gradient energy $\bar{D}(x)$ by region (specular glare: highlight cells; normal substrate: remaining cells; tear fissure: tear-box cells), (b) mean spatial gate $\bar{w}_s$, (c) summed weights of the learned 7×7 convolution kernel for each descriptor, and (d) exemplar test sample (100k note with tear under specular glare). Although the kernel weights for $D(x)$ are positive in (c), the net post-BN gate w_s in (b) is negative over both highlights and tear boxes (Table 5 of the main paper), which is consistent with attenuation or normalisation rather than amplification of tears.

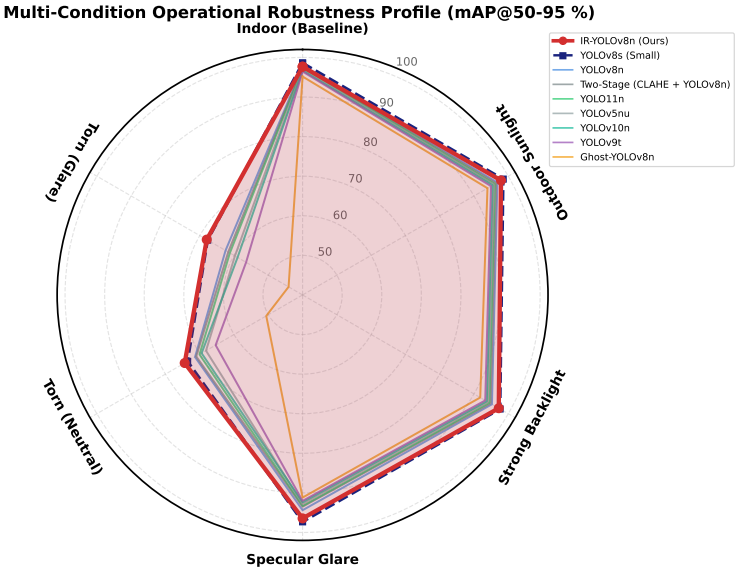


Figure S3: Per-regime profile of the compared models. “Two-Stage” denotes the CLAHE + YOLOv8n pipeline.

References

[1] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141.

[2] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11534–11542.

[3] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, CBAM: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.

[4] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13713–13722.

[5] L. Yang, R.-Y. Zhang, L. Li, X. Xie, SimAM: A simple, parameter-free attention module for convolutional neural networks, in: Proceedings of the International Conference on Machine Learning (ICML), 2021, pp. 11863–11874.

[6] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proc. Int. Conf. Machine Learning (ICML), Vol. 37 of Proceedings of Machine Learning Research, 2015, pp. 448–456.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Proc. European Conf. Computer Vision (ECCV), Vol. 8693 of Lecture Notes in Computer Science, 2014, pp. 740–755.