SONY

Rawgment: Noise-Accounted RAW Augmentation Enables Recognition in a Wide Variety of Environments

Masakazu Yoshimura, Junji Otsuka, Atsushi Irie, and Takeshi Ohashi

(Sony Group Corporation)

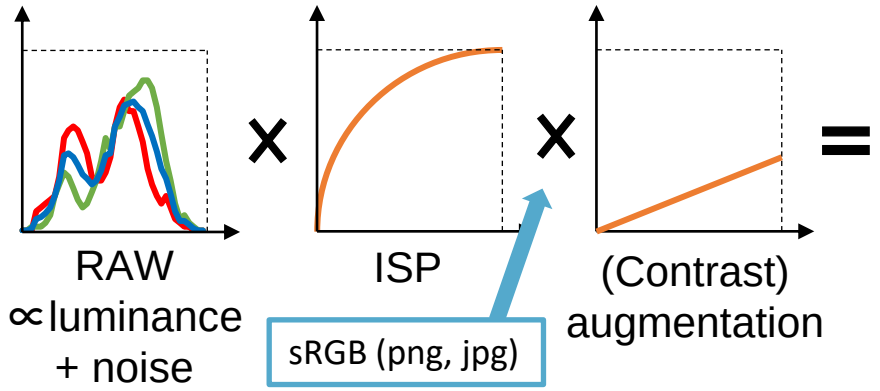
WED-PM-158

Quick Summary

Proposed Concept

Achieve recognition in any environments (dark, blurry, HDR, …) with augmentation

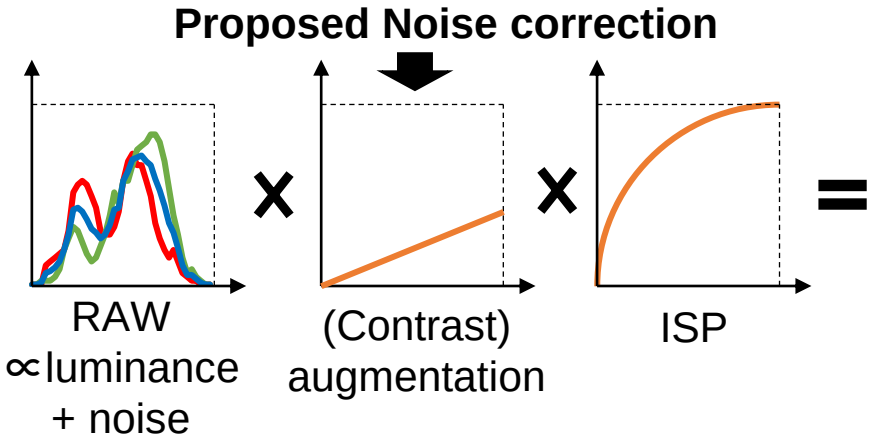
- Usual training pipeline



× **unrealistic pixel distribution**

× **unrealistic noise**

- Proposed Noise-accounted RAW augmentation



✓ **realistic pixel distribution**

✓ **realistic noise**

Results

Evaluated with human detection in challenging environment

augmentation		noise	mAP@0.5:0.95 [%] black-box ISP	simple ISP simplest	simple ISP parameterized
Color	after	-	45.2	19.3	-
	before (ours)	-	-	40.9	44.4
		w/o prior	-	43.5	47.7
		ours	-	**44.6**	**48.1**
Color + Blur	after	-	46.8	20.4	-
	before (ours)	-	-	43.3	43.8
		ours†	-	43.4	47.9
		ours	-	**45.3**	**48.3**

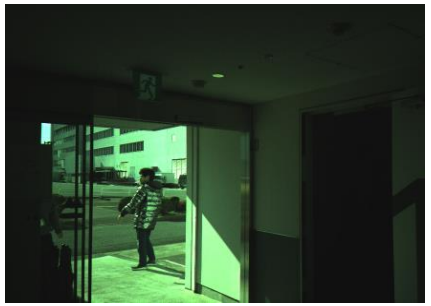
†: The noise alignment is only applied to the color jitter augmentation.

Double the accuracy with the same ISP setting

Improved accuracy even with a simple ISP

Introduction

Standard RGB (sRGB) images (png, jpg, …) are heavily preprocessed images



RAW image
(image sensor's output)

$\propto$ luminance (+ noise)

Image Signal Processor (ISP)

digital gain
color correction
auto-white balance
denoise
tone mapping
gamma
…

sRGB image (png, jpg)

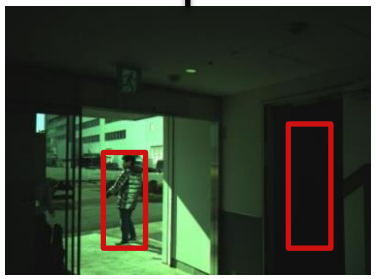
$\not\propto$ luminance

RAW images have physically meaningful values and suitable for environment conversion

Objective

Create recognition models that work in any environments with easy RAW training data and augmentation

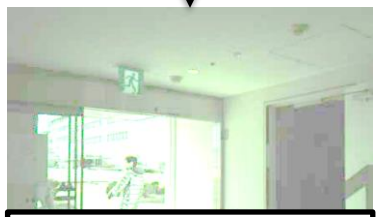
No need for annotation to challenging environment



original data with annotation

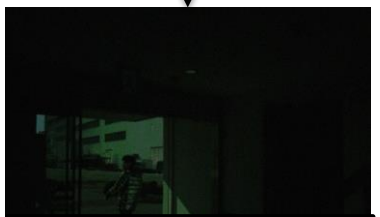
Augmentation

simulate realistic challenging environment

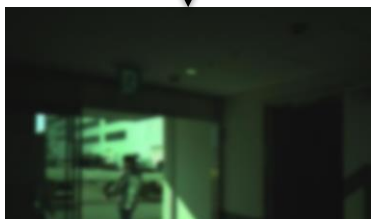


if light environment

if dark but brighten with analog gain



if dark

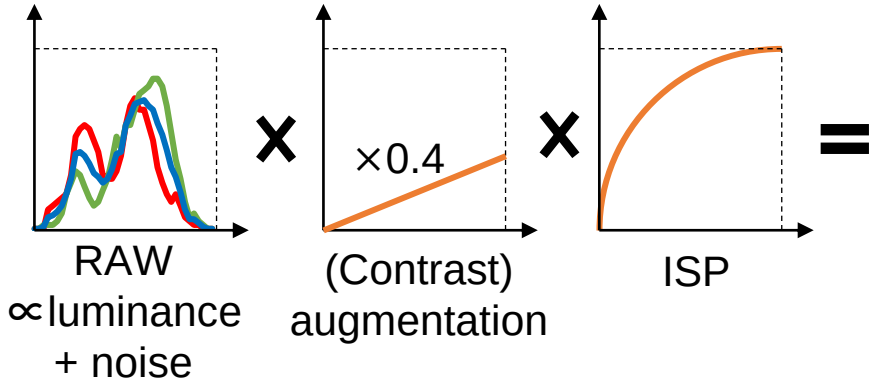


if blurry

Proposed Method

1. To ensure realistic pixel intensity distribution

- Prepare RAW data and apply augmentation **before** applying an ISP

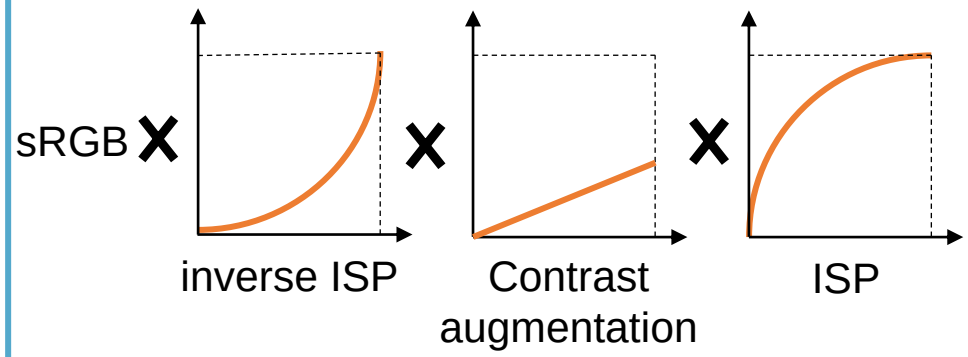


✓ **pixel distribution that can be taken if the environment light intensity is 0.4 times.**

- Although several previous works like UPI proposed similar idea for image restoration,

1. We want to emphasize the importance again, especially to image recognition field
2. Differently, we recommend starting from RAW instead of sRGB
3. Differently, we recommend applying not only contrast aug. but also other augmentations like blur aug. before ISP

UPI (T. Brooks et al, CVPR2019)



problem:

- sRGB contains less info. (8 bit)
  - ➡ Quantization error with small contrast factor like 0.01

  Day (~10,000 lux) to night (~1 lux) conversion needs very small contrast factor
- Difficulty of inverting ISP

Proposed Method

2. To ensure realistic pixel noise

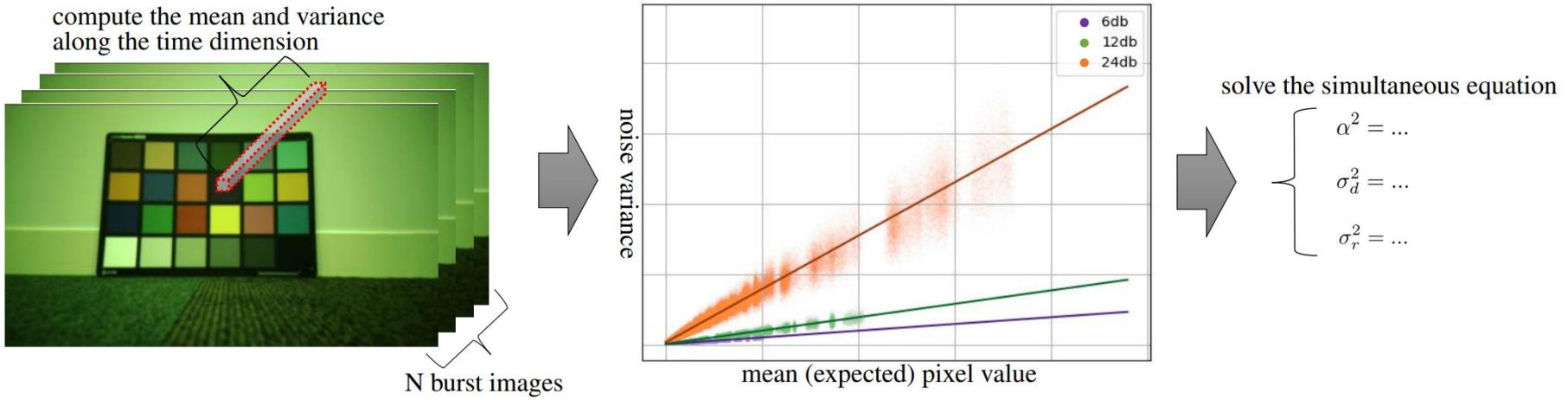
- Noise amount is corrected after augmentations

x	pixel value of RAW
u	Photon number hitting a photodiode
g	Analog gain
α	Quantum efficiency
σ_d^2	Circuit noise var. before analog gain
σ_r^2	Circuit noise var. after analog gain

1. Start from a well-established noise model (A. Foi et al, TIP2008)

$$x \sim \mathcal{N}\left(\mu_x = g\alpha\bar{u},\ \sigma_x^2 = g^2\alpha^2\bar{u} + g^2\sigma_d^2 + \sigma_r^2\right)$$

2. Calibrate the hyperparameters of the noise model (α, σ_d^2, σ_r^2) per image sensor



(a) burst capture with several analog gains (b) estimate linear relationship of the mean and variance (c) estimate the sensor specific parameters

Proposed Method

2. To ensure realistic pixel noise

3. Derive noise-accounted augmentation formulas

Color Jitter Aug.

$$x_{new} \sim \mathcal{N}\begin{pmatrix} (p_g g)\alpha(p_u \bar{u}), \\ (p_g g)^2\alpha^2(p_u \bar{u}) + (p_g g)^2\sigma_d^2 + \sigma_r^2 \end{pmatrix}$$

$$\sim p_u p_g x_{pre} + \mathcal{N}(0,\ p_u(1-p_u)p_g^2 g\alpha\mu_x + (1-p_u^2)p_g^2 g^2\sigma_d^2 + (1-p_u^2 p_g^2)\sigma_r^2)$$

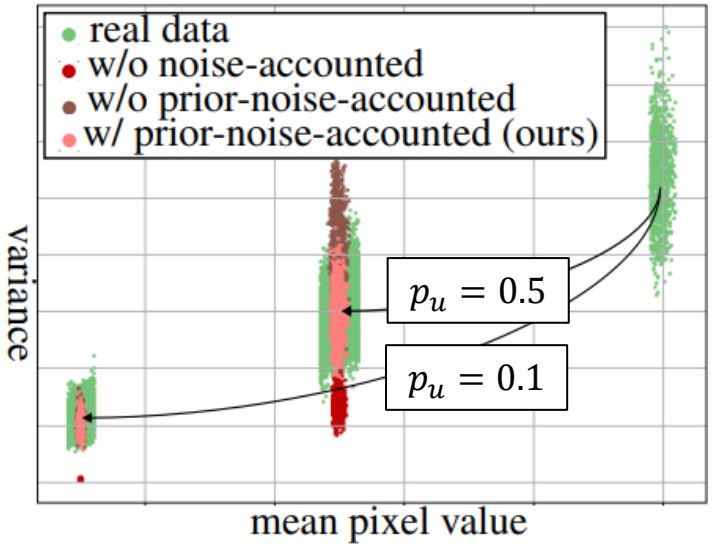
If analog gain g was p_g times and (exposure time)×(environment light intensity) was p_u times

Many previous works like Day-to-night (A. Punnappurath, CVPR2022)

$$x_{new} = p_u p_g x_{pre} + \mathcal{N}(0, p_u p_g x_{pre} + p_g^2 g^2 \sigma_d^2 + \sigma_r^2)$$

Disregard prior input noise

Our formula is more accurate and can be applied to any data including noisy input



Proposed Method

2. To ensure realistic pixel noise

1. Derive noise-accounted augmentation formulas

Blur Aug.

$$x_{new} \sim \mathcal{N}\left(g\alpha \sum_k w_k \bar{u_k},\ g^2\alpha^2 \sum_k w_k \bar{u_k} + g^2\sigma_d^2 + \sigma_r^2\right)$$

$$\sim \sum_k w_k x_{pre} + \mathcal{N}(0,\ g\alpha \sum_k (1 - w_k) w_k x_{pre,k} + (1 - \sum_k w_k^2)(g^2\sigma_d^2 + \sigma_r^2))$$

If more blur $\sum_k w_k$ was exist

Previous works

$$x_{new} = \sum_k w_k x_{pre}$$

No works correct noise amount

Evaluation

Training a detector with images in normal environment

Test with images in challenging (dark, blurry, HDR) environments

augmentation		noise	mAP@0.5:0.95 [%] black-box ISP	simple ISP simplest	simple ISP parameterized
Color	after	-	45.2	19.3	-
	before (ours)	-	-	40.9	44.4
		w/o prior	-	43.5	47.7
		ours	-	**44.6**	**48.1**
Color + Blur	after	-	46.8	20.4	-
	before (ours)	-	-	43.3	43.8
		ours†	-	43.4	47.9
		ours	-	**45.3**	**48.3**

†: The noise alignment is only applied to the color jitter augmentation.

More accurate than other noise-dealing method proposed for image restoration

method	AP@0.5:0.95 [%] w/o ISP	w/ ISP
concat [2]	16.5	21.5
aug. + concat [2]	**35.0**	31.6
our aug. + concat [2]	33.7	40.4
K-Sigma [45]	14.3	27.5
K-Sigma [45] + aug.	25.0	34.1
aug. + K-Sigma [45]	26.6	42.1
our aug. + K-Sigma [45]	26.3	44.0
our aug.	32.8	**45.3**

Evaluation

Visualization

ISP	simplest gamma	simplest gamma	simplest gamma	simplest gamma
augmentation order	after ISP	before ISP (ours)	before ISP (ours)	before ISP (ours)
noise-accounted	w/o	w/o	w/o prior	w/ (ours)

Conclusion

Conclusion

- We propose a noise-accounted RAW augmentation
  - Augmentation before ISP
  - accurate noise correction considering prior input noise

Future Direction

- We believe you can use Rawgment without any annotation

1. Record RAW (+ corresponding sRGB) data in normal environment
2. Create pseudo labels with existing recognition models
3. Train a new detector with the pseudo label and Rawgment to detect in challenging environments