Robust Image Denoising of No-Flash Images Guided by Consistent Flash Images

Geunwoo Oh, Jonghee Back, Jae-Pil Heo, Bochang Moon

STRUCT Group Seminar
Presenter: Haowei Kuang
2023.02.05

OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

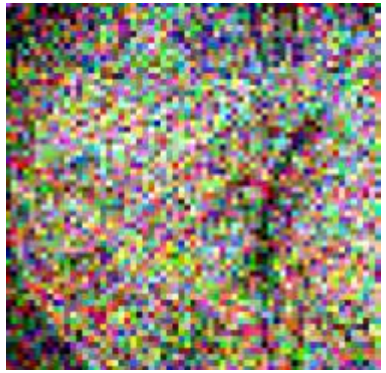
OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

BACKGROUND: KPN

Image Denoising

- Recover clean images from noisy input images
- Noise Model: $I_i^N = I_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(I_i, \sigma_r^2 + \sigma_s I_i)$
- Estimate: $\hat{I}_c = \frac{1}{\sum_{i \in \Omega_c} w_{ci}} \sum_{i \in \Omega_c} w_{ci} \{I_i^N\}$



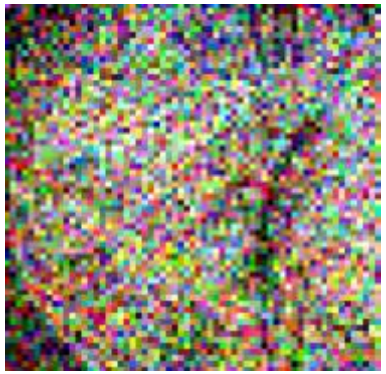
Noisy Image

Clean Image

BACKGROUND

Image Denoising

- Bottleneck:
  - Loss of high-frequency details
  - Blending colors across edges
  - Over-smooth



Noisy Image

Ground Truth

Denoise Result

BACKGROUND

Image Denoising Guided by Flash

- Contain high-frequency details
- Serve as edge-stopping functions
- Bottleneck: Additional image structures

OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

METHOD: Deep Combiner

- For Noisy Image:

$$I_i^N = I_i + \epsilon_i$$

- For Flash/No-flash Pairs:

$$I_c^F - I_i^F = I_c - I_i + \epsilon_{ci}$$

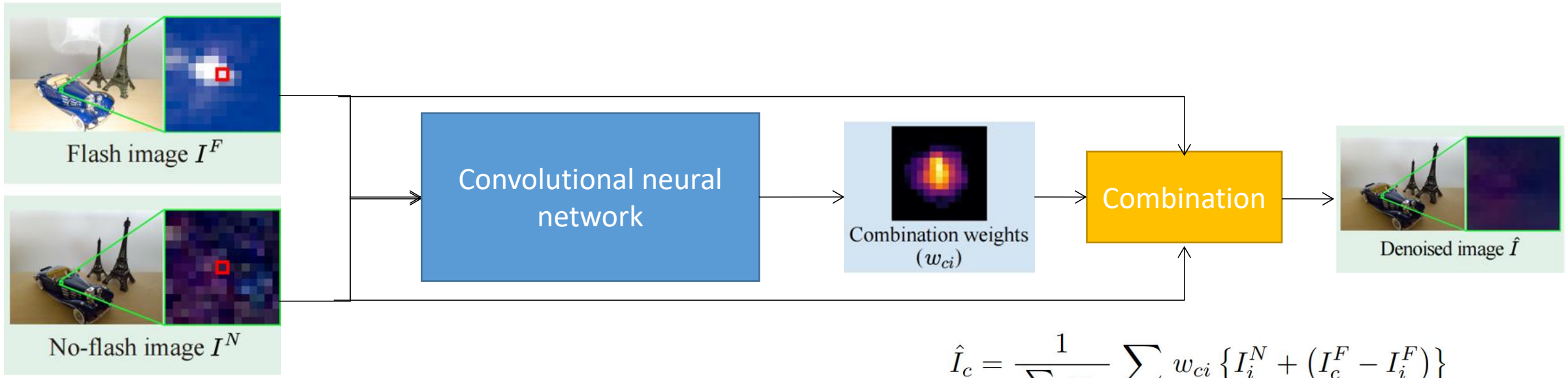
- To estimate the c-th pixel $\hat{I}_c$

$$J_c = \frac{1}{2} w_{cc} \left(I_c^N - \hat{I}_c \right)^2 + \sum_{i \in \Omega_c, i \neq c} w_{ci} \left(I_i^N - \hat{I}_i \right)^2$$

$$+ \sum_{} w_{ci} \left\{ \left(I_c^F - I_i^F \right) - \left(\hat{I}_c - \hat{I}_i \right) \right\}^2,$$

$$\hat{I}_c = \frac{1}{\sum_{i \in \Omega_c} w_{ci}} \sum_{i \in \Omega_c} w_{ci} \left\{ I_i^N + \left(I_c^F - I_i^F \right) \right\}$$

METHOD: Deep Combiner

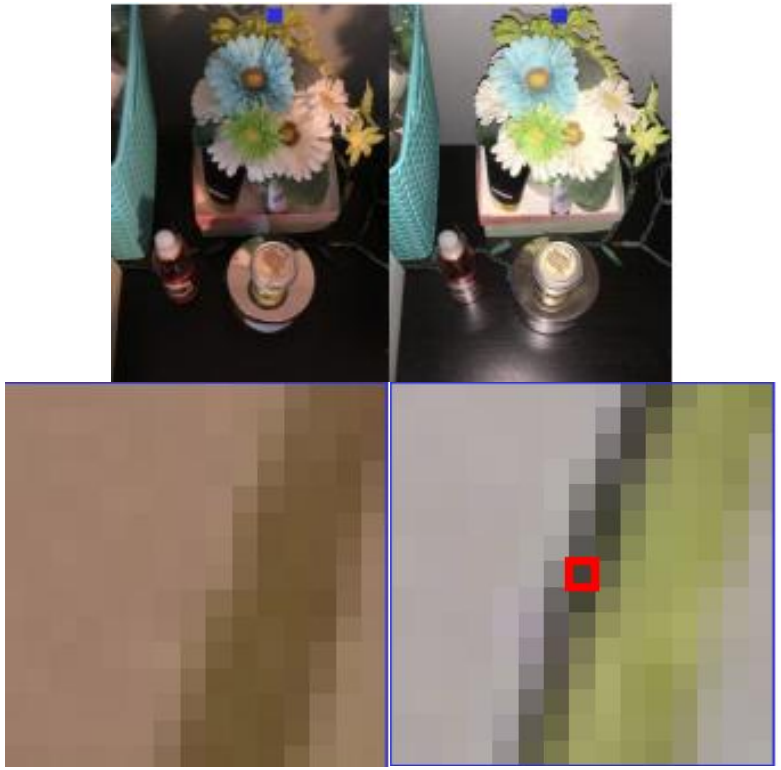


$$\hat{I}_c = \frac{1}{\sum_{i \in \Omega_c} w_{ci}} \sum_{i \in \Omega_c} w_{ci} \left\{ I_i^N + \left(I_c^F - I_i^F \right) \right\}$$

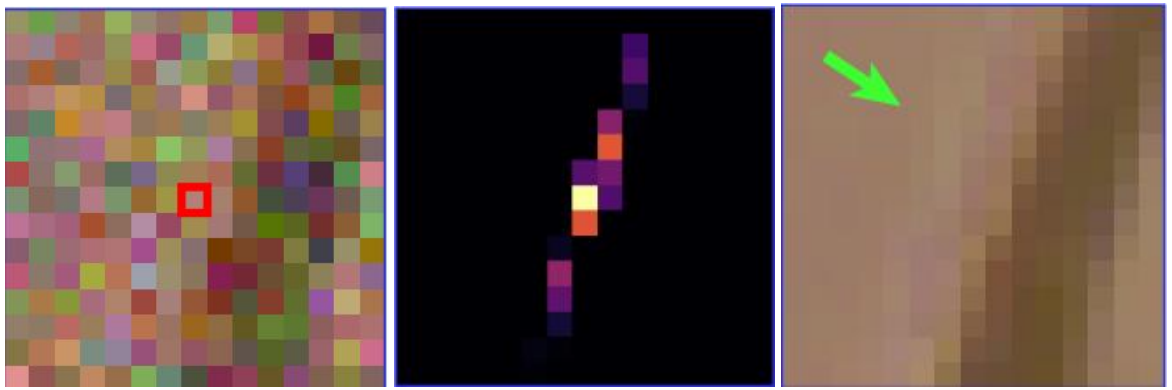
METHOD

Drawbacks

$$I_c^F - I_i^F = I_c - I_i + \epsilon_{ci}$$



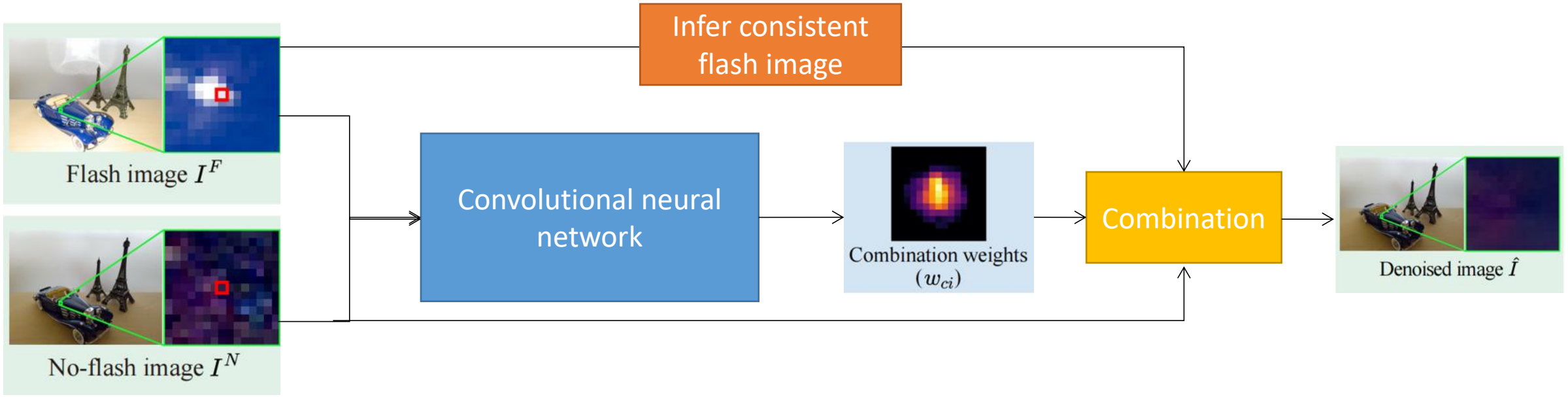
Ground Truth/Flash Image



Noisy Image Combination Weights Result

METHOD

Improvement



METHOD

- Deep Combiner:

$$I_c^F - I_i^F = I_c - I_i + \epsilon_{ci}$$

- RIDFnF:

$$\left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i = I_c - I_i + \epsilon_{ci}$$

- To estimate the c-th pixel $\hat{I}_c$

$$J_c = \frac{1}{2} w_{cc} \left(I_c^N - \hat{I}_c\right)^2 + \sum_{i \in \Omega_c, i \neq c} w_{ci} \left(I_i^N - \hat{I}_i\right)^2$$
$$+ \sum_{i \in \Omega_c, i \neq c} w_{ci} \left[\left\{\left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i\right\} - \left(\hat{I}_c - \hat{I}_i\right)\right]^2$$

- minimized by setting its gradients with respect to $\hat{I}_c$ and $\hat{I}_i$ zero

$$\frac{\partial J_c}{\partial \hat{I}_c} = w_{cc} \left(I_c^N - \hat{I}_c\right) + 2 \sum_{i \in \Omega_c, i \neq c} w_{ci} \left[\left\{\left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i\right\} - \left(\hat{I}_c - \hat{I}_i\right)\right] = 0,$$

$$\frac{\partial J_c}{\partial \hat{I}_i} = -w_{ci} \left(I_i^N - \hat{I}_i\right) + w_{ci} \left[\left\{\left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i\right\} - \left(\hat{I}_c - \hat{I}_i\right)\right] = 0.$$

METHOD

- Setting $\frac{\partial J_c}{\partial \hat{I}_i} = 0$:

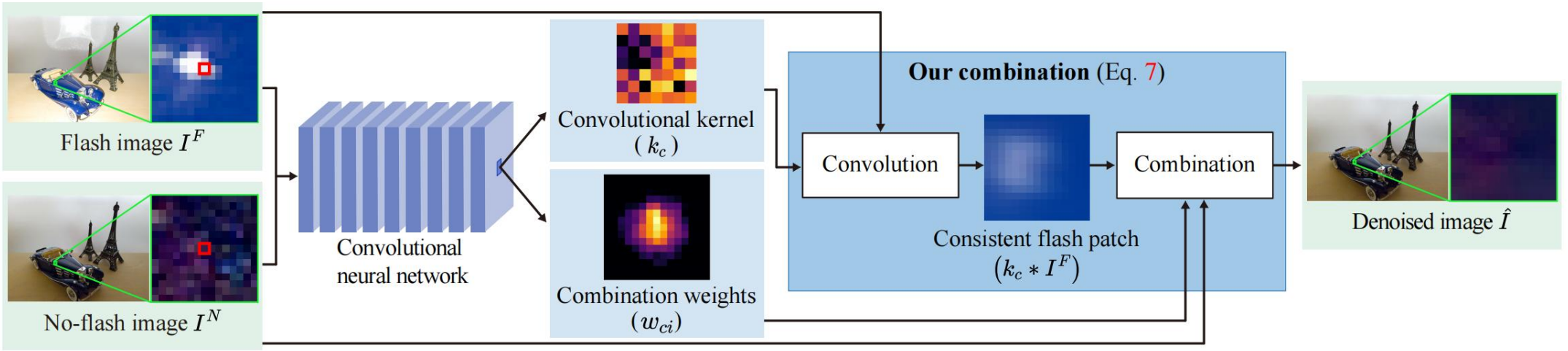
$$\hat{I}_i = \frac{1}{2}\left\{I_i^N - \left(k_c * I^F\right)_c + \left(k_c * I^F\right)_i + \hat{I}_c\right\}$$

- Plug this equation into $\frac{\partial J_c}{\partial \hat{I}_c}$

$$\hat{I}_c = \frac{1}{\sum_{i\in\Omega_c} w_{ci}} \sum_{i\in\Omega_c} w_{ci}\left\{I_i^N + \left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i\right\}$$

METHOD

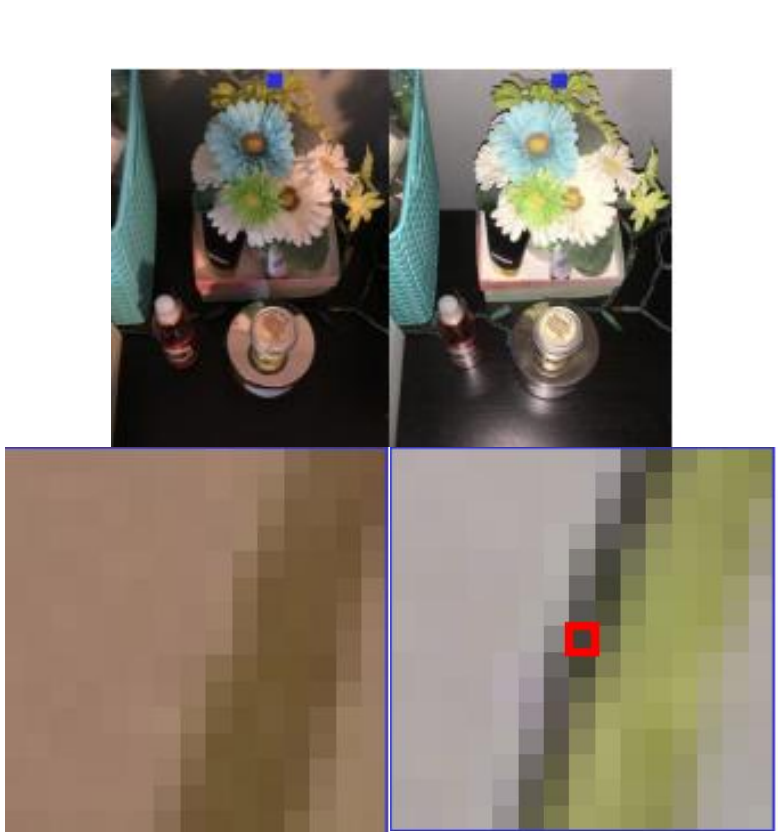
Network Architecture



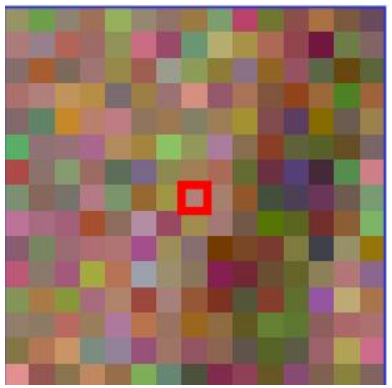
Network Details

- k_c a normalized kernel whose elements are non-negative
- k_c size: (7×7) Ω_c size: (15×15)
- Training loss: L2 Loss
- Trainable parameters: 1.84M

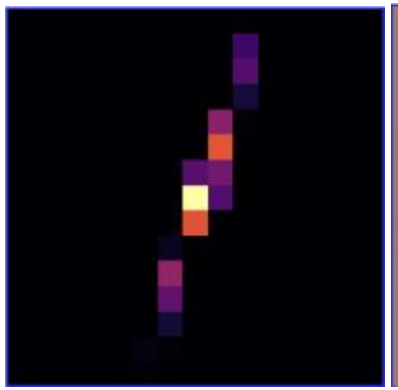
METHOD



Ground Truth/Flash Image

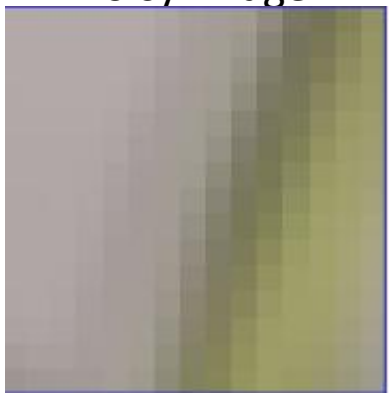


Noisy Image

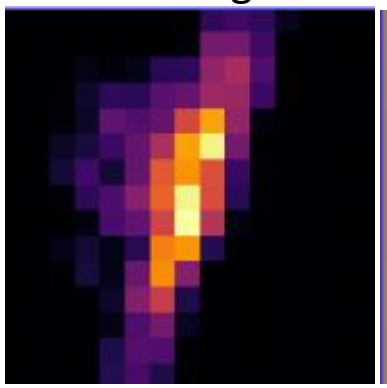


DC Weights

Result of DC



Consistent flash



RIDFnF Weights

Result of RIDFnF

OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

EXPERIMENTS

Datasets: Flash and Ambient Illuminations Dataset(FAID)

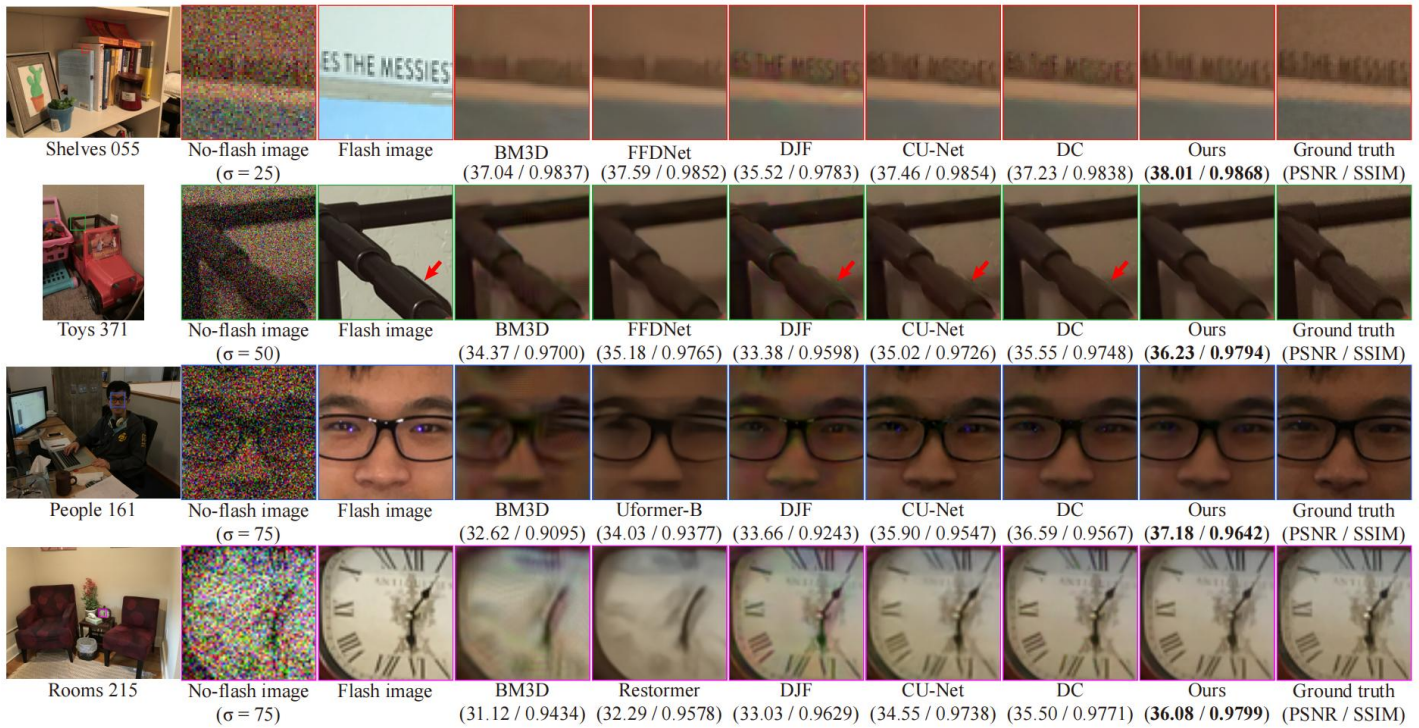
- Includes 2775 flash/no-flash image pairs categorized into six classes
- 2263 for training, 256 for validation, 256 for testing

Training Details

- Training for 50 epochs
- Learning rate: 5e-4 → 1e-4
- Image patches: (64×64)
- Batch size: 64

EXPERIMENTS

Comparisons using Gaussian noise



Noise Level	Method	BM3D	FFDNet	Uformer-B	Restormer	DJF	CU-Net	DC	Ours
$\sigma = 25$	PSNR ↑	35.72	36.19	36.58	36.32	34.51	36.47	36.75	**37.09**
	SSIM ↑	0.9621	0.9654	0.9681	0.9665	0.9524	0.9673	0.9684	**0.9710**
$\sigma = 50$	PSNR ↑	32.65	33.50	34.04	33.88	32.11	33.87	34.71	**35.02**
	SSIM ↑	0.9348	0.9461	0.9515	0.9500	0.9295	0.9506	0.9558	**0.9595**
$\sigma = 75$	PSNR ↑	30.89	31.82	32.31	32.60	30.36	32.29	33.31	**33.62**
	SSIM ↑	0.9120	0.9298	0.9361	0.9400	0.9082	0.9369	0.9442	**0.9491**

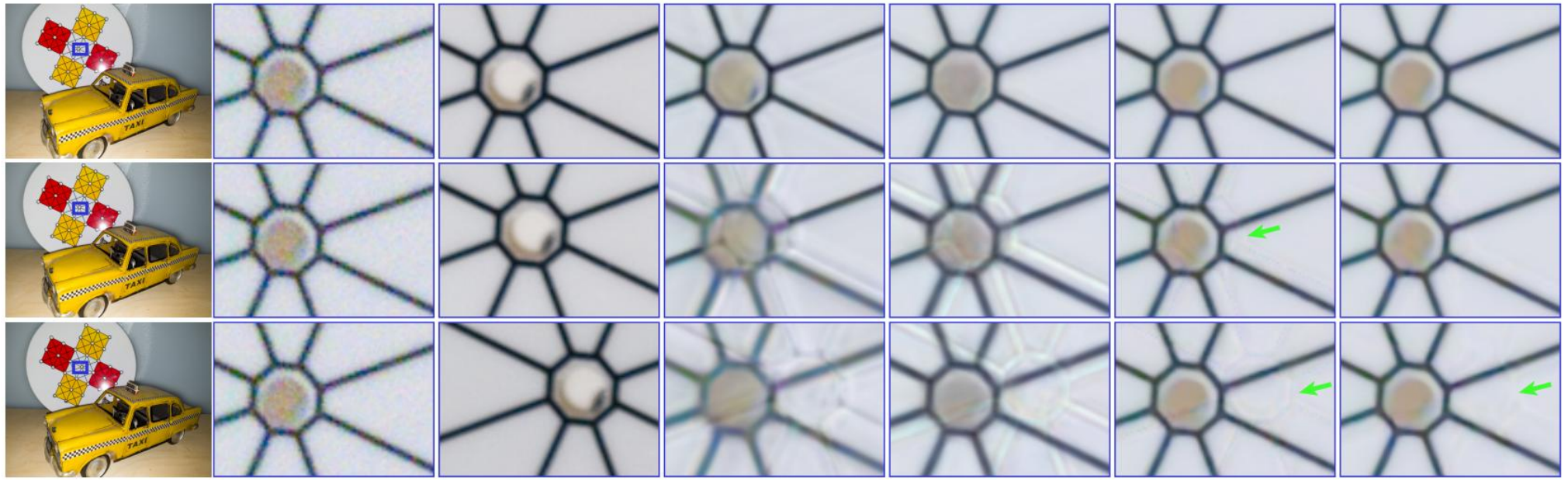
EXPERIMENTS

Comparisons using Real Noise

(a) No-flash image (b) Flash image (c) BM3D (d) FFDNet (e) DJF (f) CU-Net (g) DC (h) Ours

EXPERIMENTS

Analysis using misaligned flash/no-flash pairs



(a) No-flash image (b) Flash image (c) DJF (d) CU-Net (e) DC (f) Ours

EXPERIMENTS

Ablation studies

- Different consistent flash generations
  - Gaussian

    Produce the band-width of a Gaussian filter per pixel
  - Direct

    Produces consistent flash images

Methods	DC PSNR ↑	Gaussian PSNR ↑	Direct PSNR ↑	Convolutional k_c PSNR ↑
$\sigma = 25$	36.75	36.98	37.02	**37.09**
$\sigma = 50$	34.71	34.92	34.89	**35.02**
$\sigma = 75$	33.31	33.54	33.45	**33.62**

EXPERIMENTS

Analysis of convolutional kernel sizes

- Varying the kernel size K ×K from 1×1 to 9×9

Kernel size	1×1 PSNR ↑	3×3 PSNR ↑	5×5 PSNR ↑	7×7 PSNR ↑	9×9 PSNR ↑
$\sigma = 25$	36.75	37.02	37.07	37.09	**37.10**
$\sigma = 50$	34.69	34.97	35.00	**35.02**	**35.02**
$\sigma = 75$	33.29	33.58	33.61	**33.62**	33.61
Inference time	0.76 s	0.78 s	1.12 s	1.70 s	2.66 s

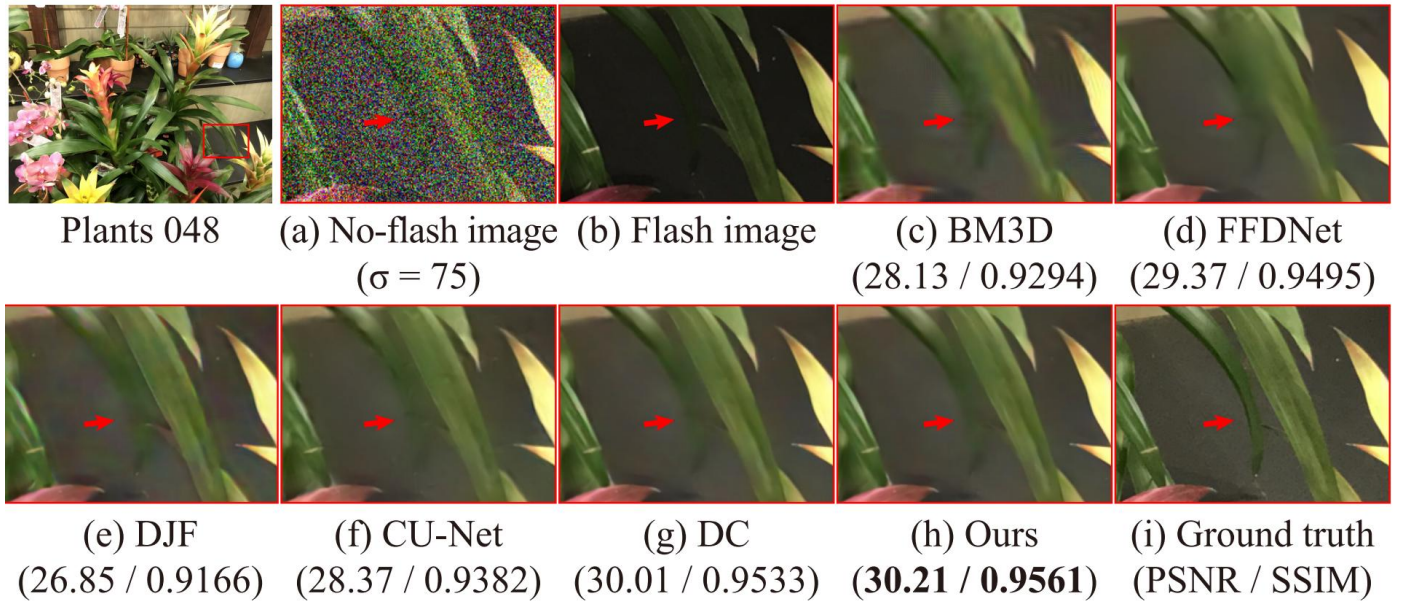
OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

CONCLUSION

Discussion of limitations and future work

- The benefit disappear when flash image doesn’t capture high-frequency details
- Do not explicitly model a misalignment



CONCLUSION

- Infer a consistent image patch, which is structurally similar to the ground truth, by applying per-pixel convolutional kernels to an input flash image locally.

- We combine a noisy no-flash image and inferred consistent image locally via a new combination model and output a denoised no-flash image.

Thanks for listening!