Deep Lensless Partial Reconstruction via Convolutional Inpainting Neural Network

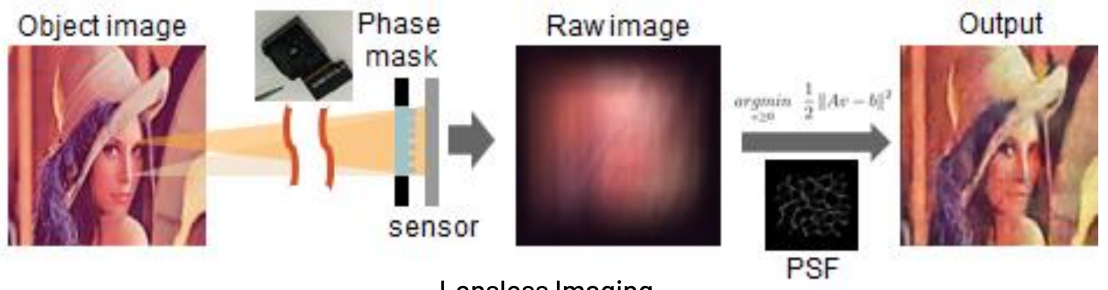
Donggeon Bae, Seung Ah Lee*
School of Electrical and Electronic Engineering, Yonsei University, Seoul, Republic of Korea
(*seungahlee@yonsei.ac.kr)

Abstract

By replacing lens modules with thin masks and computation, it can be built at a low-cost and wide application. Though iterative optimization and deep learning approaches can be used in computation these days, there are more problems facing commercial distribution. Representative problem is that not all lights from a scene can be in the camera's sensor for a wide field of view(FOV) because of diffusing property. This makes the quality of the restored image worse. In this work, This research proposes a three-staged reconstruction network for cropped(or randomly compressed) measurements, deriving missing parts that would exist without the limit of the sensor size or physical flaw in the camera.

Introduction

Lensless Imaging[1]



Lensless Imaging

- Forward Equation

$$\mathbf{b}(x,y) = \text{crop}[\mathbf{h}(x,y) * \mathbf{x}(x,y)] = \mathbf{CHx}$$

- Optimization Problem

$$\hat{x} = \underset{w\geq 0,u,v}{arg\,min}\frac{1}{2}\|\mathbf{b}-\mathbf{C}v\|_2^2 + \tau\|u\|_1$$
$$s.t.\ v = \mathbf{Hx}, u = \boldsymbol{\psi}\mathbf{x}, w = \mathbf{x}$$

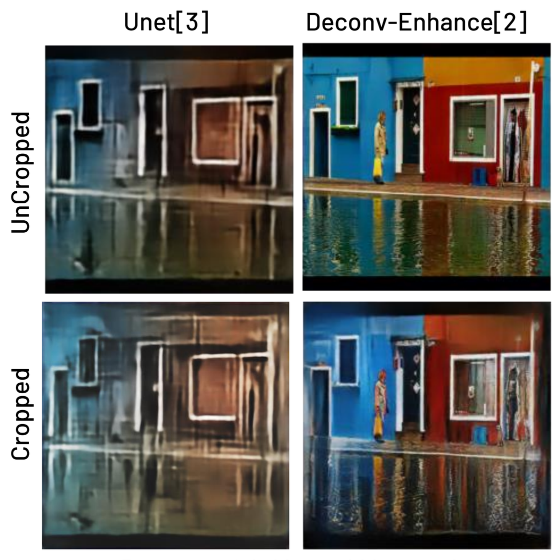
- We can model the lensless imaging system as a convolution between a scene and a point spread function(PSF).
- The measurement would be reconstructed by the iterative deconvolution means, otherwise by a learnable network[2, 3].

Lensless Reconstruction with Wiener Deconvolution & Neural Network[2,3]

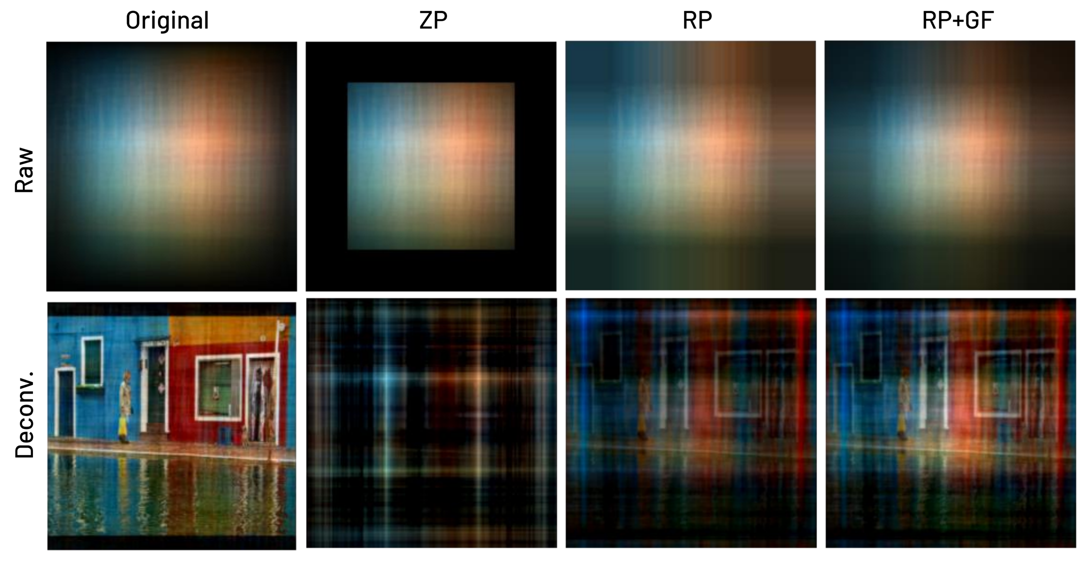
Given system $y = h * x + n$
Find g: $\hat{x} = g * y$

$$G(f) = \frac{H^*(f)S(f)}{|H(f)|^2SNR(f)+N(f)}$$

- With the given lensless system, we define the problem as filtering to separate desired scenes from the system(at this time, * PSF) with noise.
- However, the performance of Wiener deconvolution collapses as the measurement cropped.
- •Because the sensor would not absorb all lights from the scene, it captured the measurement in cropped form.



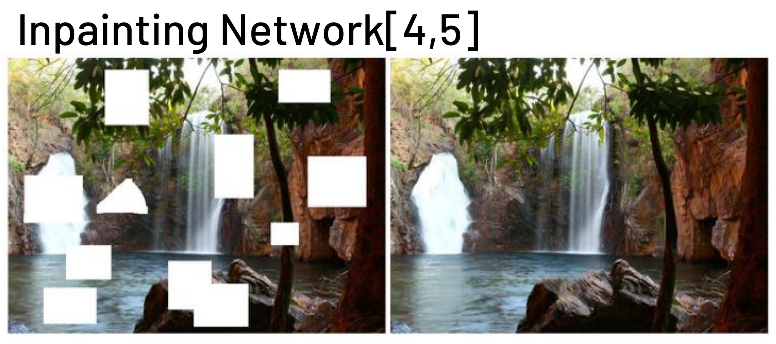
Recon. with Neural Networks



Effect of Padding on Wiener Deconvolution for Cropped Measurement

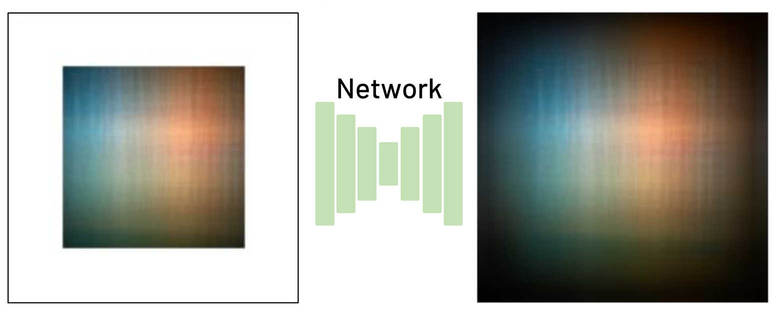
- Till now, better than zero padding(ZP), replicated padding(RP) to maximum computational size(scene+PSF feature) is an alternative to deal with this problem, additionally with gaussian filtering(GF)[2].
- We try this padding method with inpainting tasks for the well-deconvolved scene from the filled measurement.

Inpainting Network[4,5]



Inpainting GAN Result Example[5]

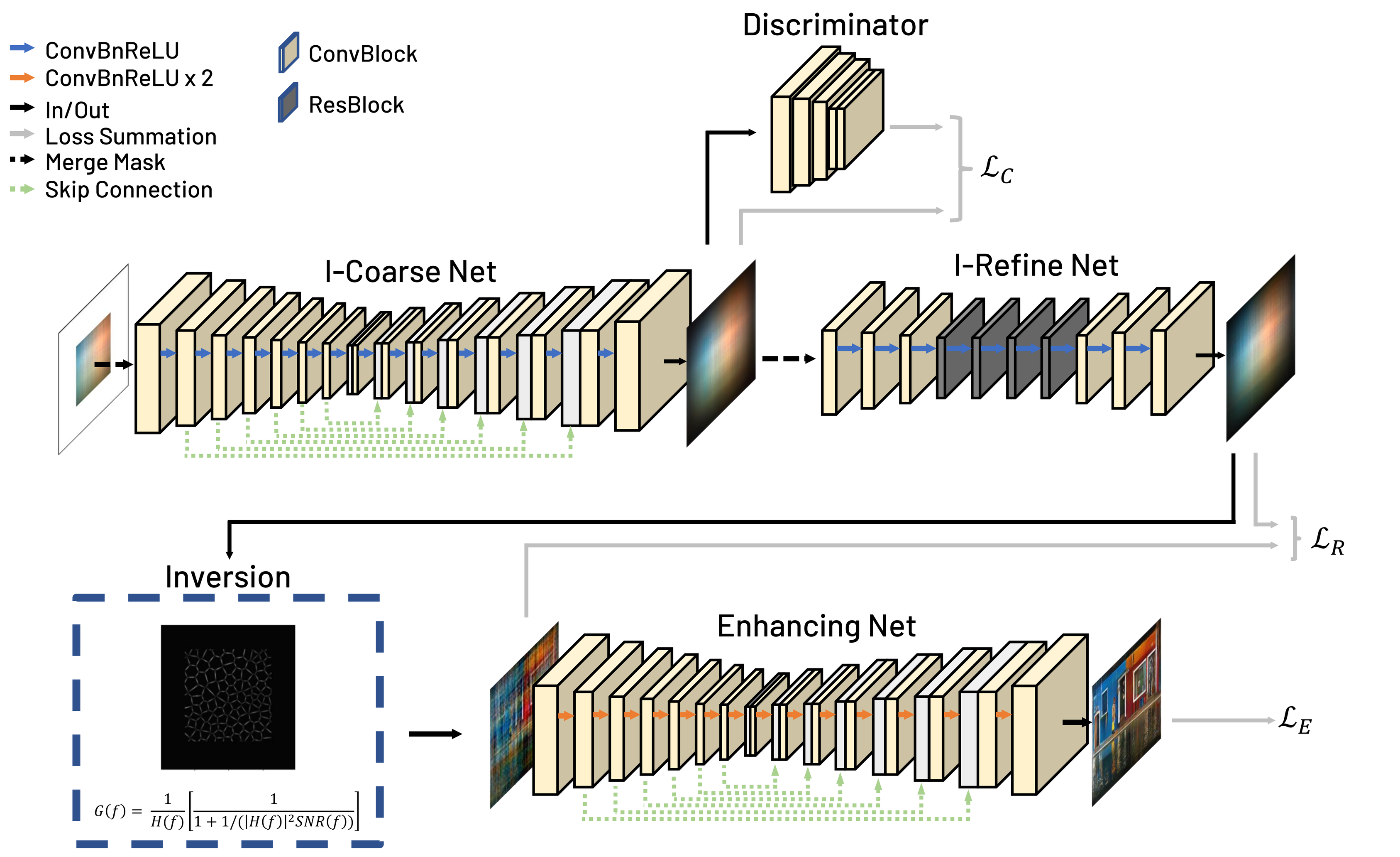
$$\mathcal{L}_G = \mathrm{E}_{\mathrm{I_{out}\sim pI_{out}}\,(\mathrm{I_{out}})}\left[(\mathrm{D}(\mathrm{I}_{out})-1)^2\right]$$
$$\mathcal{L}_D = \frac{1}{2}\mathrm{E}_{\mathrm{I\sim pdata}\,(\mathrm{I})}\left[(\mathrm{D}(\mathrm{I_{GT}})-1)^2\right] + \frac{1}{2}\mathrm{E}_{\mathrm{I_{out}\sim pI_{out}}\,(\mathrm{I}_{out})}\left[\mathrm{D}(\mathrm{I}_{out})^2\right]$$
$$\mathcal{L}_{valid} = \frac{1}{\sum \mathrm{N}(1-\mathrm{M}=1)}\|(\mathrm{I_{out}}-\mathrm{I_{GT}})\odot(\mathbf{1}-\mathrm{M})\|_1$$
$$\mathcal{L}_{hole} = \frac{1}{\sum \mathrm{N}(\mathrm{M}=1)}\|(\mathrm{I_{out}}-\mathrm{I_{GT}})\odot \mathrm{M}\|_1$$



Inpainting for Cropped Measurement

- This research aims inpainting task for a specific boundary mask following a case of cropped measurement by the sensor size.
- Inpainting network uses information from the uncropped scene, with contexts, a tendency to spread.
- By the system, a portion of the measurement contains all of the scene's information.

Model Architecture

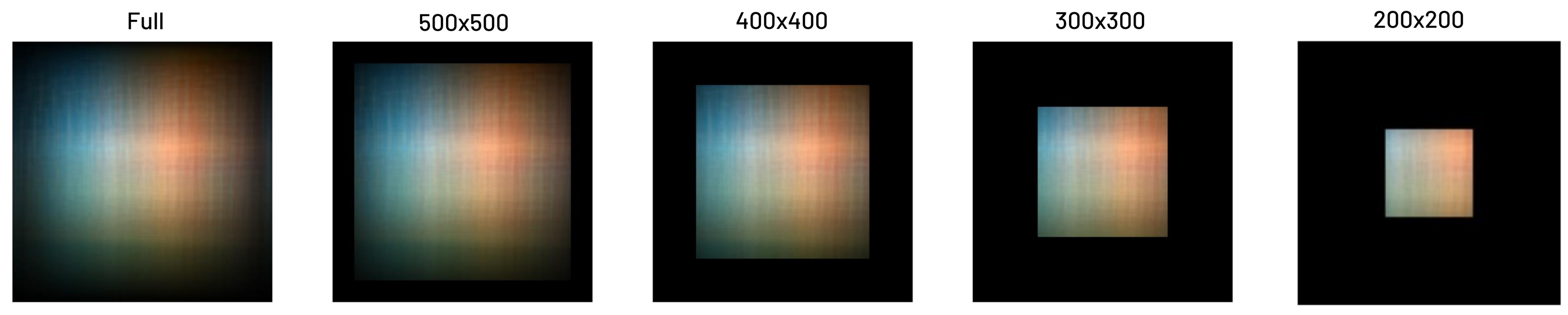


Our Proposed Model Architecture

$$\mathcal{L}_C = \lambda_{valid}\mathcal{L}_{valid}^C + \lambda_{hole}\mathcal{L}_{hole}^C + \lambda_g\mathcal{L}_G^C$$
$$\mathcal{L}_R = \lambda_{valid}\mathcal{L}_{valid}^R + \lambda_{hole}\mathcal{L}_{hole}^R + \lambda_{tv}\mathcal{L}_{tv}^R + \lambda_{per}\mathcal{L}_{per}^R + \lambda_{sty}\mathcal{L}_{sty}^R + \lambda_{decon}\mathcal{L}_{decon}^R$$
$$\mathcal{L}_E = \mathcal{L}_{\mathrm{recon}}^{\mathrm{E}}$$

- The model contains Coarse Inpainting(C), Refine Inpainting(R), Inversion(Deconv) and Enhancing (E).
- After coarse and refining with each mask merged in/out, enhancing network takes the input as Wiener-deconvolved.
- For the train, a weighted sum of a discriminator and mask-specific loss in C, plus tv, perceptual, style, and deconv. loss in R.
- For the train network E and the test, a weighted sum of MS-SSIM and L1 loss is used(same with deconvolution loss).

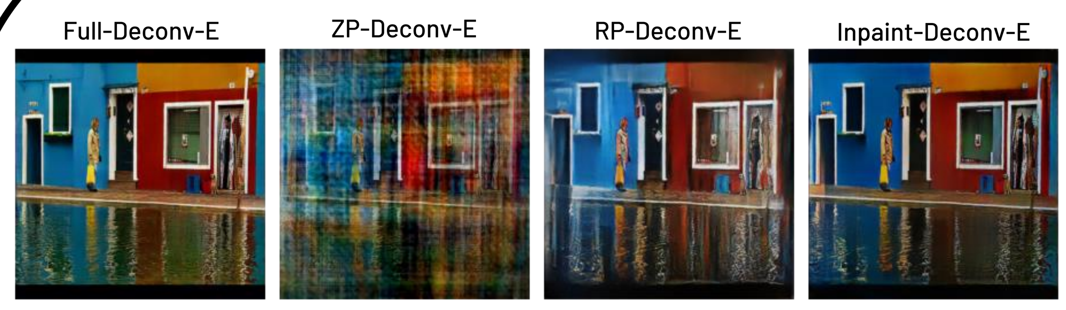
Method



Measurement Crop

- Given measurement size: 600x600, Label image size: 400x400, PSF feature size: 200x200
- For training, we convolved the PSF and 24,000 MirFlickr images for the measurement and masked with random masked area portion 0(Full size) to 8/9(200x200).
- Learning rate of 0.0001, batch size of 10, and total 100 epochs for the train.
- For the test, we used 1,000 images with random masked areas.

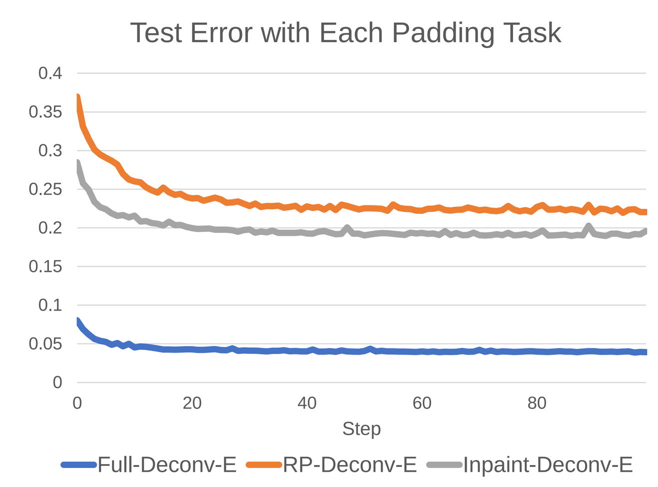
Result

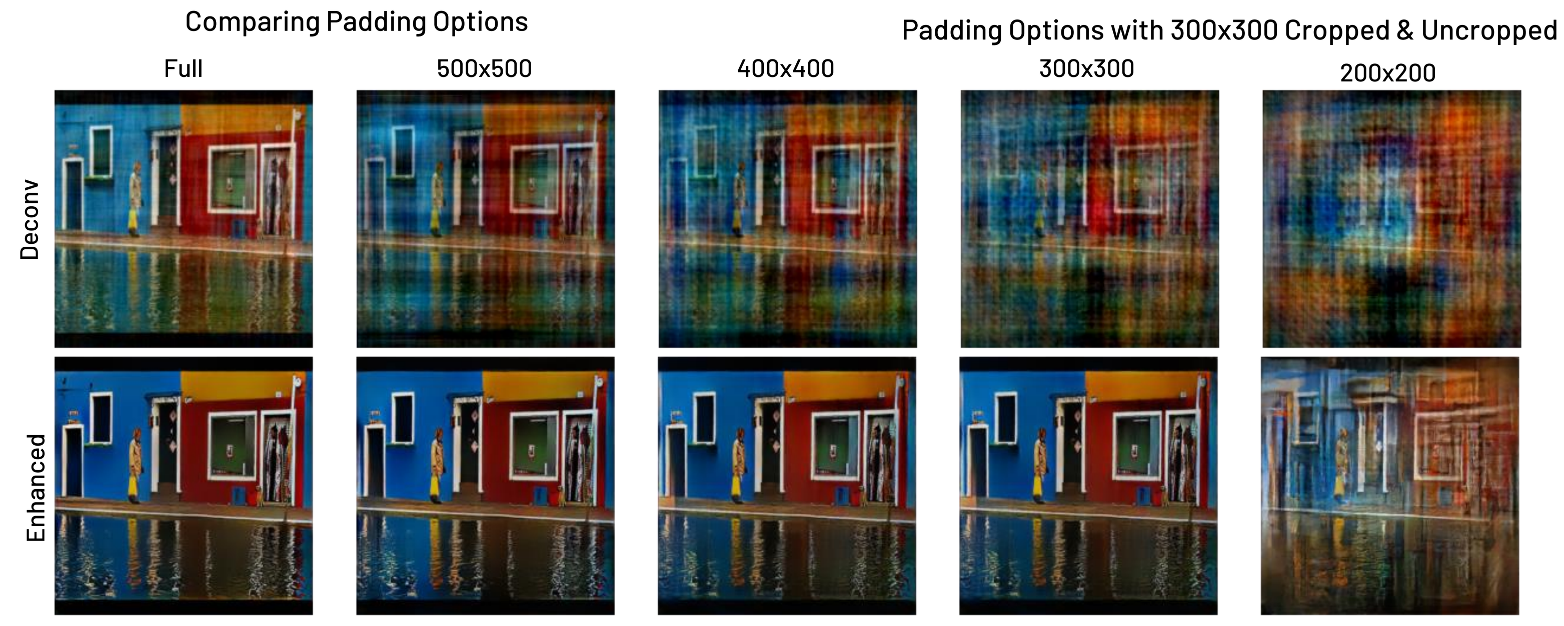


Padding option with 300x300 cropped & uncropped recon.

	Train Full Size Data Only		RP-Deconv-E	Inpaint-Deconv-E
	Full-Deconv-E	ZP-Deconv-E		
L1	0.0420	0.1526	0.1200	**0.1173**
LPIPS	0.1619	0.5617	0.4603	**0.4308**
PSNR(dB)	24.5725	14.1086	15.8507	**16.1397**

Comparing Padding Options





Padding Options with 300x300 Cropped & Uncropped

	Full	500x500 (25/36)	400x400 (4/9)	300x300 (1/4)	200x200 (1/9)
L1	0.0584	0.0852	0.1005	0.1173	0.1706
LPIPS	0.1925	0.2821	0.3298	0.4308	0.5927
PSNR(dB)	22.4523	19.8544	17.4332	16.1397	13.3119

Effect of Crop Factor on Our Model

	Unet(E)			C-R-E			RP-Deconv-E			C-Deconv-E			C-R-Deconv-E		
	L1	LPIPS	PSNR (dB)	L1	LPIPS	PSNR (dB)	L1	LPIPS	PSNR (dB)	L1	LPIPS	PSNR (dB)	L1	LPIPS	PSNR (dB)
Full	0.1352	0.5926	14.86	0.1440	0.6194	14.32	**0.0584**	**0.1925**	**22.45**	0.0631	0.2040	21.70	0.0623	0.2012	21.84
500x500	0.1379	0.6004	14.82	0.1617	0.6322	14.14	0.0892	0.2900	19.34	**0.0837**	0.2921	19.43	0.0852	**0.2821**	**19.85**
400x400	0.1428	0.6187	14.47	0.1844	0.6732	12.11	0.1040	0.4126	17.03	0.1059	0.4259	17.12	**0.1005**	**0.3298**	**17.43**
300x300	0.1676	0.6514	13.39	0.2016	0.6912	11.78	0.1200	0.4603	15.85	0.1220	0.4412	15.72	**0.1173**	**0.4308**	**16.14**
200x200	0.2198	0.7138	11.29	0.2402	0.7493	10.11	0.1800	**0.5669**	12.65	0.1751	0.5913	12.92	**0.1706**	0.5927	**13.32**
avg	0.1606	0.6354	13.77	0.1863	0.6730	12.49	0.1103	0.3845	17.46	0.1100	0.3909	17.38	**0.1072**	**0.3673**	**17.72**

Total Comparison for the Experiment

- Results show that our model improves performances in terms of L1, LPIPS, PSNR for cropped measurements otherwise slightly lower score for the full-size reconstruction. This degradation comes from robust adaptation.
- Because of inpainting tasks, training time for the whole process increases x5, while inference time increases x1.5.

Conclusion

- Our proposed network can predict unknown parts of the measurement cropped by sensor FOV, and as result, make the model separable, at the same time gets higher quality reconstruction.
- Our research also can be applied to random masks condition, concerning robust compressive imaging, which was unstable in previous methods.
- Even though there're limits such as long-time reduction or system-specific problem, the model shows developmental performances and has worth to be investigated.

Reference & Acknowledgement

[1] KUO, Grace, et al. DiffuserCam: diffuser-based lensless cameras. In: Computational Optical Sensing and Imaging. Optical Society of America, 2017. p. CTu3B. 2.
[2] KHAN, Salman Siddique, et al. Flatnet: Towards photorealistic scene reconstruction from lensless measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
[3] BAE, Donggeon, et al. Lensless imaging with an end-to-end deep neural network. In: *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2020. p. 1-5.
[4] QUAN, Weize, et al. Image inpainting with local and global refinement. IEEE Transactions on Image Processing, 2022, 31: 2405-2420.
[5] UDDIN, SM Nadim; JUNG, Yong Ju. Global and local attention-based free-form image inpainting. Sensors, 2020, 20.11: 3204.