# ReRAW: RAW-from-RGB Image Reconstruction via Stratified Sampling for Efficient Object Detection on the Edge

Radu Berdan[1!] Beril Besbinar[1*] Christoph Reinders[2*] Junji Otsuka[3] Daisuke Iso[1]
[1]Sony AI [2]Leibniz University Hannover [3]Sony Group Corporation
[!]radu.berdan@sony.com [*]Equal Contribution.

## Abstract

*Edge-based computer vision models running on compact, resource-limited devices benefit greatly from using unprocessed, detail-rich RAW sensor data instead of processed RGB images. Training these models, however, necessitates large labeled RAW datasets, which are costly and often impractical to obtain. Thus, converting existing labeled RGB datasets into sensor-specific RAW images becomes crucial for effective model training. In this paper, we introduce ReRAW, an RGB-to-RAW conversion model that achieves state-of-the-art reconstruction performance across five diverse RAW datasets. This is accomplished through ReRAW's novel multi-head architecture predicting RAW image candidates in gamma space. The performance is further boosted by a stratified sampling-based training data selection heuristic, which helps the model better reconstruct brighter RAW pixels. We finally demonstrate that pretraining compact models on a combination of high-quality synthetic RAW datasets (such as generated by ReRAW) and ground-truth RAW images for downstream tasks like object detection, outperforms both standard RGB pipelines, and RAW fine-tuning of RGB-pretrained models for the same task. The code is available at:* https://anonymous.4open.science/r/ReRAW-0C87/

## 1. Introduction

The lifecycle of a digital image begins at the camera sensor, where incoming light from a scene is converted into electrical signals to form a RAW image – a single-channel Bayer-pattern array [36] where each pixel value corresponds linearly to the scene's luminosity. These RAW images are then processed locally through a camera-specific Image Signal Processor (ISP), which applies multiple functions such as demosaicking, white balancing, tone mapping etc., to yield a compressed RGB image optimized for human perception, as shown in the conventional pipeline of Fig. 1.

RGB images are preferred over RAW images due to their much smaller compressed size and compatibility with human vision, making them fast to transmit from device to device and convenient to store. However, unprocessed RAW images retain more detail, have a larger dynamic range (12-16 bits), and contain more information than their ISP-processed RGB outputs (see Supplementary **??**). This suggests that all else being equal, computer vision models such as object detectors trained directly on RAW data could outperform those trained on RGB. Hence, when operating in resource-constrained environments such as on the edge, the reduction in model accuracy due to limited compute can be mitigated by utilizing a richer signal source such as RAW images, compared to RGB images. Additionally, bypassing the ISP improves power consumption and speed, as in the optimised pipeline in Fig. 1.

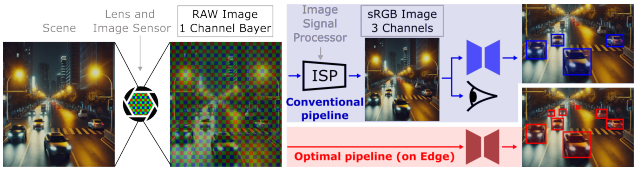



Figure 1. The conventional imaging pipeline involves an image sensor capturing a scene into a RAW image, converting that image into standard RGB fit for human consumption, and running computer vision tasks on these RGB images. The optimal pipeline would involve performing high-level tasks directly on the RAW images, on chip, and physically close to the image sensor.

Nevertheless, RGB pre-trained object detectors perform suboptimally on RAW images due to the domain gap. Several approaches have attempted to bridge this gap with learnable adapters [46, 62] or other non-linear scaling functions beyond standard ISP processing [21, 39] as well as adopting traditional feature extraction methods, such as Histogram of Oriented Gradients (HOG) [68]. Although end-to-end training of object detectors with these adaptations yields promising results, performance is still limited by the additional computational burden and the scarcity of labeled RAW datasets. Alternatively, a reverse ISP function that converts RGB images back to RAW sensor output could leverage widely available labeled RGB datasets.

Recent methods have explored learning reverse ISP functions [2, 26, 61] or adopting generative models such as CycleGAN [31] or diffusion models [51]. However, capturing the true color profile or details in bright regions in RAW images, particularly for dark scenes, remains a challenge.

Hence, in this work, we introduce ReRAW, a multi-head reverse ISP model designed to reconstruct sensor-specific RAW images from RGB inputs, without RAW meta-data, faithfully capturing the color characteristics of a target camera sensor. ReRAW enables the conversion of large RGB datasets into realistic camera-specific RAW to train object detectors that are mainly targeted for edge deployment. Our design improves reconstruction accuracy for both day and night images, effectively reproducing over-exposed regions in dark RAW images, such as those captured at night. Our primary contributions are as follows:

- We propose a novel reverse ISP model, ReRAW, capable of reconstructing sensor-specific RAW images from RGB with high fidelity. The model employs a multi-head ensemble architecture that generates multiple gamma-corrected RAW images, subsequently scaled and combined to match the sensor characteristics.
- A stratified sampling technique for training data, which results in the trained ReRAW to better capture bright regions in the converted RAW images, compared to using full training datasets or random sampling. Adding a logarithm-based loss function further boosts performance achieving state-of-the-art conversion accuracy compared to competing methods, on five diverse RAW datasets.
- Empirical evidence that pretraining object detectors on high-quality synthetic RAW datasets produced by ReRAW, followed by fine-tuning on real RAW data for specific tasks, outperforms models fine-tuned from an RGB-pretrained baseline. This approach removes the need for ISPs in traditional imaging pipelines on the edge and eliminates the need for extra fixed or learnable adapters to align RAW data with RGB-pretrained models. Our direct RAW training from scratch is effective, provided the synthetic RAW images are of high quality.

## 2. Related Work

### 2.1. RAW Images

RAW images, with their higher dynamic range and linear noise profile, offer advantages over standard RGB images, especially in low-light conditions. However, they are usually $5-10\times$ larger in size than compressed RGB, and large RAW datasets ($>100K$ images), to the best of our knowledge, do not exist. Nonetheless, with limited data, recent studies have demonstrated improved outcomes in image classification [42, 43], object detection [11, 13, 40, 59, 63, 64], semantic segmentation [11], and instance segmentation [7] with models designed for the RAW domain.

### 2.2. RGB-to-RAW Reconstruction

The advantages of RAW images, coupled with the extremely limited availability of RAW datasets, have fueled interest in reconstructing RAW images from RGB counterparts to expand labeled datasets. Traditional methods determine the relationship between a camera's output intensity and the incident light by capturing multiple images at various controlled exposure levels, with varying levels of complexity [5, 6, 12, 20, 25, 34, 45]. However, these approaches require calibration for each camera, necessitating multiple parameterized models for different settings. In contrast, modern data-driven algorithms [1, 3, 8, 9, 17, 18, 33, 37, 56, 66] leverage advanced machine learning to address this complex inverse problem without calibration. A common strategy involves simulating single or groups of ISP functions with neural networks [33, 37, 64], which requires sensor-specific configuration and training while limiting the flexibility of a data-driven perspective. Alternatively, other methods encapsulate ISP functions within a single network [3, 18, 66] yet they demand extensive RGB-RAW paired datasets.

### 2.3. Object Detection on the Edge

Modern deep learning techniques have significantly advanced object detection performance [19, 35, 38, 50, 50, 52, 58]. Beyond algorithmic advancements, this improvement could also be attributed to the increasing scale of detection models and the availability of large labeled datasets. However, many practical object detection applications operate at the edge, where limited computational power, memory, and a restricted power budget are common constraints. These requirements can be addressed either by compressing large models through knowledge distillation [30], quantization [28], and/or pruning [15, 32] or by designing lightweight models from scratch [23, 24, 29, 54, 57, 67].

We hypothesize that when edge computing is paired with RAW sensor data, the resulting edge-based imaging and sensing systems can achieve greater versatility in monitoring challenging scenes and improved overall performance.

## 3. ReRAW

Motivated by the lack of large-scale labeled RAW datasets, and analysing the shortcomings of previous reverse ISPs, we design ReRAW as a universal RGB-to-RAW converter that can handle both daytime and nighttime images, with mild or strongly skewed RAW pixel distributions.

### 3.1. Overview

ReRAW is designed to reconstruct a $W/2\times H/2\times 4$ packed RGGB (RAW) image patch ($\hat{I}_{\text{RAW}}$), given both an input $W\times H\times 3$ RGB image patch ($\hat{I}_{\text{RGB}}$) and the full RGB image ($\hat{I}_{\text{F,RGB}}$) from where the RGB patch originates from.

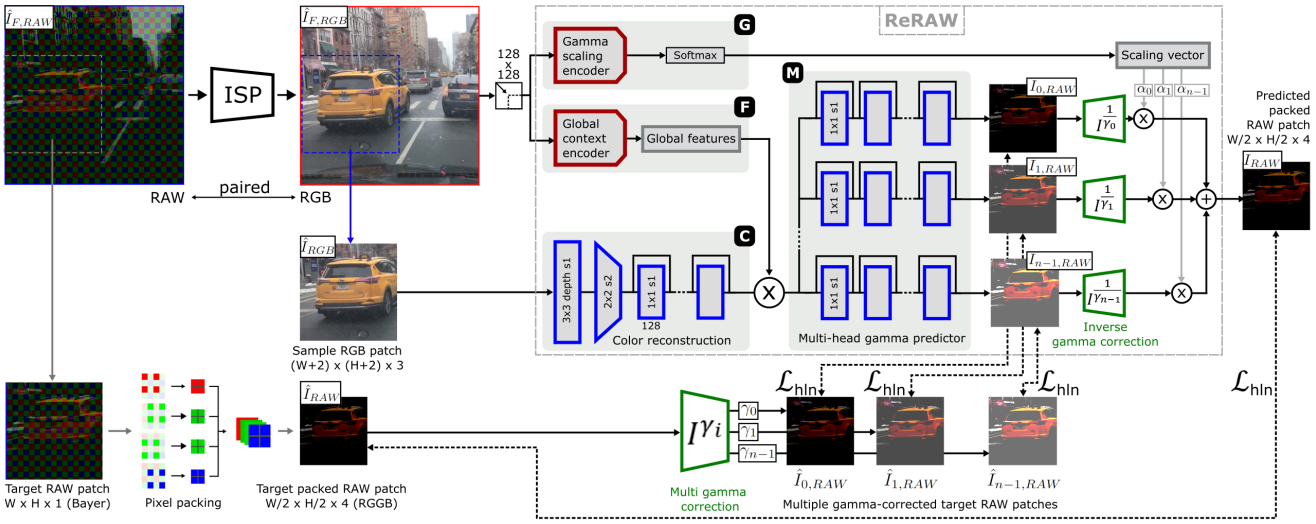



Figure 2. Illustration of ReRAW architecture and training data flow. A Global Context Encoder (F) extracts features from the full RGB image to guide the Color Reconstruction network (C), while a Multi-head Gamma Predictor (M) generates multiple gamma-corrected RAW patches. These patches are then degammaed (inverse gamma correction), scaled by a scaling vector, predicted by a Gamma Scaling Encoder (G) from the original RGB image, and summed to form the final RAW patch. Losses are applied between each intermediate gamma-corrected RAW patch and target, as well as between the final RAW output and target RAW.

We convolve ReRAW over an input RGB image to reconstruct the full required RAW image ($\hat{I}_{\text{F,RAW}}$).

The model's unique feature is its prediction in gamma space, over multiple gamma candidates, via a multi-head architecture. Gamma-corrected patch candidates are re-linearised (by applying an inverse gamma process) and proportionally averaged by a weight vector predicted by a Gamma Scaling Encoder from the original full RGB image. In this way, the model learns to select input image-dependent gamma transformations that would facilitate a better RAW conversion. Additionally, training via a stratified sampling data selection technique helps in mitigating the extreme skew of pixel values commonly found in RAW images.

### 3.2. Architecture

The full network architecture is shown in Fig. 2. The model consists of a Color Reconstruction Network (C), a Global Context Encoder (F), a Multi-head Gamma Predictor (M), and a Gamma Scaling Encoder (G). The model is trained to predict $n$ gamma-corrected RAW target patches ($I_{\text{i,RAW}}$) from an input RGB patch and its container RGB image:

$$\{I_{i,RAW}\}_{i=0}^{n-1} = M(C(\hat{I}_{\text{RGB}}) \times F(\hat{I}_{\text{F,RGB}})) \qquad (1)$$

The gamma-corrected patch candidates $I_{i,RAW}$ are then degammaed (inverse gamma process to re-linearize the images) and each multiplied by a scaling factor predicted by the Gamma Scaling Encoder from the original full RGB image. The linearised and scaled RAW candidate patch images are then summed to output the final RAW patch $I_{\text{RAW}}$.

**Color Reconstruction Network (C)** This module consists of an initial depth-wise convolutional layer with kernel size $3 \times 3$, stride 1, 3 groups, and 96 channels. It is then followed by a $2 \times 2$ stride 2 convolutional layer with 128 channels. This reduces the input spatial dimension of the RGB patch from $(W+2) \times (H+2)$ (not using padding) to $(W/2) \times (H/2)$. The output is then fed into a residual network consisting of 8 point-wise convolutional layers with depth 128. The output of the $C$ network is thus a latent tensor of $(W/2) \times (H/2) \times 128$, where each spatial multi-channel pixel has a receptive field size of only $4 \times 4$ in the original RGB patch.

**Global Context Encoder (F)** This module encodes general characteristics from the original RGB image (scaled to $128 \times 128$) such as luminosity and color space features, and uses this information to modulate the RGB-to-RAW color conversion. The module consists of a ResNet18 [22] where the last classification layer has been replaced with a linear layer of $1 \times 1 \times 128$ output size. This output tensor is then expanded to shape $(W/2) \times (H/2) \times 128$ and multiplied with the output tensor of the Color Reconstruction Network (C). We found empirically that multiplication gave better results than addition or concatenation.

**Multi-head Gamma Predictor (M)** This network holds $n = 10$ parallel heads, each consisting of 8 residual point-

wise convolutional layers with depth 128, and output depth 4. Each head outputs a candidate gamma-corrected RAW patch $I_{\text{i,RAW}}$. The motivation for using a multi-head approach is that converting to RAW in gamma space can be helpful when there are significant differences between the input RGB and output RAW pixel distributions [44]. For instance, daytime datasets with minimal ISP adjustments typically present an easier reconstruction task, whereas nighttime datasets, which necessitate more extensive ISP operations, pose a more challenging reconstruction problem. To address these varying complexities, the multi-head strategy is designed to learn distinct transformation pathways.

**Gamma Scaling Encoder (G)** The Gamma Scaling Encoder, also a ResNet18, learns to encode the full RGB image into a scaling vector of softmax-normalized values of size $n$:

$$\{\alpha_i\}_{i=0}^{n-1} = G(\hat{I}_{\text{F,RGB}}), \sum_{i=0}^{n-1} \alpha_i = 1 \tag{2}$$

The output gamma-corrected RAW patches ($I_{\text{i,RAW}}$) from the Multi-head Gamma Predictor (G) are then re-linearized (de-gammaed), scaled by each $\alpha_{\text{i}}$ value from the dynamic scaling vector and summed in order to output the final RAW predicted patch $I_{\text{RAW}}$:

$$I_{RAW} = \sum_{i=0}^{n-1} I_{i,RAW}^{\frac{1}{\gamma_i}} \times \alpha_i \tag{3}$$

### 3.3. Training Objective

The network is optimized to predict a RAW image patch $I_{\text{RAW}}$, given an input RGB patch $\hat{I}_{\text{RGB}}$ and the corresponding full RGB image $\hat{I}_{\text{F,RGB}}$. The network outputs $n$ intermediate gamma-corrected RAW patches $I_{\text{i,RAW}}$, and a final RAW patch $I_{\text{RAW}}$. In low-level image processing, training objectives normally comprise of minimising the $L_1$ or $L_2$ distance between a target and a predicted image [69]. Further, in order to address the data distribution skew towards lower pixel values, a logarithm-based loss function is sometimes used [14]. We design our own loss function *hard-log* ($\mathcal{L}_{\text{hln}}$) which heavily penalizes wrongly reconstructed pixel values, whilst converging to a $L_1$ loss for lower error pixels (Fig. 3c). This loss function helps better reconstructing sparse bright pixels in mostly dark images.

$$\mathcal{L}_{hln}(\hat{I}, I) = \frac{-1}{CHW} \|(ln(1 - |\hat{I} - I| + \epsilon)\|_1 \tag{4}$$

For the Multi-Head Gamma Predictor we chose 10 gamma values $\gamma_{\text{i}} \in \{0.1, 0.2, ...1\}$. The target gamma-corrected candidate RAW patches are therefore:

$$\hat{I}_{i,RAW} = \hat{I}_{RAW}^{\gamma_i} \tag{5}$$

The overall training objecting is thus minimizing the loss both between the candidates ($I_{\text{i,RAW}}$) and target gamma-corrected patches ($\hat{I}_{\text{i,RAW}}$), and between the final RAW patch ($I_{\text{RAW}}$) and target RAW patch ($\hat{I}_{\text{RAW}}$):

$$\mathcal{L} = \mathcal{L}_{hln}(\hat{I}_{RAW}, I_{RAW}) + \sum_{i=0}^{n-1} \mathcal{L}_{hln}(\hat{I}_{i,RAW}, I_{i,RAW}) \tag{6}$$

### 3.4. Stratified Sampling

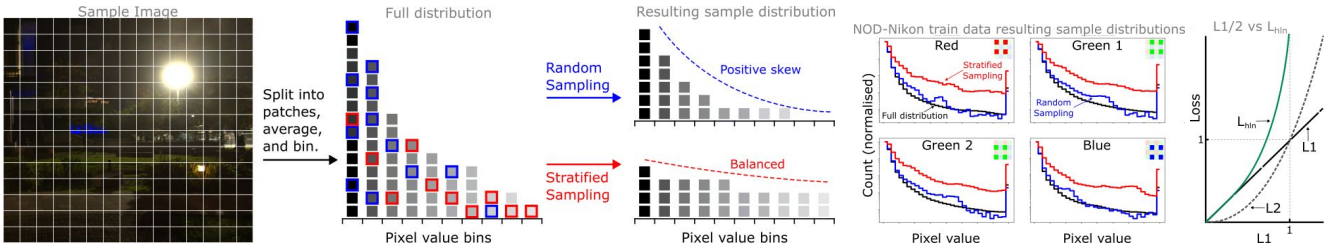
Training ReRAW needs paired RGB and RAW images. These can be created by capturing RAW images with a camera and using the camera's ISP or a generic ISP to produce the paired RGB images. The training set is made up of small image patches taken from each RGB image, along with the matching patch from the paired RAW image. The network takes one RGB patch and the full RGB image it came from as inputs.

The pixel distribution in RAW images is in usual cases (natural scenes) skewed toward darker values, and especially in nighttime images. Examining pixel distributions from all patches in a dataset shows a bias toward low-intensity values. This bias remains even when sampling a subset of patches randomly from the RAW image dataset. Training an RGB-to-RAW converter would therefore tend to prioritize reconstructing darker pixels over brighter regions, which may contain important information useful for other high-level computer vision tasks.

To address this bias, we propose a *stratified sampling* technique to create a paired RAW to RGB patch dataset that better balances the pixel distributions in both domains. This aims to improve reconstruction performance, especially in the brighter regions of the image. The process and its impact on pixel distribution are shown in Fig. 3a. The steps for the stratified sampling method are listed below:

1. Split each RGB image in the dataset into patches and compute the average brightness for each color channel.
2. Bin the patches based on their average brightness for each channel, resulting in three vectors of binned patches. We use 10 bins: $[0 - 0.1), ..., [0.9 - 1.0)$.
3. Uniformly select one bin, then uniformly pick an RGB patch from that bin and its corresponding RAW patch.

The above steps describe our method for selecting an RGB-RAW patch pair from the paired images. We repeat this process multiple times for each color channel and each image in the dataset to build our training set. Compared to random sampling, the stratified sampling method results in a more even pixel intensity distribution, as shown in Fig. 3b. While it is not possible to achieve a perfectly uniform distribution (as we sample full 4-channel pixel values, not individual channels), this method significantly improves uniformity.




(a) Stratified vs random sampling illustration. (b) Sampling on NOD-Nikon. (c) Losses

Figure 3. (a) Comparison of random vs stratified sampling for RGB-to-RAW conversion training data preparation. As explained in 3.4, random sampling results in a pixel intensity distribution similar to the original image, while stratified sampling results in a much more balanced pixel intensity distribution. (b) Histograms show pixel value distributions for each color channel from patches of NOD-Nikon RAW dataset for random (red), full (black) or stratified (blue) sampling methods showing the desired effect. (c) Comparisons of losses.

# 4. Experiments

We perform extensive experiments analyzing the performance of our RGB-to-RAW reconstruction network ReRAW. We benchmark our network on several challenging RAW datasets (daytime and nighttime) and against competing methods and show state-of-the-art performance. Further, we apply ReRAW to convert large labeled RGB datasets to sensor-specific RAW, and show that training RAW object detectors (OD) from scratch on combined synthetic and ground-truth labeled RAW data outperforms traditional RGB data pretraining and finetuning. We demonstrate this via a strict 1-to-1 comparison, over three different object detectors and over two different RAW/RGB datasets (one day and one nighttime) to show the validity of our training recipe.

## 4.1. Datasets

**RGB-to-RAW conversion** We utilize five different RAW datasets. From the MIT-Adobe FiveK [4] collection we select images taken with the Nikon D700 and Canon EOS 5D SLR cameras to create two datasets of 542 and 707 RAW images, respectively. We name these: FiveK-Nikon and FiveK-Canon. We also utilize the NOD [47] RAW nighttime dataset consisting of two sets of images captured by a Nikon D750 and Sony RX100 VII SLR cameras. We named these as NOD-Nikon and NOD-Sony, each consisting of 4.0k and 3.8k images, respectively. Finally, we utilise PASCALRAW [48], comprising of 4.2k RAW images captured by a Nikon D3200 DSLR camera.

For all RAW datasets, we use *rawpy* [53] to convert the RAW files to RGB images to create RAW-RGB pairs at full resolution. We use a 80/20 train/test split.

**Object Detection** We utilise PASCALRAW and NOD-Nikon RAW OD datasets to benchmark our models trained on synthetic RAW images. PASCALRAW contains mostly daytime images while NOD-Nikon contain strictly nighttime images. Both datasets are labeled with objects of 3 classes (person, car and bicycle).

To create our large synthetic labeled RAW datasets, we utilise the BDD100K [65] autonomous driving OD dataset. We select only images that contain at least one of the 3 classes of interest, and further split this into daytime and nightime images, using the provided image meta-labels. We extract a 3 class OD *daytime subset* of 36.5k images with 476k instances, and a 3 class OD *nighttime subset* of 27.5k images and 263k annotations.

Utilising PASCALRAW, the daytime RAW OD dataset, and the daytime BDD subset, we create three variations:

1. **BDD-RGB**: contains a mix of daytime BDD RGB images and the ground truth RGB images from the PASCALRAW train split.
2. **BDD-ReRAW-R**: contains daytime BDD RGB images converted to synthetic RAW by the ReRAW model trained on PASCALRAW via *random sampling* patch selection (ReRAW-R), combined with the ground truth RAW images from the PASCALRAW train split.
3. **BDD-ReRAW-S**: contains daytime BDD RGB images converted to synthetic RAW by the ReRAW model trained on PASCALRAW via *stratified sampling* patch selection (ReRAW-S), combined with the ground truth RAW images from the PASCALRAW train split.

Utilising NOD-Nikon, we prepare the same 3 variations of datasets, however using the nighttime BDD subset, and the ReRAW variations trained on NOD-Nikon. A visualisation of the converted images is shown in Supplementary Fig. **??**.

## 4.2. RGB-to-RAW Reconstruction

**Training Setup** We sample about six $68 \times 68$ RGB patches and their corresponding $32 \times 32$ RAW patches per image pair, once randomly and once using our stratified sampling method to create two separate training subsets per RGB-RAW dataset. We use each of these to train a separate ReRAW variant, ReRAW-R utilising the random sampled subset and ReRAW-S utilising the stratified sampled subset. Each target RAW patch is black-level subtracted and max-normalised. The full context RGB image is down-

| Dataset | NOD - Nikon | | NOD - Sony | | FIVEK - Nikon | | FIVEK - Canon | | PASCALRAW | |
|---|---|---|---|---|---|---|---|---|---|---|
| Metric | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| CycleR2R [31] | 24.51 | 0.5805 | 22.06 | 0.5069 | 24.60 | 0.8768 | 25.28 | 0.8582 | 26.65 | 0.7785 |
| UNet [10] | 34.58 | 0.9279 | 34.93 | 0.9067 | 27.18 | 0.8882 | 25.83 | 0.8895 | 27.81 | 0.8831 |
| SRISP [49] | 35.04 | 0.8953 | 33.89 | 0.8628 | 24.28 | 0.8313 | 26.34 | 0.8164 | 31.79 | 0.9490 |
| InvISP [61] | 27.98 | 0.8843 | 28.37 | 0.8764 | 27.09 | 0.9142 | 23.81 | 0.8596 | 27.34 | 0.9120 |
| InvISP$^+$ [61] | 37.20 | 0.9708 | 35.86 | 0.9499 | 26.41 | 0.9093 | 26.61 | 0.8995 | 31.07 | 0.9507 |
| ISPLess [39] | 27.27 | 0.8867 | 27.18 | 0.8714 | 27.79 | 0.9173 | 24.86 | 0.8728 | 26.30 | 0.9057 |
| ISPLess$^+$ [39] | 37.14 | 0.9688 | 35.69 | 0.9489 | 27.88 | 0.9093 | 26.85 | 0.8898 | 30.45 | 0.9336 |
| RAW-Diffusion [51] | 39.82 | 0.9804 | 38.22 | 0.9658 | 28.84 | 0.9258 | 28.89 | **0.9333** | 35.34 | 0.9695 |
| **ReRAW-R** | 40.12 | **0.9915** | 38.64 | 0.9929 | **30.52** | **0.9492** | 29.85 | 0.9103 | 38.51 | 0.9860 |
| **ReRAW-S** | **41.00** | 0.9914 | **40.07** | **0.9931** | 30.18 | 0.9466 | **30.45** | 0.9122 | **38.88** | **0.9861** |

Table 1. RGB to RAW reconstruction performance comparison by PSNR (dB) (↑) and SSIM(↑). ReRAW, particularly the stratified sampling variant ReRAW-S, outperforms competing methods. Best result is highlighted in bold, second best underlined.

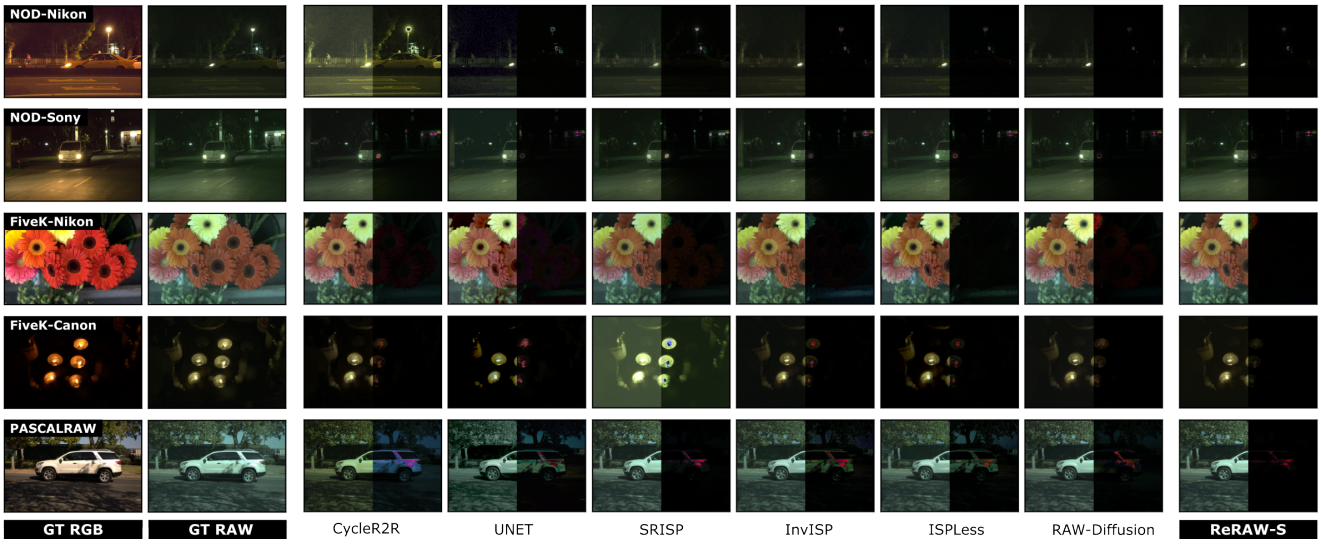



Figure 4. Qualitative comparison of RAW reconstruction for several competing reverse ISP models and ReRAW-S. First two columns show ground truth RGB followed by ground truth RAW. For each model, the input is formed just from the ground truth RGB image. Each row shows one example from each dataset. Each reconstructed image is split in half, where the left half shows the gamma-corrected reconstructed image, and right half shows an error map vs ground truth RAW image. Best seen in color.

scaled to $128 \times 128$ and randomly cropped to 0.9 of its area for each patch sample. We train each ReRAW model using the Adam optimizer [27] with a batch size of 32. Training is done for 128 epochs using cosine annealing with warm restarts every 16 epochs, with a starting learning rate of $10^{-3}$ and decaying to $10^{-5}$.

**Evaluation** For each trained model we convert the RGB test set for each RGB-RAW datasets into their synthetic RAW counterparts at full resolution and evaluate both PSNR and SSIM [60] (Structural Similarity Index Masure) compared to the original RAW images. We average PSNR and SSIM for all reconstructed images in each dataset and report the results.

**Results** We compare ReRAW with several state-of-the-art reverse ISPs: CycleR2R [31], a CycleGAN based method, UNet [10], SRISP [49], InvISP [61], ISPLess [39], and RAW-Diffusion [51], a diffusion-based method. For SRISP, we use the mean global feature of all training images as test-time reference. For InvISP and ISPLess, we also include variants where during the inverse process, the ground truth RGB image is used as input, denoted as InvISP$^+$ and ISPLess$^+$. The conversion results on all the five RAW datasets listed before are shown in Table 1. ReRAW outperforms all listed reverse ISPs in terms of PSNR and SSIM. RAW-Diffusion achieves second best overall results. We report two variations of our model: ReRAW-R - trained with a dataset of patches selected randomly; and ReRAW-

| Loss | NOD Nikon | NOD Sony | FIVEK Nikon | FIVEK Canon | PASCAL RAW |
|---|---|---|---|---|---|
| $\mathcal{L}_1$ | 40.71 | 39.97 | **31.35** | 29.85 | 38.87 |
| $\mathcal{L}_2$ | 40.65 | 39.43 | 28.30 | 27.93 | 38.34 |
| $\mathcal{L}_{hln}$ | **41.00** | **40.07** | 30.18 | **30.45** | **38.88** |

Table 2. Comparison on different loss function and their effect of ReRAW reverse ISP conversion PSNR (dB) (↑). Our proposed $\mathcal{L}_{hln}$ loss achieves tops performance on the majority of datasets.

| Global Context Encoder | Gamma Predictor no. heads | Gamma Scaling Encoder | NOD Nikon | FIVEK Canon |
|---|---|---|---|---|
| | 1 | | 36.04 | 26.55 |
| ✓ | 1 | | 40.95 | 29.5 |
| | 2 | | 36.10 | 27.44 |
| ✓ | 2 | | 40.98 | 29.27 |
| ✓ | 2 | ✓ | 40.87 | 29.75 |
| ✓ | 10 | ✓ | **41.00** | **30.45** |

Table 3. Ablation study on the components of ReRAW and their effect on conversion PSNR (dB) (↑). Global Context Encoder's impact is high, while both increasing the no. of heads and adding the scaling encoder further boosts performance. When (M) has multiple heads and (G) is not used, the outputs are just averaged.

S - trained on a dataset of patches selected via our stratified sampling method. ReRAW-S achieves the highest reconstruction PSNR for NOD-Nikon, NOD-Sony, FiveK-Canon, and PASCALRAW datasets. Also, ReRAW-S, enabled by the stratified sampling technique, reconstructs the high intensity RAW pixel values better compared to ReRAW-R, as can be seen in Supplementary Fig. **??**.

Figure 4 shows qualitative results of synthetic RAW images reconstructed from original RGB images using our models and competing reverse ISPs, compared to their ground truth RAWs. Due to both the stratified sampling technique and logarithm-based loss function, highlighted regions of the RAW images are better reconstructed by ReRAW-S compared to competing methods. This better reconstruction can also be visualised when plotting the ground truth RAW pixel values vs synthetic RAW pixel values for all RAW color channels, as shown in Supplementary **??**. ReRAW-S achieves a more linear relationship between predicted and real pixel values, boosting a higher PSNR.

**Ablation Study** We perform experiments to study how each component and learning heuristic impacts the performance of ReRAW.

We test the impact of different loss functions on training ReRAW, as shown in Table 2. Using the stratified-sampling training set, we train the model with $\mathcal{L}_1$, $\mathcal{L}_2$, and our proposed $\mathcal{L}_{hln}$ loss functions. The logarithm-based loss performs best across datasets, likely because it penalizes poorly reconstructed high-value pixels more strongly than $\mathcal{L}_1$ and $\mathcal{L}_2$, leading to better overall conversion performance.

We also ablate modules and modify network hyperparameters, with the results shown in Table 3. The context encoder proves important for PSNR performance, since it proves global color modulation parameters can be extracted from the converted RGB scene. Additionally, the multi-head architecture allows the model to convert various gamma-corrected patches and select the best ones.

### 4.3. Object Detection

**Training Setup** We evaluate several recipes of training small object detectors for running on the edge. We train 3 different single-stage object detectors: RTMDet-s [41], YOLO-X-s [16], and SSD with a MobileNet-v2 backbone [55], each designed for efficient, real-time object detection through simplified architectures that optimize inference speed and accuracy. We train each detector in two stages: a pretraining stage on variations of a large custom dataset extracted from BDD100K, as listed in Section 4.1, and a finetuning stage on small real-world datasets of interest. Four different pretraining/finetuning combinations are tested in order to fairly evaluate the differences in performance between RGB and RAW trained OD models, and on two datasets: PASCALRAW (daytime) and NOD-Nikon (nighttime). We pretrain each detector from scratch for 50 epochs on the full custom BDD dataset (BDD-RGB or BDD-ReRAW-R/S), using stochastic gradient descent (SGD) with a cosine annealing schedule. Base learning rates are $0.001$ for RTMDet, $0.002$ for YOLOX and $0.015$ for SSD, decaying to $0.1\times$. Random flip, scale and mosaic are applied as augmentations only during pretraining. Each detector was then finetuned for 8 epochs on the ground-truth train set of PASCALRAW or NOD-Nikon, with starting learning rate of $0.1\times$ of base and decaying to $0.01\times$, also on a cosine annealing schedule. We keep the training heuristics identical per detector in order to allow a fair 1-to-1 comparison between each training dataset combinations.

**Evaluation** Each trained detector was evaluated on the PASCALRAW and NOD-Nikon RGB or RAW test sets, depending on the modality of the finetuning set, on mean Average Precision (mAP, mAP50, and mAP75).

**Results** The object detection training results are shown in Table 4. The results under the PASCALRAW columns have been obtained using the daytime BDD-RGB and BDD-ReRAW datasets, and for the NOD-Nikon column, using the nighttime sets (as explained in Section 4.1).

The line a) result for each detector represents the baseline traditional pipeline of RGB pretraining and finetuning.

| Model | Pretraining | Finetuning | PASCALRAW | | | NOD - Nikon | | |
|---|---|---|---|---|---|---|---|---|
| | | | mAP | mAP50 | mAP75 | mAP | mAP50 | mAP75 |
| RTMDet-s [41] | a) BDD-RGB | GT-RGB | 57.62 | 86.60 | 60.69 | 20.46 | **39.74** | 18.39 |
| | b) BDD-RGB | GT-RAW | 56.20 | 85.65 | 58.33 | 20.21 | 38.45 | 18.91 |
| | c) BDD-ReRAW-R | GT-RAW | 62.44 | **91.12** | 65.23 | **21.53** | 38.64 | **20.76** |
| | d) **BDD-ReRAW-S** | **GT-RAW** | **63.19** | 90.45 | **66.14** | 21.09 | 38.14 | 20.49 |
| YOLOX-s [16] | a) BDD-RGB | GT-RGB | 65.36 | 91.45 | 71.70 | 27.14 | 49.58 | 25.45 |
| | b) BDD-RGB | GT-RAW | 64.00 | 90.33 | 70.06 | 27.30 | 49.76 | 26.13 |
| | c) BDD-ReRAW-R | GT-RAW | 62.64 | 90.88 | 67.75 | 27.09 | 50.52 | 25.49 |
| | d) **BDD-ReRAW-S** | **GT-RAW** | **65.85** | **91.76** | **71.73** | **29.03** | **52.92** | **27.49** |
| SSD [55] | a) BDD-RGB | GT-RGB | 62.50 | 90.63 | 66.38 | 22.97 | 40.98 | 21.83 |
| | b) BDD-RGB | GT-RAW | 62.22 | 90.53 | 65.67 | 23.06 | 41.05 | **22.64** |
| | c) BDD-ReRAW-R | GT-RAW | 60.96 | 89.60 | 64.53 | 22.56 | 40.28 | 22.30 |
| | d) **BDD-ReRAW-S** | **GT-RAW** | **63.09** | **90.98** | **67.38** | **23.40** | **41.39** | 22.32 |

Table 4. Object detection training results: 3 OD models $\times$ 4 training variants. Training heuristic d), involving pretraining on a mix high quality synthetic RAW dataset converted by ReRAW and ground truth RAW data (BDD-ReRAW-S), then finetuning on a RAW dataset of interest, generally achieves the highest performance in terms of mAP, compared to other training heuristics, including a full RGB pipeline.

Line b) involves taking the RGB pretrained model and finetuning it on RAW images (with the 2 green channels averaged). Although this is an immediate solution for adapting foundation RGB models to RAW, this method generally underperforms due to the domain gap. For lines c) and d), the detectors have been pretrained on synthetic large RAW image datasets converted from BDD-RGB using ReRAW-R (c) and ReRAW-S (d), and finetuned on real RAW images.

Training heuristic d) always outperforms the traditional RGB pipeline a), on a 1-to-1 comparison, for multiple detectors and on both a daytime, and a more difficult nighttime dataset. Additionally, training heuristic c) underperforms both d) and a) (for YOLOX and SSD), due to the large synthetic pretraining RAW dataset being lower fidelity (PSNR) than d). This underscores that the quality of the reverse ISP used to generate the large synthetic RAW pretraining dataset is important. The second performing training heuristic for the daytime dataset (PASCALRAW) is the RGB pipeline a). This is because the pixel distribution of daytime RGB images is closer to their original RAW versions. In contrast, for the nighttime dataset (NOD-Nikon), the second-best heuristic is b) or c), showing that in low-light conditions, RAW is optimal.

## 5. Discussion and Future Work

Our proposed stratified sampling technique helps in boosting the conversion performance of ReRAW, and proved that training data curation is beneficial even for low-level image tasks such as RGB to RAW conversion. We plan to explore other sampling methods such as filtering training patches based on high dynamic range, or high entropy, that might further boost conversion performance.

The experiments in Table 4 showed that maximizing object detection accuracy of small detectors on the edge operating on unprocessed RAW image signal directly, the training recipe listed in d) is the most salient. The RTMDet detector showed the highest sensitivity to input domain, where training on PASCALRAW yielded a large variability in mAP results, whilst YOLOX and SSD showed less variability. This suggests that developing detector architectures specifically tailored for RAW images is a promising direction for future research. We also acknowledge that the RTMDet results on NOD-Nikon show preference to the lower PSNR synthetic RAW dataset (BDD-ReRAW-R), which is surprising. This shows a limitation of the PSNR metric when relating synthetic RAW conversion performance and downstream task accuracy, and it's worth exploring further.

## 6. Conclusion

We introduced ReRAW, a state-of-the-art high-PSNR reverse ISP for converting RGB images into sensor-specific RAW. ReRAW achieves the highest reconstruction accuracy against competing state-of-the-art methods, for five different datasets, due to its unique multi-head architecture predicting RAW image candidates in gamma space, and stratified training data sampling technique. Using ReRAW to generate high-quality synthetic RAW datasets for pretraining OD models and fine-tuning on real RAW data results in superior performance compared to models trained on traditional RGB pipelines. This method, thanks to ReRAW's high reconstruction accuracy, optimizes model training for edge devices, bypassing the ISP hence saving energy and time, and enhancing OD accuracy over standard RGB workflows.

## References


[1] Mahmoud Afifi, Abdelrahman Abdelhamed, Abdullah Abuolaim, Abhijith Punnappurath, and Michael S Brown. Cie xyz net: Unprocessing images for low-level computer vision tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4688–4700, 2021. 2

[2] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. *CoRR*, abs/1811.11127, 2018. 2

[3] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11036–11045, 2019. 2

[4] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5

[5] Ayan Chakrabarti, Daniel Scharstein, and Todd E Zickler. An empirical camera model for internet color vision. In *British Machine Vision Conference*, 2009. 2

[6] Ayan Chakrabarti, Ying Xiong, Baochen Sun, Trevor Darrell, Daniel Scharstein, Todd Zickler, and Kate Saenko. Modeling radiometric uncertainty for vision with tone-mapped color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2185–2198, 2014. 2

[7] Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. Instance segmentation in the dark. *International Journal of Computer Vision*, 131(8):2198–2218, 2023. 2

[8] Marcos V Conde, Steven McDonagh, Matteo Maggioni, Ales Leonardis, and Eduardo Pérez-Pellitero. Model-based image signal processors via learnable dictionaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 481–489, 2022. 2

[9] Marcos V Conde, Radu Timofte, Yibin Huang, Jingyang Peng, Chang Chen, Cheng Li, Eduardo Pérez-Pellitero, Fenglong Song, Furui Bai, Shuai Liu, and others. Reversed image signal processing and RAW reconstruction. AIM 2022 challenge report. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, pages 3–26. Springer, 2022. 2

[10] Marcos V Conde, Radu Timofte, Yibin Huang, Jingyang Peng, Chang Chen, Cheng Li, Eduardo Pérez-Pellitero, Fenglong Song, Furui Bai, Shuai Liu, et al. Reversed image signal processing and raw reconstruction. aim 2022 challenge report. In *European Conference on Computer Vision*, pages 3–26. Springer, 2022. 6

[11] Ziteng Cui and Tatsuya Harada. Raw-adapter: Adapting pre-trained visual model to camera raw images. In *European Conference on Computer Vision*, pages 37–56. Springer, 2024. 2

[12] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 643–652. Association for Computing Machinery, 2023. 2

[13] Ujjal Kr Dutta. Seeing objects in dark with continual contrastive learning. In *European Conference on Computer Vision*, pages 286–302. Springer, 2022. 2

[14] Eilertsen, Gabriel, Joel Kronander, Gyorgy Denes, Rafał Mantiuk, and Jonas Unger. HDR image reconstruction from a single exposure using deep CNNs. *ACM Transactions on Graphics (TOG)*, 36(6), 2017. 4

[15] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1039–1048, 2017. 2

[16] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021. 7, 8

[17] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 2

[18] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 2

[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[20] Michael D Grossberg and Shree K Nayar. Determining the camera response from images: What is knowable? *IEEE Transactions on pattern analysis and machine intelligence*, 25(11):1455–1467, 2003. 2

[21] Patrick Hansen, Alexey Vilkin, Yury Khrustalev, James Imber, David Hanwell, Matthew Mattina, and Paul N. Whatmough. Isp4ml: Understanding the role of image signal processing in efficient deep learning vision systems, 2021. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[23] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 2

[24] Andrew G Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[25] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin, and Michael S Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2289–2302, 2012. 2

[26] Woohyeok Kim, Geonu Kim, Junyong Lee, Seungyong Lee, Seung-Hwan Baek, and Sunghyun Cho. Paramisp: Learned forward and inverse isps using camera parameters, 2024. 2

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 6

[28] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2810–2819, 2019. 2

[29] Yunsheng Li, Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Lu Yuan, Zicheng Liu, Lei Zhang, and Nuno Vasconcelos. Micronet: Improving image recognition with extremely low flops. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 468–477, 2021. 2

[30] Zhihui Li, Pengfei Xu, Xiaojun Chang, Luyao Yang, Yuanyuan Zhang, Lina Yao, and Xiaojiang Chen. When object detection meets knowledge distillation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10555–10579, 2023. 2

[31] Zhihao Li, Ming Lu, Xu Zhang, Xin Feng, M. Salman Asif, and Zhan Ma. Efficient visual computing with camera RAW snapshots. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2024. 2, 6

[32] Siyuan Liang, Hao Wu, Li Zhen, Qiaozhi Hua, Sahil Garg, Georges Kaddoum, Mohammad Mehedi Hassan, and Keping Yu. Edge yolo: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23 (12):25345–25360, 2022. 2

[33] Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang. Cameranet: A two-stage framework for effective camera isp learning. *IEEE Transactions on Image Processing*, 30:2248–2262, 2021. 2

[34] Stephen Lin and Lei Zhang. Determining the radiometric response function from a single grayscale image. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 66–73. IEEE, 2005. 2

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[36] Jiaming Liu, Chihao Wu, Yuzhi Wang, Qin Xu, Yuqian Zhou, Haibin Huang, Chuan Wang, Shaofan Cai, Yifan Ding, Haoqiang Fan, and Jue Wang. Learning raw image denoising with bayer pattern unification and bayer preserving augmentation. *CoRR*, abs/1904.12945, 2019. 1

[37] Shuai Liu, Chaoyu Feng, Xiaotao Wang, Hao Wang, Ran Zhu, Yongqiang Li, and Lei Lei. Deep-flexisp: A three-stage framework for night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1211–1220, 2022. 2

[38] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2

[39] William Ljungbergh, Joakim Johnander, Christoffer Petersson, and Michael Felsberg. *Raw or Cooked? Object Detection on RAW Images*, page 374–385. Springer Nature Switzerland, 2023. 1, 6

[40] William Ljungbergh, Joakim Johnander, Christoffer Petersson, and Michael Felsberg. Raw or cooked? object detection on raw images. In *Scandinavian Conference on Image Analysis*, pages 374–385. Springer, 2023. 2

[41] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors, 2022. 7, 8

[42] Bruce A Maxwell, Sumegha Singhania, Heather Fryling, and Haonan Sun. Log rgb images provide invariance to intensity and color balance variation for convolutional networks. In *BMVC*, pages 635–642, 2023. 2

[43] Bruce A Maxwell, Sumegha Singhania, Avnish Patel, Rahul Kumar, Heather Fryling, Sihan Li, Haonan Sun, Ping He, and Zewen Li. Logarithmic lenses: Exploring log rgb data for image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17470–17479, 2024. 2

[44] T. Mitsunaga and S.K. Nayar. Radiometric self calibration. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, pages 374–380 Vol. 1, 1999. 4

[45] Tomoo Mitsunaga and Shree K Nayar. Radiometric self calibration. In *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, pages 374–380. IEEE, 1999. 2

[46] Igor Morawski, Yu-An Chen, Yu-Sheng Lin, Shusil Dangi, Kai He, and Winston H. Hsu. Genisp: Neural isp for low-light machine cognition, 2022. 1

[47] Igor Morawski, Yu-An Chen, Yu-Sheng Lin, Shusil Dangi, Kai He, and Winston H Hsu. GenISP: Neural ISP for low-light machine cognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 630–639, 2022. 5

[48] Ta David Omid-Zohoor, Alex and Boris. Murmann. Pascalraw: Raw image database for object detection. stanford digital repository. In *Available at: http://purl.stanford.edu/hq050zr7488*, 2014-2015. 5

[49] Junji Otsuka, Masakazu Yoshimura, and Takeshi Ohashi. Self-Supervised Reversed Image Signal Processing via Reference-Guided Dynamic Parameter Selection. *arXiv preprint arXiv:2303.13916*, 2023. 6

[50] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[51] Christoph Reinders, Radu Berdan, Beril Besbinar, Junji Otsuka, and Daisuke Iso. Raw-diffusion: Rgb-guided diffusion models for high-fidelity raw image generation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 2, 6

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. 2

[53] Maik Riechert. Rawpy. 2024. 5

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2

[55] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. 7, 8

[56] Eli Schwartz, Raja Giryes, and Alex M Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, 2018. 2

[57] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2

[58] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2

[59] Yujin Wang, Tianyi Xu, Fan Zhang, Tianfan Xue, and Jinwei Gu. Adaptiveisp: Learning an adaptive image signal processor for object detection. *arXiv preprint arXiv:2410.22939*, 2024. 2

[60] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13:600 – 612, 2004. 6

[61] Yazhou Xing, Zian Qian, and Qifeng Chen. Invertible image signal processing, 2021. 2, 6

[62] 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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13384–13393, 2023. 1

[63] 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. 2

[64] Masakazu Yoshimura, Junji Otsuka, Atsushi Irie, and Takeshi Ohashi. Dynamicisp: dynamically controlled image signal processor for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12866–12876, 2023. 2

[65] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642, 2018. 5

[66] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2696–2705, 2020. 2

[67] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 2

[68] Xiangyu Zhang, Ling Zhang, and Xin Lou. A raw image-based end-to-end object detection accelerator using hog features. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(1):322–333, 2022. 1

[69] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *CoRR*, abs/1511.08861, 2015. 4