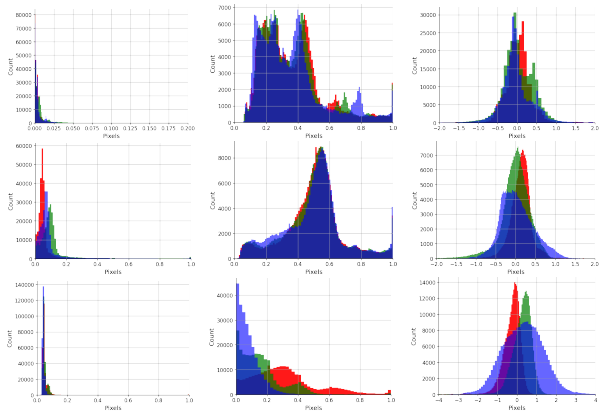
Beyond RGB: Adaptive Parallel Processing for RAW Object Detection

Supplementary Material



(a) RAW (b) sRGB (c) RAM (ours)

Figure 1. Example histograms of R, G, and B channels for RAW, sRGB, and RAM outputs. The top row shows results for ROD-Day dataset (24-bit), the middle for LOD-Normal (14-bit), and the bottom row shows results for NOD-Nikon dataset (14-bit). The RAW and sRGB data are divided by the maximum pixel value for visualization. The RAM pre-processing method produces a distribution resembling a normal distribution, centered around zero.

A. Distribution Analysis

In Fig. 1, we present the histograms of three data types: RAW sensor data, sRGB, and the output of our pre-processing method on multiple datasets. Observing these distributions, we note that RAW data tends to cluster near zero, making it less effective for DNN-based learning and limiting the ability of the network to extract meaningful features [8, 9]. sRGB data provides a distribution that is better suited for DNN input, as the ISP processing stretches the histogram and redistributes pixel intensities across the available range by enhancing the darker pixels and increasing global contrast. However, this processing often leads to information loss and reduced dynamic range. An example of this information loss can be seen in the histograms (column b of Fig. 1), where pixel values cluster near saturation, leading to clipping. On the other hand, we observed that the pixel distribution produced by RAM resembles a normal distribution, centered around zero—a behavior that has been discussed and shown experimentally [1, 3, 5, 7] to better support the ability of the network to learn and converge efficiently during training. This behavior arises as a byproduct of the complex, learned transformation performed by RAM, which go beyond simple normalization methods, such as scaling and shifting.

Table 1. Experimental results comparing RAM applied to RAW images vs RAM applied to sRGB images.

Method	LOD-Dark		LOD-Normal	
	mAP	mAP_{50}	mAP	mAP_{50}
RAW	28.5	50.8	32.7	53.8
sRGB	28.7	51.2	34.5	57.0
RAM (sRGB)	32.9	56.0	37.9	60.4
RAM (RAW)	**34.9**	**57.6**	**40.1**	**61.4**

B. Experimental Setup

We use the following settings for training and evaluation in our experiments presented in Section 4: Faster R-CNN is trained with SGD [6] (lr=0.02, momentum=0.9, weight decay=$1e-4$), using a multi-step learning rate scheduler with decay at epochs 29, 49, and 79, for a total of 100 epochs. DINO is trained with AdamW [4] (lr=$2e{-}4$, weight decay=$1e-4$) and a multi-step learning rate scheduler with decay at epochs 27 and 33, for 50 epochs. Data augmentation included random resizing and horizontal flipping. For RAM, inputs are downsampled to 256×256 before the RPE module, and RPD params outputs are applied to the original input images. We use mean-std normalization, calculated on the dataset, for both RAW and sRGB experiments. Performance is evaluated using mean average precision (mAP) across thresholds 0.5:0.95 and 0.5.

C. Additional Evaluations

C.1. RAM: RAW vs RGB

RAM is designed as a RAW-specific pre-processing module that optimizes ISP functions for object detection by leveraging the rich, unprocessed information available in RAW images. However, one might question whether RAM's effectiveness is inherently tied to RAW data or if similar improvements could be achieved by applying it to standard RGB images. We believe the strength of our approach lies in its ability to learn and adapt ISP operations directly from unprocessed sensor data. While sRGB images have already undergone fixed ISP processing, RAW data preserves the complete sensor information, allowing RAM to discover optimal processing parameters specifically for detection tasks.

To validate this intuition empirically, we compare the performance of RAM when applied to RAW images versus sRGB images. The results on Tab. 1 show that while applying RAM to sRGB can improve the image for detection, it does not perform as well as RAM on RAW. This emphasizes

Table 2. Comparison of sRGB and RAM performance across different YOLOX model sizes (Small, Medium, and Large) on the ROD-Night dataset.

Method	Data Type	mAP	mAP_{50}
YOLOX-S	RAW	46.7	72.3
	sRGB	50.3	76.4
	RAM	**54.8**	**80.7**
YOLOX-M	RAW	50.9	76.4
	sRGB	54.8	79.7
	RAM	**58.0**	**83.0**
YOLOX-L	RAW	53.3	78.0
	sRGB	56.9	81.9
	RAM	**59.8**	**84.0**

Table 3. Comparison of training each dataset separately vs training one model on all datasets. Results are reported using mAP.

Training Approach	PASCALRAW	NOD-Nikon	NOD-Sony	LOD-Dark
Single dataset	67.7	31.9	32.8	48.0
All datasets	**69.6**	**32.8**	**35.8**	**48.2**

that RAM is specifically tailored to utilize the full potential of RAW data, making it an effective pre-processing solution for RAW-based object detection.

C.2. Scalability Across Different Detector Sizes

To test the scalability of our approach, we evaluate its performance on different YOLOX detector sizes (Small, Medium, and Large) trained from scratch on the ROD-Night dataset. Each model size corresponds to a different number of parameters, allowing us to examine how RAM adapts across varying computational capacities.

The results in Tab. 2 show that RAM consistently outperforms sRGB across all model sizes, demonstrating its adaptability and effectiveness regardless of the detector's scale. Notably, the performance gains achieved by integrating RAM surpass those obtained by simply increasing the detector size when using sRGB. For instance, YOLOX-S with RAM achieves higher mAP_{50} than YOLOX-M with sRGB, and similarly, YOLOX-M with RAM outperforms YOLOX-L with sRGB. This highlights the significant impact of optimizing RAW pre-processing, confirming that our efficient approach improves detection performance more effectively than just increasing the model size.

C.3. Generalization Across Datasets

To demonstrate RAM's ability to generalize across diverse data, we compare training a separate model on each dataset with training a single model jointly on all four datasets, which vary significantly in dynamic range and lighting conditions. This setting is especially challenging for RAW data: unlike RGB datasets that typically share a standardized 8-bit dynamic range, RAW datasets differ widely in their dynamic ranges and sensor-specific properties. Nevertheless, as shown in Tab. 3, RAM adapts effectively by generating optimal parameters for each input image, enabling it to handle such variability. It not only performs well across individual datasets, but also benefits from the additional data, despite distribution differences, achieving improved performances overall.

D. RAM vs RAM-T

D.1. Model Configuration

Tab. 5 details the layer configurations for both RAM and RAM-T (Tiny) architectures, with primary differences lying in the number of channels and kernel sizes, which ultimately affect the overall FLOPs and parameter count. The RPEncoder performs the main processing by transforming the input image into a compact feature vector, which is then fed into each RPDecoder. The RPDecoder is lightweight, making it efficient to add additional ISP functions to the pipeline without a significant computational cost. The Feature Fusion module employs a reverse-hourglass design, where the input and output channels are the smallest, while the middle layers are the largest. This structure enables efficient fusion of all processed inputs, capturing the most essential features needed for high-quality object detection.

D.2. Quantitative Evaluation

The comparison between RAM and RAM-T across multiple RAW object detection datasets shown in Tab. 4 demonstrates the effectiveness of RAM-T in achieving near-identical performance to the full RAM module. While RAM shows superior performance on most datasets, RAM-T remains a highly competitive alternative, providing a nearly equivalent detection quality with fewer parameters and reduced memory usage as shown in section 4.4. This makes RAM-T especially suitable for scenarios where computational resources and memory are more constrained, without a significant sacrifice in accuracy.

E. Additional Visualizations

In this section, we present additional visualizations comparing the detection results of models trained on RAW, sRGB and our method, RAM. The images in these visualizations represent different conditions synthesized on the ROD-Night dataset.

The visualizations, shown in Fig. 2, illustrate the effects of synthesized weather conditions—rain, snow, and fog—on object detection performance, corresponding to the experiments discussed in section 4.5.2. The images in the first row demonstrate the impact of rain on detection, where the RAW and sRGB models struggle to detect small cars in rainy conditions, while RAM successfully identifies these

Table 4. Comparison between RAM and RAM-T across different RAW object detection datasets. Results are reported using mean Average Precision (mAP) and mAP at 50% IoU (mAP50).

Method	ROD-Day		ROD-Night		NOD-Nikon		NOD-Sony		LOD-Dark		LOD-Normal		PASCALRAW	
	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}
RAM	**28.3**	**45.1**	**44.5**	**69.0**	**31.0**	**56.3**	**32.4**	**59.1**	34.9	57.6	**40.1**	**61.6**	**66.4**	**92.3**
RAM-T	27.9	44.4	44.2	68.5	30.7	55.8	32.2	58.1	**35.8**	**58.4**	**40.1**	61.4	66.3	92.2

Table 5. Layer configurations for RAM and RAM-T. ConvBlock includes: Conv2d, BatchNorm2d and LeakyReLU layers.

Module	Layer	Type	RAM	RAM-T
RPEncoder	1	ConvBlock	3→16 channels, 7x7 kernel	3→16 channels, 3x3 kernel
	2	MaxPool	2x2 kernel	2x2 kernel
	3	ConvBlock	16→32 channels, 5x5 kernel	16→32 channels, 3x3 kernel
	4	MaxPool	2x2 kernel	2x2 kernel
	5	ConvBlock	32→128 channels, 3x3 kernel	32→64 channels, 3x3 kernel
	6	MaxPool	2x2 kernel	2x2 kernel
	7	AdaptiveAvgPool2d	1x1 kernel	1x1 kernel
RPDecoder	1	Linear	128 units	64 units
	2	LeakyReLU	-	-
	3	Linear	128 units	64 units
Feature Fusion	1	ConvBlock	12→16 channels, 3x3 kernel	12→16 channels, 3x3 kernel
	2	ConvBlock	16→64 channels, 3x3 kernel	16→32 channels, 3x3 kernel
	3	ConvBlock	64→16 channels, 3x3 kernel	32→16 channels, 3x3 kernel
	4	Conv2D	16→3 channels, 1x1 kernel	16→3 channels, 1x1 kernel

objects. In the second row, heavy snow in the night images hides the cyclists on the left, making them nearly undetectable even to the human eye. Remarkably, RAM's output shows an ability to "remove" most of the snow from the image, despite not being explicitly trained to do so, allowing it to accurately capture the cyclists features. In the third row, fog covers the entire scene unevenly, making detection challenging for the model. Although RAW and sRGB manage to detect some objects, their performance falls short of RAM, which generates a more consistent image less impacted by the fog.

Fig. 3 presents detection results on noisy synthetic images and their denoised versions, as discussed in section 4.5.1. The presence of noise makes it challenging to detect occluded pedestrians and cyclists, including for RAM, as its output remains noisy without any specialized denoising component. The denoised images, produced by the state-of-the-art LED model [2], do not perfectly restore the original images and may introduce artifacts, which in some cases lead to false positives. In comparison, the best results appear in the bottom right image, where RAM, trained on denoised data, remains unaffected by any potential artifacts from the denoising process.

In conclusion, these visualizations emphasize the robustness of RAM in extreme conditions. Furthermore, they illustrate how RAM interprets images, often disregarding unnecessary or distracting features to focus on critical object features. Notably, we demonstrate these results on driving scenes, where accurate object identification in challenging conditions is essential and can be life-saving.

F. Limitations

While our proposed method demonstrates strong performance across a variety of datasets, it relies on a specific set of commonly used ISP functions tailored to these datasets. We are aware that these functions may not be suitable for all datasets, and other ISP functions might be necessary for different cases. However, the flexible design of RAM allows for easy addition or removal of ISP functions as needed.

Another challenge of working with RAW data is its sensor-specific nature. Unlike sRGB, where the ISP normalizes images to a standard 8-bit range, RAW images vary significantly in dynamic range, distribution, and characteristics across different sensors. This variability makes it more difficult to generalize a model trained on one sensor's data to another compared to sRGB, where cross-dataset generalization is more straightforward. However, as shown in Sec. C.3, this limitation can be addressed by training a robust model on diverse sensor data, enabling better adaptability across different hardware.

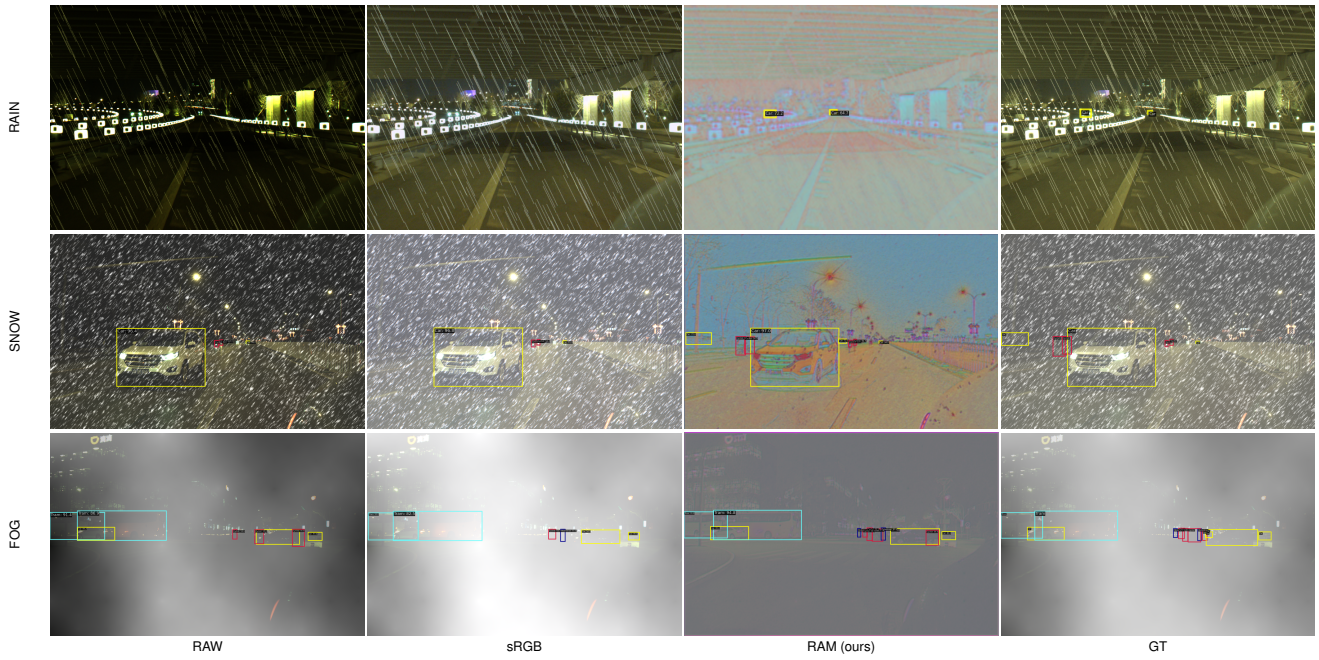


Figure 2. Object detection results in challenging weather conditions—rain, snow, and fog—synthesized on the ROD-Night dataset. The columns, in left-to-right order, show RAW, sRGB, RAM (our method), and the ground truth (GT).

Figure 3. Qualitative comparison of images across RAW, sRGB, and RAM representations. The top row shows the ground truth (GT) image. The second row presents results on the noisy images, while the third row shows results on the denoised images.

References

[1] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1

[2] Xin Jin, Jia-Wen Xiao, Ling-Hao Han, Chunle Guo, Ruixun Zhang, Xialei Liu, and Chongyi Li. Lighting every darkness in two pairs: A calibration-free pipeline for raw denoising. In *ICCV*, pages 13275–13284, 2023. 3

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016. 1

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 1

[5] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 1

[6] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 1

[7] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016. 1

[8] Ruikang Xu, Chang Chen, Jingyang Peng, Cheng Li, Yibin Huang, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Toward raw object detection: A new benchmark and a new model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13384–13393, 2023. 1

[9] Masakazu Yoshimura, Junji Otsuka, and Takeshi Ohashi. Pq-dynamicisp: Dynamically controlled image signal processor for any image sensors pursuing perceptual quality. *arXiv preprint arXiv:2403.10091*, 2024. 1